٨ - لغات البرمجة

حين ظهر الجيل الاول من الحاسبات فى نهاية العقد الرابع من القرن العشرين ، بات جليا اختلافه اختلافا جذريا عن كل الآلات والماكينات والاجهزة الالكترونية التى ابتكرها الانسان خلال رحلته الحضارية الممتدة ، فكل ما سبق من آلات واجهزة لها دالة واحدة ووظيفة محددة وكيان واحد فيما عدا الحاسب له كيانان آلى وبرمجى ، احدهما ملموس ومدرك وآخر غير ملموس لكنه هو العامل المؤثر وخلفه يكمن النجاح فى استخدام مختلف انواع الحاسبات.

حتى الكيان البرمجى ذاته شمل قسمين ، اولاهما البيانات أو المادة الخام اللازمة لتشغيل هذا العداد الالكترونى الجبار وثانيها برامج معالجة البيانات والذى بدوره يجب نقلة من فكر انسانى الى آلات لا تعى ولا تدرك فكل ما بداخلها هو الصفر الواحد يستوى فى ذلك البيانات الرقمية أو الابجدية وإيعازات البرامج وكل الكيان البرمجى.

والواقع ان تحويل منطق حل مشكلة الى إيعازات يدركها ويتعامل معها الكيان الآلى احتاج هو الآخر الى وسيط اتصال أو لغة إتصالية بين الانسان والحاسب ، لغة مشتركة يفهمها الانسان ويسطر بها فكره ومنطقه ويدفعها الى الحاسب فاذا الكيان الآلى يستجيب لها ويعمل وفق اوامرها.

وقد ادعى ان العلماء حاولوا استخدام اللغات الطبيعية مثل اللغة الانجليزية فى تسيطر البرامج ، وقد ازعم انه جرت محاولات متعددة فى هذا الصدد باءت كلها

بالفشل لا لشيء سوى أن للغات الطبيعية تراكيب لغوية SYNTAX بالغة التعقيد كما أن دلالاتها SEMANTICS قد يشوبها كثير من عدم الدقة أو تداخل المعانى ، فاذا استخدمنا العبارة "لا تضع البيض كله فى سلة واحدة" قد تعنى المعنى المجرد ، وقد تعنى شيئا آخر لدى رجل المال والاعمال وتعنى شيئا ثالثا لدى المهندس ، وقد لا تعنى للحاسبات شيئا يذكر وربما يسمعها اناس كثيرون ولا يدركون مغزاها ودلالاتها.

من هنا جاءت الحاجة الماسة الى ابتداع لغات برمجة ذات تراكيب ودلالات لغوية محددة وواضحة لا تحتمل لبس أو اجتهاد ، يستوعبها الانسان وتزداد سيطرته عليها كلما مارس كتابة برامجه بها ، وحتى يملك ادوات اللغة قد يمضى قرابة عام فى تدارسها والتعامل بها.

على الجانب الآخر أى جانب الحاسب لابد من تحويل هذه اللغة الى الزاد الوحيد للآلات واقصد الصفر والواحد ، وهذا يؤكد لنا أن ابتكار لغة برمجة ليس امرا معضلا أو نهاية المطاف بل يتحتم ابتكار اللغة والمترجم الخاص بها ، الذى يطلقون عليه بالانجليزية TRANSLATOR أو COMPILER ، وهو ايضا برنامج مكتوب بمفردات الصفر والواحد أى لغة الآلة.

ومجمل القول أن لغة البرمجة هى مفتاح التعامل مع الحاسبات وهى التى تخضع آلاته لسيطرة الانسان وتنفيذ تعليمات برنامجة سطرا سطرا أو إيعازا إيعازا ، ويجب أن تتصف بالوضوح وتضم كلمات محجوزة RESERVED WORDS كل منها ذات دلالة واحدة ومحددة.

مستويات لغات البرمجة :

منذ ظهرت الحاسبات ظهر على الساحة اكثر من الف لغة برمجة بمعدل ٢٠ لغة جديدة كل عام تقريبا ، وفى رأيى ليس هذا دلالة قوة أو تعبير عن حاجة فعلية لهذا الكم الهائل من اللغات ، بل لو كان العالم منضبطا ومتفقا ومتفاهما ماظهر

هذا العدد من اللغات اذ يكفى ما لايزيد عن خمس عشرة لغـــة لحـــل أى مشكلـــة ، ودليلى على هذا ما اسلفت أن ما استقر لدى مستخدمى الحاسب ثلاث لغـــات هـــى كوبول - فورتران- بيزك وزمرة محدودة اخرى ، وما عداها واجه مشاكل عديـــدة وصعوبات جمة وانهارت لغات بالمئات وغاصت فى عمق الزمن وطواها التاريخ.

ومن حيث المستوى هناك لغات ارتبطت بالكيان الآلى ارتباطا كبير ، ومنها لغات فضت هذه الاعتمادية وتحولت الى لغة جيده لمختلف انواع الحاسبات ومن هذا المنطق يمكن تقسيم مستويات لغات البرمجة الى ثلاث مستويات.

١ - **لغات منخفضة المستوى ومنها:**

أ - لغة الآلة MACHINE LANGUAGE

ب - لغة التجميع ,, .ASSEMBLY

٢ - **لغات راقية المستوى:**

أ - كوبول.

ب - فورتران.

ج - بيزك ..الخ.

٣ - **لغات رفيعة المستوى.**

٤ - **اللغات الطبيعية.**

ولكل لغة من المستويات الاربع نقاط قوة ومناحى ضعف ، ونعرض الـــى المفاهيم الاساسية لكل مستوى.

١ - **لغات منخفضة المستوى :**

وهى لغات اكثر التصاقا الى الكيان الآلى ويستخدم فى صياغتها اختصـــارات اقرب الى الشفرة منها الى اللغة الطبيعية ، وقد ظهرت هذه اللغات مع الجيـــل الأول والجيل الثانى للحاسبات ، وكان لكل حاسب لغته الخاصـــة واســـلوب عنونته المميز وادخال البيانات على السجلات.

obeikandi.com

أ - لغة الآلة :

وهى ادنى مستويات اللغات منخفضة المستوى اذ هى تعليمات الآلة ذاتها مكتوبة بالنظام الثنائى حيث تدخل الى ذاكرة الحاسب مباشرة ، وتقوم الدوائر الالكترونية على تفسير وتنفيذ التعليمات لكن مع زيادة سرعة الحاسبات تلاشى استخدام هذه اللغة تماما.

ب - لغة التجميع :

وتعتبر الخطوة الاولى نحو الارتقاء بلغات البرمجة وجعلت كتابة البرامج اسهل نسبيا من استخدام لغة الآلة ، فاذا اراد المبرمج كتابه امر جمع ٥٣ ، ٧٥ فانه يجب عليه كتابة الايعازين التاليين :

حمل المسجل أ	11010101	LOAD A, 53 ;
٥٣	00110101	
حمل المسجل ب	10100011	LOAD B. 75 ;
٧٥	01001011	

ويقوم المترجم المدعو اسملبر بتحويل هذه الايعازات الى المناظر الثنائى لها ويعيب لغات المستوى المنخفض مايلى:

(١) التكرارية.

(٢) استهلاك وقت كبير فى صياغة برامج بسيطة.

٢ - لغات راقية المستوى HIGH LEVEL :

ظهرت لغات المستوى الراقى على الساحة عام ١٩٦٠ بظهور لغة فورتران ولغة كوبول والتى احتلتا الساحة حتى الآن ، وجاءتا اقرب الى اللغة الإنجليزية منها الى لغة التجميع ، وان امتازت هذه اللغات بالدقة والحسم والتحديد مما ساعد المبرمجون على اعداد برامج اكثر تعقيدا ، كما فكت الارتباط بين الكيان الآلى والبرامج.

بالطبع تتطلب هذه اللغات عمليات ترجمة الى لغة الآلة بواسطةCOMPILER خاص بكل لغة ويتمشى مع طرز الحاسب ، وهنا يجدر القول أن مترجم لغة كوبول لحاسب IBM لايصلح لمترجم حاسب من انتاج NCR أو ICL .

٣ - **لغات رفيعة المستوى VERY HIGH LEVEL :**

وتسمى ايضا لغات الجيل الرابع للحاسبGENERATION LANGUAGE 4 Th واكتسبت اسمها ومستواها 4 GL ان ايعاز واحد منها يكافئ ويعادل عشرات ومئات الإيعازات من اللغات راقية المستوى لانها ليست لغة اجرائية مثل اللغات السابقة بحيث يجب على المبرمج كتابة كل التفاصيل اللازمة للمعالجة وكل خطوة يريد اجرائها ، انما عليه أن يعرف ماذا يريد فقط WHAT وليس شرطا أن يعرف كيف HOW يحقق هذا الهدف وهذا ماتحققة لغات 4 GL .

وعلى هذا فهى لغات تمتاز بالبساطة والسهولة والانتاجية العالية وتحتاج جهدا اقل فى التدريب وصولا الى الاجادة الكاملة.

٤ - **اللغات الطبيعية:**

وهى لغات يصعب حتى الآن اعداد مترجم لها لاعتبارات خاصة بالقواعد والدلالات ، ولازالت تجرى عليها بحوثا مجهده ومكلفة لتحقيق الذكاء الصناعى للحاسبات.وعموما فهى خارج عن نطاق هذا الكتاب.

بعض لغات البرمجة راقية المستوى:

١ - **لغة فورتران FORTRAN FORMULA TRANSLATOR :**

ابتكرتها شركة IBM عام ١٩٥٧ وبظهورها انسحبت امامها كل اللغات فيما عدا لغة كوبول وتمتاز بقدرة على معالجة العمليات الرياضية والهندسية وتضم مكتبة كبيرة من الدوال الرياضية كما يمكنها التعامل مع المصفوفات بسهولة ويسر وبرامجها لايسهل قراءتها حتى على بعض مبرمجيها.

٢ - لغة COBOL كوبول:

وهى لغة تصلح للتعامل مع المدخلات الكثيفة والعمليات الحسابية البسيطة لذا لاقت نجاحا كبيرا فى التطبيقات التجارية ، كما تشبة كوبول فى صياغتها اللغة الإنجليزية ويسهل قراءتها وفهمها حتى على غير ذوى الخبرة فى الحاسبات ، وتنقسم الى فقرات وجمل ونقط ايقاف الى جانب وصف البيانات والآلات والاجراءات وصفا ممتدا ، مما يستدعى استخدام كلمات محجوزة كثيرة ولذلك تتطور اللغة باستمرار لدرجة أن مترجم لاحدى اصداراتها لايصلح لترجمة برنامج مكتوب باصدار آخر .

٣ - لغة البيزك BASIC :

صدرت النسخة الاولى منها عام ١٩٦٥ وكان الهدف منها خدمه المبتدئين لبساطتها الشديدة ، مثلا عاملت الارقام على انها ارقام حقيقية مثل (١،٧٦٨٥٨) اما اذا رغب المبرمج فى استخدام الاعداد الصحيحة فلابد أن ينص على ذلك ، لذلك لاقت اللغة نجاحا كبيرا على الحاسبات الشخصية والصغيرة ، ورغم التطورات التى ادخلت عليها مؤخرا مثل تربو بيزك TURBO فلا زالت ابسط واسهل اللغات قاطبة .

٤ - لغة بسكال PASCAL :

ظهرت عام ١٩٧١ وتعتبر من ابسط اللغات واحدى اللغات المنافسة للغة البيزك وتستخدم اسماء مسميات ممتدة مما يسهل فهم البرنامج .

٥ - لغة C :

اقرب الى لغات المستوى المنخفض منها الى اللغات الراقية وتستخدم فى كتابة برامج التشغيل ، وهى لغة صعبة الى حد ما .

٦ - لغات اخرى:

البرولوج - الفورت - LP / 1 - لوجو - ليسب - APL .

ترجمة البرامج COMPILATION :

اوضحنا أن اللغات راقية المستوى يجب ترجمتها الى لغه الآلة وبذا يتحول البرنامج المصدر " SOURCE PROGRAM أى البرنامج الذى تمت صياغته بـــاحدى لغات البرمجة الراقية" ، الى برنامج هدف أى تحويله الى لغة اله وقد جرى اتفـــاق عام على جملة من التعريفات لبرامج الترجمة على النحو:

أ - المجمع اسملبر ASSEMBLER :

هو البرنامج الذى يحول البرنامج المصدر بلغه التجميع الى لغة الآلة.

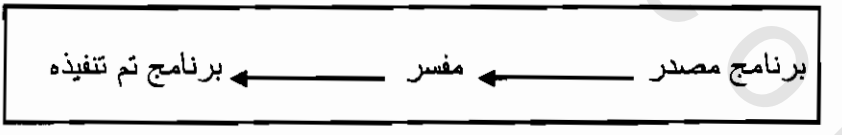
ب - المترجم TRANSLATOR أوCOMPILER :

وهو برنامج يترجم برامج اللغات مرتفعه المستوى وذلك دفعه واحدة ويحوله الى برنامج هدف يصبح جاهز للتنفيذ متى طلب ذلك.

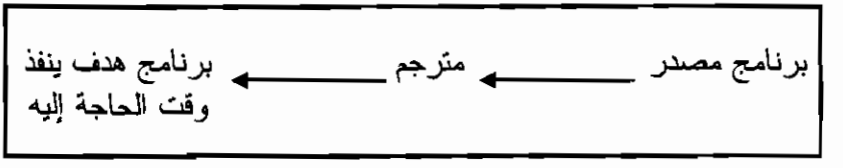
ج - المفسر INTERPRETOR :

ويقوم على ترجمة البرنامج المصدر ايعازا ايعاز أو خطوة خطوة ولا ينتقـــل الى الايعاز التالى الا بعد تنفيذ الايعاز الذى سبق ترجمته.

ان افضل خصائص المفسرات انها تقوم على تنفيذ البرنامج مبـــاشره خـــلال مرحلة واحدة لذلك تفضل المفسرات على المترجمات فى الحاسبات الشخصية وغالبا ماتستخدم مع البرامج المكتوبه بلغه البيزك فيما نركز عليه فى الاطـــار التالى :



بينما نبرز خصائص المترجمات:



ان اجراء مناظرة من واقع الحياة يبين البرنامج المترجم والبرنامج المفسر تقودنا الى هذا المثال ، فعند تشغيل البرنامج الهدف على الحاسب فانه يشبه كمن يقرأ رواية مترجمه عن لغتها الاصلية الى اللغة العربية ، بينما تنفيذ برنامج مفسر تشابه الى حد كبير قراءة نفس الرواية بلغتها الأصلية ، وعند قراءة كل جملة يتوقف القارئ ليترجم معناها ، ويتضح من المثالين مدى سرعة انجاز قراءه الرواية المترجمة سلفا ومدى بطىء الاخرى ، ومع هذا فقد ازعم ان الفارق بين المترجم COMPILER المفسر فرق فلسفى لا اكثر ولا اقل لان كلاهما مضطر الى تحويل البرنامج الى لغة الآلة.

مزايا المفسرات :

هناك مزايا تفيد المبرمجون من استخدام المفسرات نذكر منها:

أ ــ التعامل مع الإيعازات مباشرة على الحاسب INTERACTIVE .

ب ــ كشف اخطاء البرنامج فوراً.

ج ــ المساعدة فى حل مشاكل ربط جهازى حاسب غير متماثلين.

د ــ مساندة جهد المبرمج عند استخدام طبعة قديمة من اصدارات لغة البرمجة.

مراحل ترجمة البرنامج المصدر:

تمر عملية ترجمه البرنامج المصدر بعدة مراحل ابرزها مرحلة التحليل اللغوى LEXICAL وخلالها يقوم البرنامج المترجم بتحديد الإيعازات التى تتطلب ترجمة ، فالمترجم ينظر الى سطور البرنامج على انها مصفوفة من الحروف يتم تحويلها الى المناظر الثنائى وعليه قبل القيام بهذا التحويل فحص الإيعازات لتحديد الكلمات المحجوزة وبديلا عن ترجمتها حرفيا يأتى بالمناظر لها مباشرة لان للكلمات المحجوزة معنى محددا عند المترجم ضمن اطار تصميم لغة البرمجة.

مثال:

```
110    LET    XYZ    =        4179896
```

فكلمه دع LET كلمه محجوزة فى لغة البيزك تترجم كما هى بينما XYZ اسم متغير يترجم X ثم Y ثم Z بعدها يعثر على علامة يساوى وهى من المشغلات الحسابية ولها مناظر ثنائى ثم الرقم ورغم انها اعداد الا انه يعبر عن قيمه لذا يتم ترجمته وتحليله، وبعد اتمام مرحله التحليل اللغوى يبدأ فى مرحله مراجعه قواعد اللغة وهل هى دقيقة التعبير وفق القواعد المحددة وماهى العبارات المرتبطة بها:

```
IF      فعل الشرط      THEN  جواب الشرط  ELSE
        DOWHILE  .....
```

وفى المرحلة النهائية للترجمة يتم اعداد الأكواد اللازمة للبرنامج المصدر وربطه مع الروتينات اللازمة وبذلك يتحول الى البرنامج الهدف.

واجبات المترجم:

تتلخص اعمال البرنامج المترجم فى تأدية الوظائف التالية :

١ - التأكد من دقة وصحة قواعد اللغة المستخدمة وتحديد الاخطاء اللغوية.

٢ - وضع البرامج الفرعية اللازمة للبرنامج ضمن سياق الترجمة (اجمع – أحسب الجذر التربيعى...إلخ).

٣ - كشف بعض الاخطاء المنطقية المحدودة مثل القسمة على الصفر أو قسمة صفر على صفر.

٤ - اصدار طبعة مقروءة من البرنامج المصدر والاخطاء التى تم الكشف عنها ان وجدت.

٥ - استخدام برامج LINK وبرنامج المحرر EDITOR بما يتيح للمبرمج القدرة على تصحيح اخطاء البرنامج على الشاشة مباشرة.

لغات البرمجة العربى:

فى محاولات جرت وتجرى لتعريب الكيان البرمجى للحاسبات الصغيرة ثم انتاج نسخ أو ترجمات عربيه للغة البيزك منها :

لغة نجلاء وقد نفذت على حاسب ماركة الفارابى ، ولغة الخوارزمى على جهاز الرائد ، ولغة سعودية على حاسب عرب رام.

وهذه اللغات الثلاث سهلة التعلم والاستخدام ولا تحتاج لاى معرفة سابقة فى الحاسبات لذلك فهى مثالية لتعليم البرمجة للهواه والمبتدئين وبما انها باللغة العربية مائه بالمائة فهى مناسبة جدا لتعليم الطلبة مبادئ البرمجة.

مثال:

برنامج يجمع الارقام الفردية من ١ الى ٩٩ ويطبع المجموعة ؟

الحـــل :

بلغة نجلاء :

```
١٠ دع  م = ٠
٢٠ س   = ١ حتى ٩٩ يضاف ٢
٣٠ دع م = م + س
٤٠ كرر س
٥٠ اطبع م
٦٠ انتهى
```

ومن هذا المثال يتضح التشابة الكبير بين لغة نجلاء ولغة بيزك ، لكنها خطوة على الطريق ، ودائما رحلة الالف ميل تبدأ بخطوة.

obeikandl.com

٩ - اساسيات لغة البيزك

يتناول هذا الباب استخدام لغة البيزك فى كتابة برامج بسيطة ترتكن على ما تم عرضه فى خرائط التدفق ومنطق البرمجة الذى يجب الالمام به المامـــا جيـــدا ، ويعرض هذا الباب الى العناصر الاساسية والضرورية لتفهم لغة البيزك التى تعتـــبر احد ابرز لغات البرمجة ويعتبر اجادتها خطوة اساسية لتفهم باقى لغات الحاسب فاذا اضفنا اليها التعديلات الهائلة التى جرت عليها فان المستقبل لن يغفل لغة البيزك بـــل ستتدعم وتقوى وتستمر خلال القرن القادم كلغة متطورة نامية سهلة الفهم والاستيعاب ، تقدم كل الخدمات للمبرمجين سيان فى اعداد البرامج مهما بلغت من التعقيد وتقـــدم لهم اساليب اعداد الرسوم وفتح الملفات... انها لغة شاملة جديرة بكل الاهتمام مـــهما ابتكر من لغات برمجة.

عناصر لغة البيزك:

دعنا نلقى نظرة على البرنامج التالى حتى نتفهم عناصر لغة البيزك:

```
10      REM THIS PROG IS  M.N. SWELAM, PROG
20      REM
30      PRINT  "MONTH" , "NAME" , "DAYS"
40      PRINT
50      FOR I  = 1  TO  12
60      READ MONTH$ ,  DAYS
70      PRINT I , MONTH$ , DAYS
80      NEXT I
```

```
90      DATA  JAN , 31, FEB , 28, MARCH ,31, APRIL ,30,
100     DATA  MAY ,30, JUNE ,30 , JUL , 31,  AUG , 31
        DATA SEPT ,30, OCT ,30, NOV ,30, DEC ,31
110     END
        RUN
```

MONTH	NAME	DAYS
1	JAN	31
2	FEB	28
3	MAR	31
.	.	.
.	.	.
12	DEC	31

من دراسة البرنامج السابق يمكننا التعرف على خصائص اللغة على النحو:

أ ــ البرنامج عبارة عن مجموعة مرتبة من الإيعازات وكل ايعاز يكتب فى سطر مستقل.

ب ــ لكل سطر رقم وحيد يتزايد تصاعديا بمقدار ١٠ لاعطاء الفرصة لاضافة إيعازات جديدة.

ج ــ كل ايعاز يبدأ بأمر مثل اقرأ.

د ــ يمكن اجراء دوارات بالامر FOR NEXT واوامر اخرى سنتناولها فى الصفحات اللاحقة.

و ــ ينتهى البرنامج بكلمه END التى تخطر الحاسب بأن هذه آخر جملة فى هذا البرنامج.

ز ــ قد يحتوى السطر الواحد على اكثر من ايعاز بشرط فصل كل ايعاز عن الآخر بالعلامة (الحرف) (:).

ح ــ تستخدم كلمة REM فى الاسطر المفكرة لكاتب البرنامج وهى اسطر لا تنفذ على الحاسب ، وكلمة REM هى اختصار REMARKS "ملاحظات".

ط ــ كتابة الامر PRINT دون ذكر اسماء متغيرات يفرز سطر طباعة خال.

إيعاز البيزك:

يتكون ايعاز البيزك من رقم السطر ــ الامر ــ المحددات:

```
nnn  { BAISC INSTRUCTION} { PARAMETERS }
70     PRINT    I, "$MONTH",    DAYS
```

STOP

قد يحتوى البرنامج على الامر STOP وهو امر لا يوقف أو ينـــهى تنفيـذ البرنامج بل يتوقف الحاسب عن التنفيذ مؤقتا حتـى زوال الظـروف مثـل سـيارة محركها يعمل لكنها لا تنطلق مما يعطى الفرصة للمبرمج لدراسة نتائج الخطـــوات السابقة ومدى مطابقة النتائج للمتوقع ويعـــاود البرنـامج التنفيـذ بـالأمر أستمر CONTINUE من على لوحة المفاتيح.

مفردات لغة البيزك:

تضم اللغة مفردات اساسية تستخدم فى صياغة الإيعازات وتشمل الآتى:

١ ــ فئة الحروف:

وتشمل الحروف الهجائية والارقام والحروف الخاصة والمعاملات الحســابية كما تشمل الفئة علامات الاقتباس والتعجـــب والســؤال وعلامـــة يســاوى والمعاملات المنطقية ، مثل ، اكبر من ، واصغر من ، ولا يســـاوى ، الـــى جانب الفصلة والنقطة والفصلة المنقوطة والعلامة (:).

٢ ــ الثوابت:

الثوابت قيم لا تتغير اثناء تنفيذ البرنامج وقد تكون ثوابت رقمية مثل الارقـــام الطبيعية والارقام الصحيحة والآسية أو ثوابت هجائية:

* ارقام صحيحة INTGERS [15- , 1- , 0 , 9 , 15 , ... ,].

* الارقام الطبيعية وهى التى تشمل كسر عشرى وقد تكون موجبة أو سالبة.
* الأسية وتسمى ارقام متحركة FLOATING ولتبسيط العمليات الحسابية تقرب لاقرب اس 52.23EO2 تعنى٢٣ ,٥٢ مضروبة فى عشره أس (٢).

اما الثوابت التى لا تجرى عليها عمليات حسابية فهى الثوابت الهجائية فقد تكون خليطا من فئة الحروف والأعداد والعلامات الخاصة وتوضع بين علامتى اقتباس مثل:

"MN X 4+ 5 - ! ? ALI KEREA "

وتطبع كما هى.

قواعد تمثيل الثوابت الرقمية :

أ - تستبدل العلامة العشرية بنقطة.

ب - يمكن تحديد القيمة موجبة (+) أو سالبة (-).

ج - يمكن أن يطول الرقم إلى تسعة أعداد متجاورة .

د - إذا زادت الأعداد عن تسعة يمكن تمثيلها بطريقة أسية.

مثال:

حدد نوع كل من الثوابت الرقمية التالية موضحاً الصواب والخطأ من حيث إمكانية تمثيلها فى لغة البيزك:

الكلمة	النوع	إمكانية تمثيله	السبب
+ ٣٥	صحيح موجب	يمكن	
٣٥	،، ،،	يمكن	
- ٣٥	،، سالب	يمكن	
٠ ,٢٤٣٦٧٩٣٥٧٢	حقيقى موجب	لا يمكن	خطأ - لزيادة الأعداد عن تسعة

مثال:

وضح كيفية تمثيل الثوابت المدرجة فى الجدول بلغة البيزك :

الثابت الرقمى	الشكل الأسى	التعبير فى لغة بيزك
٩٧٦ ,٠٠٠ ,٠٠٠	٩,٧٦ × ١٠ ^٨	9.76 E + 08 0.976 E + 09
٠,٠٠٠٠٠٠٩٧٦	٩,٧٦ × ١٠ ^-٨	9.76 E - 08 97.6 E - 07
٨,٠٩ × ١٠^١٦		8.09 E + 16

٣ ــ المتغيرات:

سبق واوضحنا معناها فى الابواب السابقة وانها اسماء لمواقع تخزين فى الذاكرة واسمها ثابت على امتداد البرنامج الواحد لكنها ذات قيم متغيرة ، ففى البرنامج الاختبارى فى صدر هذه الباب نلحظ MONTH اسم لموقع فى الذاكرة تغيرت قيمته اثنى عشرة مرة ، وتنقسم المتغيرات الى نوعين:

أ ــ متغيرات رقمية:

ويستخدم فى تسميتها حرف واحد أو عده حروف ابجدية أو حرف ابجدى كبداية ثم أى حرف من فئة الحروف ، ويبلغ عدد المتغيرات الرقمية المتاحة للبرنامج الواحد ٢٥٦ متغير ويجب التأكد من عدم استخدام علامة الدولار $ كآخر حرف فى اسم المتغير الرقم لان اضافة علامة الدولار تعنى فى لغة البيزك أن المتغير هجائى: ونعرض الى بعض اسماء المتغيرات الرقمية مثل:

A , B2 , ASHRAF - 5X , XY1Z , M

ب ــ متغيرات هجائية:

ويستخدم فى تعريفها حرف أو عدة احرف يتلوها فــى النهايــة علامــة الدولار على النحو:

EHAB$, 5A$, M$, XYZ$ ويبلغ عدد المتغيرات الهجائية ٢٦ متغــيرا على امتداد البرنامج الواحد.

٤ ــ العمليات الحسابية والمنطقية:

التعبيرات أو الجمل الحسابية عبارة عن تعبير رياضى يضم ثوابت ومتغيرات ومعاملات رياضية مثل الجمع [+] ، الضرب [*] ، الطرح [-] ، القسمة [/] ، الاس [↑] أو [**] كما تفصل التعبيرات عن بعضها البعــض باســتخدام الاقواس الداخلية ثم الخارجية وهكذا وعند تنفيذ البرنامج على الحاسب تســبق عمليات اس عمليات الضرب تتلوها القسمة ثم الجمع والطرح.

كما تشمل العمليات المنطقية المعاملات التالية:

يساوى = لا يساوى < >

اكبر من > اصغر من <

اكبر من يساوى => اصغر من يساوى =<

مثال:

اكتب كيف يحل الحاسب المعادلة الرياضية التالية:

$$س = ل^{٢} + \frac{٥ (ب - جـ)}{د هـ}$$

الحــل:

س = ل ** ٢ + ٥ * (ب - جـ) / (د * هـ)

ويتم ترتيب الحل على النحو:

أ ــ فك الاقواس وبالتالى تنفذ العملية (ب - جـ) والعملية (د * هـ).

ب ــ ينفذ الاس ل ** ٢.

ج ــ تجرى عملية الضرب.

د ــ تجرى عملية القسمة.

هـ ــ تتم عمليات الجمع والطرح.

و ــ رصد النتيجة النهائية.

٥ ــ اوامر التخصيص LET :

وتستخدم كما اوضحنا فى باب منطق البرمجة فى الآتى :

أ ــ فتح مخازن مؤقتة فى الذاكرة لرصد النتائج.

ب ــ فتح العدادات.

ج ــ تخصيص عملية رياضية.

د ــ ادخال البيانات وان كان يعيب هذه الطريقة عدم سهوله تغيير البيانات خلال مراحل تنفيذ البرنامج لذلك يجب قصر الأمر على عمليات التخصيص وفتح المخازن والعدادات على النحو الموضح فى الامثلة التالية:

```
10      LET   A  = 573
10      LET   A$ = "SOWE 4179896"
10      LET   M  = X ** 2 + y
10      LET   COUNT = 0
10      LET   SUM = 0
```

امثلة توضح استخدام الامر LET "دع" فى ادخال البيانات:

مثال (أ):

```
10      LET     A = 5
20      PRINT   A
30      END
        RUN
         5
```


obeikandl.com

مثال (ب):

```
10      LET    A = 25
20      LET    B = 44
30      PRINT A
40      PRINT  B
50      END
              RUN
        25
        44
```

مثال (ج):

```
10      LET    A = 45
20      LET    B = 126
30      LET    C = 8
40      LET    N = (A + B ) / C
50      PRINT N
60      END
        RUN
        21.375
```

مثال (د):

```
10      LET    A$ = " AHMED "
20      LET    B$ = "ALI "
30      N$  =  A$ + B$
40      PRINT N
50      END
        RUN
        AHMED ALI
```

مثال (هـ):

```
10      LET    A  =  25 : LET B = 44
20      LET    S  = A+ B : PRINT S
```


obeikandl.com

```
30      END
        96              RUN
```

ملحوظة:

يمكن لمخطط البرامج عدم كتابه LET ويتعامل معها النظام بـ DEFAULT.

أوامر المدخلات والمخرجات :

■ **اوامر ادخال البيانات:**

يتم ادخال البيانات باحدى اسلوبين خلاف استخدام الامر LET :

١ ـ استخدام الامر READ ، والامر DATA :

مع البرنامج على هيئه ثوابت فى نهاية اسطر البرنامج عند استخدام الامر اقرأ READ على أن يفصل بين كل قيمة والتى تليها فاصلة:

```
10      READ X, Y,Z                 ١٠ - اقرأ [أسماء المتغيرات]
20      .........                   ٢٠ - .....
30      .........                   ٣٠ - .....
70      PRINT  ,  ,                 ٧٠ - أطبع على الشاشة
90      LPRINT  ,  ,                ٩٠ - أطبع على الطابعة
100     DATA  3, 5, 7, 8, 11, -2, 15, 173, 14   ١٠٠ - البيانات
110     END                         ١١٠ - نهاية
        RUN
```

وجدير بالذكر أن سطر البيانات يمكن وضعه فى أى مكان من البرنامج مع مراعاة أن يكون ترتيب قيم المتغيرات متمشيا مع الامر DATA وفق ترتيبها ونوعيتها فى الامر READ ، وهذا يدل على أن الامر READ

obeikandl.com

والامر DATA كلاهما مكمل للآخر وتوأم له ولا يمكن استخدام أيا منهما منفردا.

٢ ـ الامر INPUT :

وهذا الامر يعنى الاتصال المباشر مع الحاسب وادخال البيانات عن طريق لوحة المفاتيح ويصاغ امر ادخال بيانات البرنامج بالامر INPUT على النحو:

```
10      INPUT  A , X$
.       .........
.       .........
.       .........

100     END
        RUN
        ?
```

وعندما يبدأ تنفيذ البرنامج تظهر علامة الاستفهام لطلب المدخلات حسب توصيفها فى الامر INPUT وبنفس الترتيب وظهور علامة الاستفهام تعنى أن البرنامج صحيح وأن الحاسب مستعد لتلقى المدخلات ، ولن ينفذ الحاسب السطر التالى الا بعد ادخالها مع الالتزام بنفس ترتيب المتغيرات فى الامر INPUT.

امثله على استخدام الأمر READ والأمر INPUT :

مثال (١):

```
10      READ  A, B, C, D
20      DATA  842, 258, 75, 25
30      K = ( A + B ) / (C -D)
40      PRINT K
```


obeikandl.com

```
50      END
RUN
22
```

مثال (٢):

```
10      READ  A, B, C
20      READ  D
30      DATA  ,  ,  ,  ,
40
50      END
RUN
```

مثال (٣):

```
10      INPUT  A, B
20      S  = A + B
30      PRINT  S
40      END
        RUN
?       23
?       85
108
```

مثال (٤):

```
10      INPUT " FIRST NUMBER " ; A
20      INPUT " SECOND NUMBER " ; B
30      M  =  A  +  B
40      PRINT  M
50      END
RUN
FIRST NUMBER  A ? 15
SECOND NUMBER  B ? 25
```

obeikandl.com

■ **اوامر اخراج المعلومات:**

يتم اخراج المعلومات باحدى الاسلوبين:

أ ــ اظهارها على شاشة الحاسب باستخدام الامر PRINT.

ب ــ طباعتها على الورق باستخدام الامر LPRINT.

ملاحظات حول الطباعة:

(١) استخدم الامر LPRINT ضمن اسطر البرنامج دون كتابه اسـماء المتغـيرات يترك سطر خال.

(٢) استخدام الفاصلة بين اسماء المتغيرات على النحو:

```
50 LPRINT  X, Y, Z, M, N, L, A$, B$
```

يظهر الطباعة على سطر واحد لخمس متغيرات وباقى المتغيرات على السطر التالى.

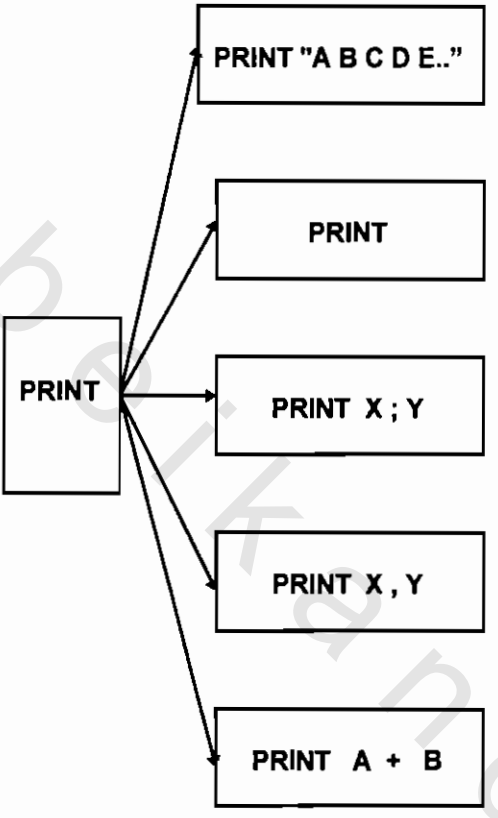
(٣) استخدام الفاصلة المنقوطة (;) بين اسماء المتغيرات يظهر جميع قيم المتغيرات على سطر واحد.

(٤) يمكن جدولة النتائج المطبوعة باستخدام الامر TAB (اختصار TABLE) مـع تحديد اسم المتغير وطول المسافة الخالية على النحو:

```
70      LPRINT
80      LPRINT TAB  (5); A; TAB (15); B, TAB (25) C;
```

ويمكن تلخيص عمل أمر الطباعة PRINT على النحو التالى:

obeikandl.com



لطباعة مسلسل حرفى كما هى مدونة دون تغيير.

تترك سطر خالى أثناء الطباعة.

يطبع القيم بفاصل مسافتين فقط ويظهر جميع القيم فى سطر واحد .

يطبع القيم بفاصل ١٥ مسافة ولايظهر أكثر من خمس قيم فى السطر الواحد.

يجرى طباعة لناتج العملية الرياضية .

مثال :

إذا كانت المتغيرات هى A , B , C , D, E, F, G, H, I, J وكانت قيمها علـــى النحو التالى 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 فما هو ناتج جمل البيزك التالية :

```
10    PRINT A ; B; C ; D ; E ; F ; G ; H ; I ; J
```

الناتج

```
0 1 2 3 4 5 6 7 8 9
10    PRINT A , B , C , D , E , F , G , H , I , J
```

الناتج

```
0   1   2   3   4
```

```
      5  6  7  8  9
10    PRINT A , B, C , D , E , F , G , , , H , I , J
```

الناتج

```
0  1  2  3  4
5  6        7
8  9
```

■ **امر نهاية البرنامج END :**

يجب أن يحتوى كل برنامج مكتوب بلغة البيزك على امر النهاية END فى آخر سطر مرقم وأى ايعاز يظهر بعد ذلك سوف يتجاهله الحاسب تمامـــا ــ راجـــع الامر STOP حيث الامر ليس ايقاف نهائى للبرنامج بل ايقاف مؤقت.

■ **الأمر RESTORE :**

استخدام الأمر RESTORE : ويؤدى هذا الأمر الى امكانية اعادة قراءة القيم المقروءة مسبقا باعتبارها قيم لمتغيرات جديدة.

```
10      READ   A1, A2
20      PRINT  A1 + A2
30      PESTORE
40      READ   B1, B2
50      PRINT  B1 * B2
60      DATA   5,9
70      END
RUN
14
45
```

مثال :

```
10      READ  A$ , B$
20      PRINT A$ , B$
30      DATA  EIHAB , ASHRAF
```

```
40      END
RUN
 EIHAB                  ASHRAF
```

استخدام الفاصلة بين A$, B$ غير من شكل الطباعة فى حين إذا استخدمت الفاصلة المنقوطة ; بينهما لجاءت الطباعة على النحو:

```
EIHAB  ASHRAF
```

مثال :

على استخدام الامر STOP :

```
10      LET   A = 8
20      PRINT A * 5
30      STOP
40      PRINT A
50      STOP
60      STOP
70      PRINT A
80      END
RUN
STOP AT LINE  30
```

تؤدى عبارة STOP الى توقف التنفيذ فقط ويمكن اعادة الحاسب الى تنفيذ البرنامج باصدار CONT [CONTINUE] وهكذا:

```
CONT
STOP AT LINE  60
```

■ **الدوال القياسية :**

وهى اسماء انماط (روتينات) خاصة بلغة البيزك يمكن استخدامها فى صياغة البرامج بمجرد ذكرها وابرز هذه الدوال هى:

obeikandl.com

أسم الدالة	الوظيفة
SQR (X)	حساب الجذر التربيعى.
ABS	حساب القيمة المطلقة أى بدون إشارة أو كسر.
COS	حساب جيب الزاوية.
SIN	حساب جيب تمام الزاوية.
TAN	حساب ظل الزاوية.
INT	القيمة الصحيحة الموجبة والكسر مقرب لأقرب رقم صحيح.
LOG	تحسب لوغارتيم للأساس ١٠.
LIN	تحسب لوغاريتم للأساس.
SGN	تحديد إشارة المتغير.
RND	تكوين أرقام عشوائية مختلفة بقيم تتراوح بين الصفر والواحد الصحيح.

ويمكن للمبرمج [مخطط البرامج] بناء دوال خاصة به ويستطيع استدعاؤها عندما يحتاج اليها فى البرنامج وتعرف هذه الدالة بالامر DEF وهى اختصار تعريف DEFINE على النحو.

ويصدر الأمر على النحو الموضح بالإطار التالى:

النص الرياضى = اسم الدالة DEF رقم السطر

```
10      REM THIS PROG IS MULTI
```

```
20      DEF FNY  X = ( 2*X  SQR  5) + 144
30      A  = 5
40      B  = 6
60      LET  D1  =  FNY  (B)
70      LET  D2  =  FNY  (A)
80      LET  D3  =  FNY (A*B)
90      PRINT D1 , D2 , D3
91      END
```

■ **الكلمات المحجوزة RESERVED WORDS :**

وهى مجموعة من الكلمات لها دلالات وظيفية محددة فى لغة البـــيزك ولا تستخدم فى تسميه المتغيرات وتضم اوامر اللغة والدوال القياسية وبعض المعاملات ومن امثلتها:

AND , AUTO, READ , INPUT, PRINT , REM , EOF, ERASE, COS, SIN

DATA , EXP , FIELD , FILE

[للاستدلال على كل الكلمات المحجوزة يرجى الرجوع الى احد المراجع فـــى لغـــة البيزك]

مثال:

```
10      REM
20      LET PRINT = 50
30      ..........................
40      ..........................
50      END
```

SYNTAX ERROR IN 20

مثال :

```
10      REM
20      REM = 6.5
```


opelkandl.com

```
30      YEM = 7.5
40      Z = REM + YEM
50      PRINT Z
60      END

SYNTAX ERROR IN 20
```

أخطاء شائعة :

10	LET 4	عدم وجود علامة =
10	LET 4 = X	القيمة تسبق أسم المتغير
10	LET NEW = 9.5	NEW كلمة محجوزة
10	LET Y = " HODA "	عدم توافق
10	LET Y$ = ALY "	عدم وجود علامة أقتباس
10	LET FIRST NAME = "AHMED"	عدم وجود علامة$ - وجود فراغ
10	LET X = AB + 1	جمع غير صحيح
10	LET Z = 9.5.	وجود نقطة فى آخر الأمر
10	END.	وجود نقطة بعد الأمر

الاعداد لتنفيذ برامج البيزك:

١ ــ عندما يظهر المؤشر C: > يكون الحاسب جاهز لتلقى برامج البيزك أو أى اعمال اخرى.

٢ ــ لتنفيذ برامج البيزك يجب أن يكون مفسر البيزك فى ذاكره الحاسب ويتطلب هذا نقله الى الذاكرة وفق الخطوات التالية الموضحة على يسار النص علما بأن ظهور كلمة OK على الجانب الايسر العلوى

```
C:\ >
```

```
C:\ > cd \ BASIC
```

للشاشة تعنى أن مفسر
لغة البيزك أصبح جاهزا
لتلقى البرامج ويظهر على
السطر الاخير من الشاشة
وظائف مجموعة مفاتيح PF .

```
C : \ > BASIC > BASIC

O.K
.
.
.
1 LIST      2 RUN ..... ETC
```

٣ ــ ادخل البرنامج عن طريق لوحة المفاتيح.

٤ ــ يمكن استخدام الامر AUTO بعد ظهور OK لاجراء عملية ترقيم السطور آليا.

اوامر النظام:

اثناء تحميل البرنامج أو خلال تنفيذه يتطلب الامر الاتصال بنظام التشغيل لاخطاره عما يجب عمله مع البرنامج وتسمى مجموعة الاوامر المستخدمة "اوامر النظام" SYSTEM COMMANDS أبرزها الاوامر التالية:

■ **اعرض LIST :**

ويصدر على أحد الصورتين :

```
LIST    nn
LIST
```

ويعرض على شاشة الحاسب الإيعاز أو السطر المحدد برقمه مما يساعد على اجراء التصحيح اللازم نتيجة اخطاء ادخال البرنامج وقبل تنفيذه ويصدر الأمر ، أما إذا صدر الأمر LIST فقط فإن البرنامج يعرض كاملا.

obeikandi.com

■ **نفذ البرنامج RUN :**

وقد سبق وتناولناه بالشرح.

■ **اعرض EDIT :**

ويناظر عمل الامر LIST .

■ **امسح CLS :**

ويمسح كل الاسطر على الشاشة ويضع محلها OK على الجـــانب اليســار العلوى من الشاشة.

■ **جديد NEW :**

ويستخدم لمسح الإيعازات فى المنطقة العازلة داخل الذاكرة والاستعداد لتنفيذ امر جديد.

■ **خزن البرنامج SAVE :**

ويستخدم فى تخزين البرنامج على القرص المغناطيسى تحت مسمى يختاره صاحب البرنامج ، ويستخدم نفس الاسم عند استدعاء أو تحميل البرنامج:

SAVE " MNSEW"

■ **تحميل البرنامج LOAD :**

وهو مكمل للامر خزن:

LOAD " MNSEW"

■ **اعاده الترقيم RENUM :**

عند اضافة سطور إيعازات جديدة أو حذف سطور قديمة يستخدم:

RENUM

■ **النظام SYSTEM :**

للرغبة فى الخروج من البيزك والعودة الى نظام تشغيل الـــدوس يســتخدم الامر SYSTEM وبعد ادخاله تظهر العلامة: C الدالة على العودة الى نظام التشغيل.

obeikandl.com

■ الأمر AUTO :

يستخدم الأمر لترقيم الأسطر آليا.

■ شاشة BASIC :

```
C : \> BASIC
       GW BASIC
       BASICA
```

دخول COMPILER إلى الذاكرة :

CLS		أمسح الشاشة
AUTO		ترقيم ذاتى
DELETE	nn	أمسح السطر
NEW		أمسح كل ما فى الذاكرة
SYSTEM		أرجع إلى الدوس
RENUM		إعادة ترقيم
CAPS		أستخدم الحروف الكبيرة
LIST	nn	أظهر السطر رقم
HOME		حرك المؤشر

أخطاء فى أعداد برامج البيزك :

قد يواجه المبرمج بأن الحاسب قد توقف عن تنفيذ البرنـــامج وأن الحاســب أعطى رسائل أخطاء تحتمل أن تكون أخطاء لغوية أو أخطاء منطقية مثل:

أ - متغيرات غير معرفة فى البرنامج.

ب - عدم وجود رقم سطر.

ج - خطأ فى أستخدام الأقواس أثناء صياغة العمليات الحسابية.

د - عدم قدرة الحاسب على إنهاء تنفيذ البرنامج بالأمر END .

وتسمى تلك الأخطاء أخطاء فى نحو اللغة.

مثال:

```
5     REM PROG. TO CALCULATE AREA AND CIRCUMFERENCE
            OF A CIRCLE
10    PI = 3.14
20    PRINT " RADIUS = " ;
30    INPUT R
40    A = ( PI * R ↑ 2 )
50    C = 2 * P IR
60    PRINT "R =" , R , "A =" ; A , "C =" ; C
70    PRINT
80    GOTO 15
```

وعند إصدار أمر RUN فإن أخطاء البرنامج تظهر على الشكل التالى :

```
?     ILLEGAL FORMULA IN LINE 50
```

لأن النص الرياضى يجب أن يكون على النحو 2 * PI * R

obeikandl.com

? UNDEFIND LINE NUMBER 15 IN LINE 80

حيث لايوجد سطر برقم ١٥

? NO END INSTRUCTION

لايوجد أمر END وبالتالى لايمكن إنهاء البرنامج.

الأخطاء البرمجية :

تعتبر أخطاء نحو اللغة بسيطة ويمكن التوصل إليها بسهولة ويسر ، عكس ، الأخطاء المنطقية التى يصعب أكتشافها حيث أستخدم المبرمج معادلات خاطئة مع أستخدام خاطئ لأسماء المتغيرات ، ويمكن إعطاء أمثلة عن هذه الأخطاء كالآتى:

أ - ناتج العمليات الرياضية أكبر من حيز التخزين المتاح فى الذاكرة أو أن النتائج صغيرة ومتدنية للغاية ويصدر الحاسب رسالة OVER FLOW.

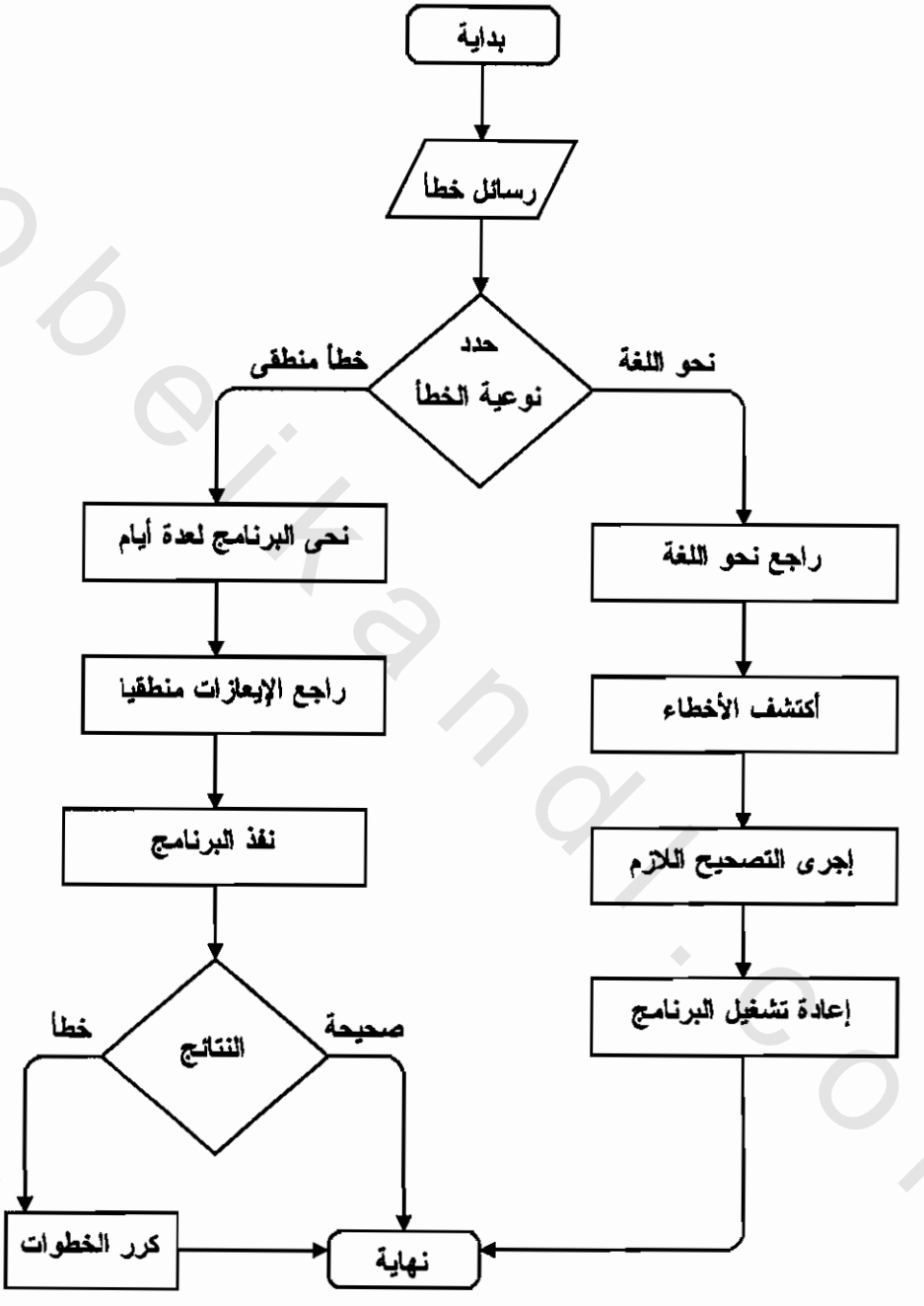
مثل : ناتج عملية قسمة [9999998 / 0.000004] القسمة على الصفر أو حساب الجذر التربيعى لقيمة سالبة $\sqrt{-١}$ وهذه قيمة تخيلية.

علاج الأخطاء :

أ - مراجعة كل إيعازات البرنامج ويفضل تنحية البرنامج جانبا لعدة أيام ثم مراجعته.

ب - مراجعة كل إيعاز منطقى وإذا فشل المبرمج فى تحديد الخطأ عليه ، تشغيل البرنامج مرة أخرى مع عدد كبير من البيانات مع أستخدام الأمر STOP على أمتداد البرنامج لإمكان مراجعة التنفيذ.

ج - إذا لم يتوصل المبرمج للخطأ يحتمل أن يكون الخطأ ناجم عن عيب فى الحاسب نفسه.



obeikandl.com

مثال : خطط برنامج لتنفيذ المعادلة التالية :

$$Y = \frac{(X-1)}{X} + \frac{(X-1)^2}{2X} + \frac{(X-1)^3}{3X} + \frac{(X-1)^4}{4X} + \frac{(X-1)^5}{5X}$$

الحل :

أ - أجرى تبسيط المعادلة على النحو التالى $U = X - 1 / X$

$$Y = U + \frac{1}{2}U^2 + \frac{1}{3}U^3 + \frac{1}{4}U^4 + \frac{1}{5}U^5$$

ب - أكتب البرنامج :

```
10      PRINT " X = " ;
20      PRINT  X
30      LET  U = X - 1 / X
40      LET  Y = U + (U/2) ↑ 2 + (U/3) ↑ 3 + (U/4) ↑ 4 + (U/5) ↑ 5
50      PRINT " Y =  " ; Y
60      END
```

ج - عند التعويض عن X بـــ (٢) تصبح النتيجة Y = ٢,٢٠٩٧١ .٢

د - إذا تم التعويض فى المعادلة الأصلية يدويا بقيمة X = ٢ فإن القيمة Y تساوى [٠ ,٦٩].

هـــ - هذا التباين فى النتائج يؤكد وجود خطأ منطقى.

و - يجرى فحص البرنامج.

ز - فى السطر ٣٠ يوجد خطأ ويجب صياغة السطر (٣٠) على النحو :

```
30      LET  U = ( X - 1 ) / X
```


obeikandl.com

ح – بإعادة تشغيل البرنامج والتعويض بقيمة X تساوى [٢] فإن قيمة Y سوف تكون [٠,٥٦٧٣٨٤] مما يؤكد وجود خطأ آخر حيث يجب أن تساوى [٠,٦٩].

ط – بمراجعة البرنامج تم أكتشاف خطأ فى السطر رقم [٤٠] ويجب صياغته على النحو:

```
40    LET Y = U + U↑2 + (U↑2) / 2 + (U↑3) / 3 + (U↑4) / 4 + (U↑5) / 5
```

ك – كرر تنفيذ البرنامج بذات قيمة X

الناتج Y = 0.688542 وتساوى تقريباً [0.69] .

رابعا: اوامر التفرغ BRANCHING:

اوامر التفرغ هى وسائل لغة البيزك لتغيير مسار تنفيذ خطوات البرنامج لرغبة مُخطط البرامج فى اجراء عمليات منطقية أو تكرار تنفيذ جزء من البرنامج ، وتضم اللغة عدة اوامر تغطى هذا المطلب على النحو التالى:

■ **التفرغ غير المشروط .**

■ **الامر .. اذهب الى (رقم السطر) GOTO nnn :**

```
10    INPUT X
20    Y = 9* X + 3*X ** 2 + 5
30    PRINT X;; Y
40    GOTO 10
50    END
```

```
RUN
?
```

ونلحظ من البرنامج مايلى:

أ ــ يبدأ تنفيذ البرنامج أول مره بالامر RUN.

ب ــ يتم ادخال قيمه X.

ج ــ يستمر التنفيذ حتى رقم ٤٠ ومن هناك يذهب مباشرة الى السطر رقم (١٠) وتظهر علامة الاستفهام ؟ طالبة ادخال قيمه X .

د ــ لن ينفذ البرنامج الخطوة فى السطر رقم ٥٠ ابدا لان الدوارة غير منتهية.

هـ ــ يتطلب وقف البرنامج تدخل المستخدم والضغط على المفاتيح ALT + CTRL+ DEL.

ويتضح من المثال السابق أن الأمر GOTO ينقل التنفيذ من المسار العادى الى سطر (ايعاز) آخر فى البرنامج دونما شروط .

مثال :

```
10    PRINT " XYZ                    - LINE # 10 "
20    GOTO 50
30    PRINT " ALI "                  - LINE # 30 "
40    GOTO 70
50    PRINT " MNS"                   - LINE # 50 "
60    GOTO 30
70    STOP
80    END
```

ناتج تنفيذ البرنامج

```
 XYZ                                 - LINE # 10
MNS                                  - LINE # 50
ALI                                  - LINE # 30
```

■ التفرغ المشروط :

١ ــ الأمر IF ....THEN GOTO :

obeikandl.com

يمكن صياغته على النحو:

```
30      IF  A = 12  GOTO 40
30      IF  A = 12  THEN  GOTO 40
```

مثال :

```
10      LET  A = 40
20      IF A = 40  GOTO 50
30      PRINT  A ** 2
40      PRINT A + 10
50      PRINT SQR A
60      END
```

٢ ــ الامر IF.... THEN :

وتعنى IF تحديد شروط لازمة لتنفيذ جواب الشرط THEN :

```
10      READ N
20      COUNT = 1
30      INPUT  X
40      Y = 9*X  +  3*X ↑ 2 + 5
50      PRINT Y
60      IF COUNT = N  THEN GOTO  90
70      COUNT = COUNT + 1
80      GOTO   30
90      END
```

السطر رقم ٢٠ عداد بدايته واحد .

السطر رقم ٣٠ اقرأ المدخلات للمتغير X .

السطر رقم ٤٠ النص الرياضى المطلوب حسابه.

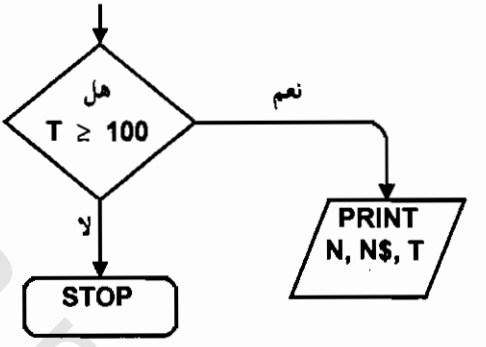
السطر رقم ٥٠ النص الشرطى الملزم بأنه اذا كانت قيمة العداد تساوى القيمة المفترضة فى N (سطر ١٠) فعلى البرنامج الخروج عن التسلسل الطبيعى وتنفيذ الايعاز بالسطر رقم (٩٠) وانهاء البرنامج وان لم يتحقق فعلية استكمال التنفيذ.

السطر رقم ٦٠ زيادة العداد فى كل دورة بمقدار واحد (الوحدة).

السطر رقم ٧٠ اذهب الى بداية البرنامج فى السطر رقم (٣٠).

مثال :

بفرض أن N رقم الطالب وأن $N أسم الطـــالب ، T مجمـــوع درجاتـــه ، والمبين أمامك جزء من خريطة تدفق أكتب قطعة البرنامج الممثلة لها ؟



الحل :

```
90      IF T ≥ 100 THEN
100     STOP
110     PRINT N, N$, T
120
```

مثال :

هل يصلح التعبير التالى للتفرغ المشروط :

```
100     IF N$ = AHMED GOTO 300
```

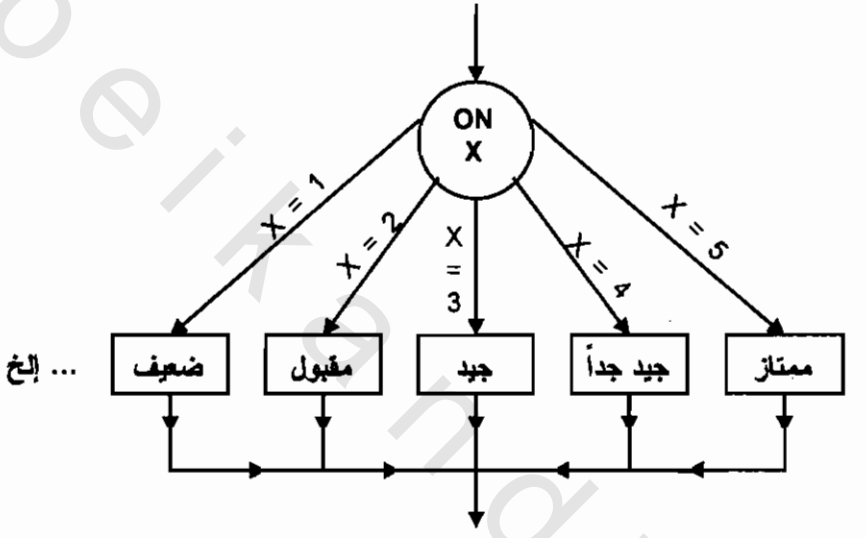
■ التفرغ المتعدد :

أثناء حل بعض المشاكل نجد أن هناك حالات كثيرة تعتمد على قيمة متغير ففى رصد التقدير التراكمى لطلبة الجامعات نجد أن إجمالى الدرجات (X) قد يفضـــى

obeikandl.com

إلى حصول الطالب على أياً من التقديرات التالية ، ممتاز – جيدا جـــداً – جيـد – مقبول – ضعيف – ضعيف جداً ، وفى لغة البيزك يوجد أمر يجرى مثل هذا التفرع المتعدد بديلاً عن أستخدام التفرع المشروط ، ونجد مثل هذا الأمر فى كـــل لغـات البرمجة.

ويمكن تمثيل التفرع المتعدد على النحو التالى:



ويجب أن تكون قيمة المتغير عدد صحيح موجب (١ ، ٢ ، ٣ ، ٤ ، ٠٠٠٠).

ويصاغ الأمر بالشكل :

```
ON   < variable >  GOTO   n1, n2, n3, n4
```

مثال :

```
ON   X   GOTO   30, 40, 50, 60, 70
```

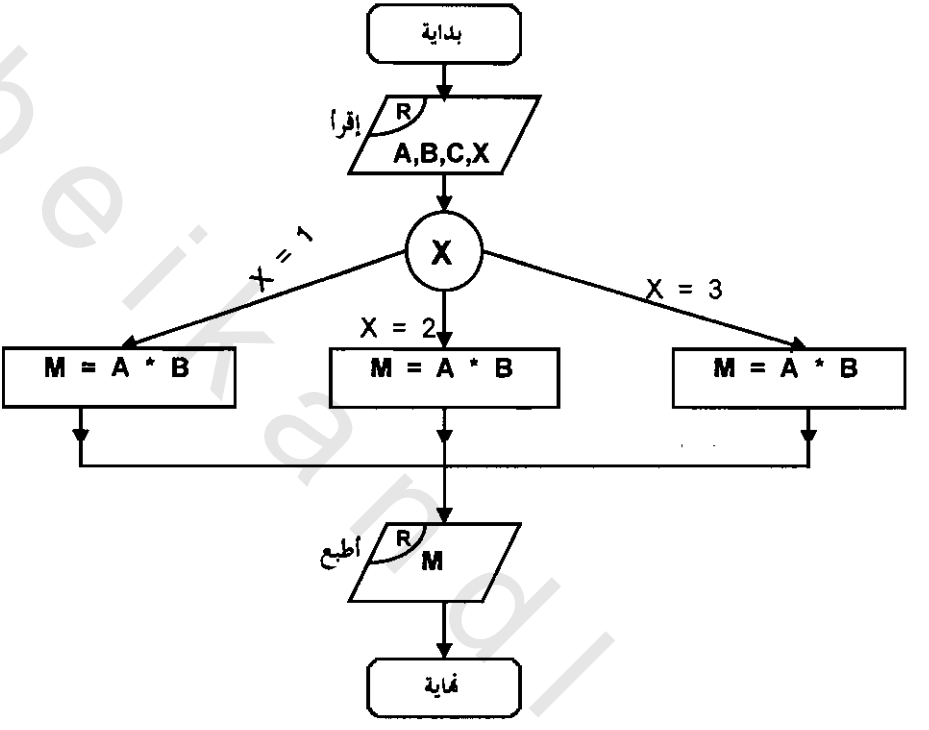
وهذا الأمر يكافئ الجمل التالية :

```
IF     K = 1 THEN GOTO 30
IF     K = 2 THEN GOTO 40
IF     K = 3 THEN GOTO 50
```

```
IF    K = 4 THEN GOTO 60
IF    K = 5 THEN GOTO 70
```

مثال :

أكتب برنامج لخريطة التدفق التالية :




```
10      INPUT A,B,C,X
20      ON X GOTO 30 , 50 , 70
30      M = A * B
40      GOTO 80
50      M  = ( A + B ) - C
60      GOTO 80
70      M = ( A - B ) + C
80      PRINT M
90      STOP
100     END
```


obeikandl.com

الدوارات LOOPS :

التكرارات :

تعتبر عملية تكرار مجموعة من إيعازات بعض البرامج خاصة الحسابية منها من أهم الأعمال التى توليها لغات البرمجة أهتماماً كبيراً ونجد لها فى لغة البيزك عدة أوامر :

- IF THEN GOTO
- FOR NEXT
- DOWHILE LOOP
- WHILE WEND

ضوابط التكرارات :

تعتمد عملية التكرار على الضوابط التالية:

أ - تحديد قيمة مبدئية لمتغير عددى أو أرقام صحيحة أو حقيقة.

ب - تحديد أقصى قيمة يمكن أن يصل إليها التكرار.

ج - تحديد معدل التزايد أو التناقص كلما بدأ تنفيذ دورة جديدة وتعنى أوامر التكرار عن ثلاثة خطوات ضرورية تتضح من المثال التالى.

مثال:

المطلوب تصميم مخطط برنامج لحساب متوسط أعمار عدد ٥٠٠ من طلبة الكلية ، ونشير إليها بالأسهم على الشكل التالى:

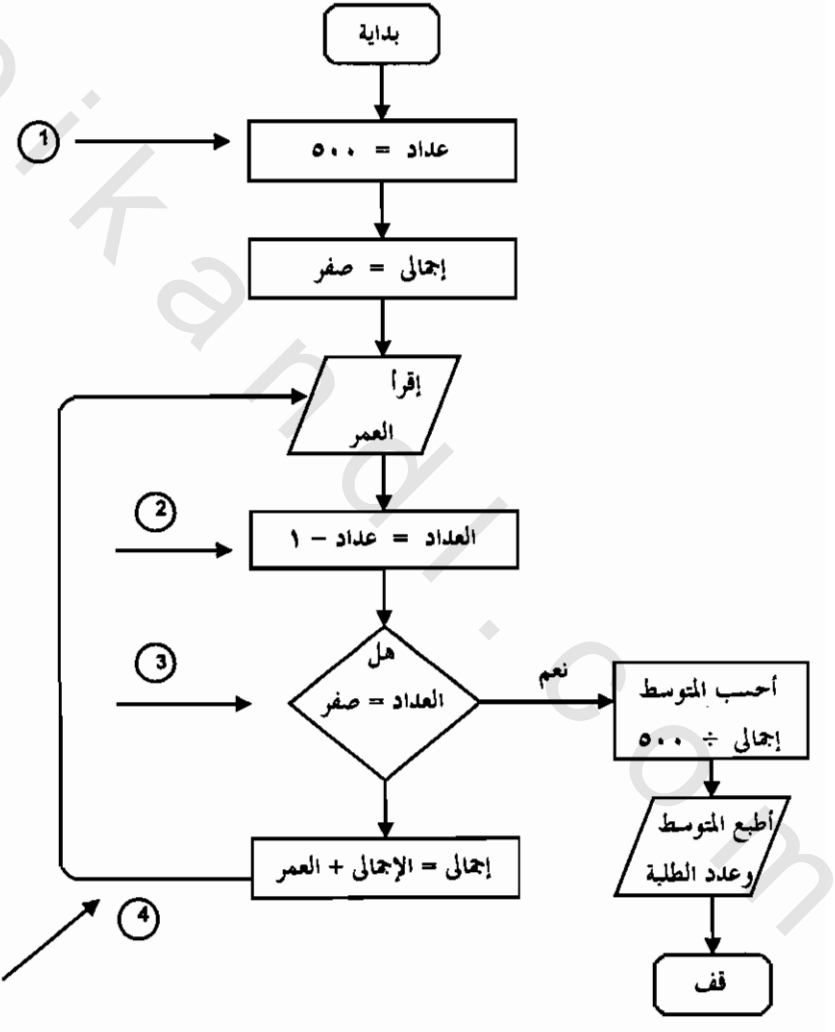
الحل :

١ - فتح عداد.

٢ - تقليل قيمة العداد مع قراءة كل بيان جديد.

٣ - إجراء مقارنة منطقية.

٤ - أستخدام الأمر GOTO .



■ امر الدوارة FOR... NEXT

يستخدم هذا الامر اذا كان عـــــدد التكرارات معروف مسبقا وينص بعد FOR على عدد التكرارات بداية مـــن ونهاية الى بمعدل تزايد أو تناقـــص

FOR X = 1 TO 500 BY 1

وهذا يعنى أن المتغير (X) تبدأ قيمته من واحد ويظل يزداد بمعدل الوحدة حتى يصل الى ٥٠٠ ، ويستمر الحاسب فى تنفيذ الإيعازات المحصورة بين FOR،NEXT حتى يتحقق الشرط الضمنى فى جمله FOR بعدها يخرج التنفيذ عن الدوارة ويستكمل خطواته بعد NEXT حتى نهاية البرنامج ويصبح شكل النص فيما يوضحه الاطار التالى والـــذى يقرأ من جهة اليسار الى جهة اليمين:

FOR	القيمة المبدائية = المتغير	TO	القيمة النهائية	BY	معدل التزايد

أسطر الدوارة

NEXT	أسم المتغير

```
10      FOR I = 1 TO 5 BY 1
10      FOR I = 50 TO 0 BY 2
10      FOR I = 1 TO 500 BY 5
```

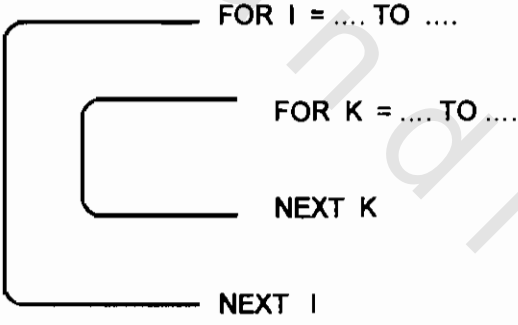
وقد يتغاضى كاتب البرنامج عن معدل الزيادة اذا كان الوحدة الموجبة.

مثال:

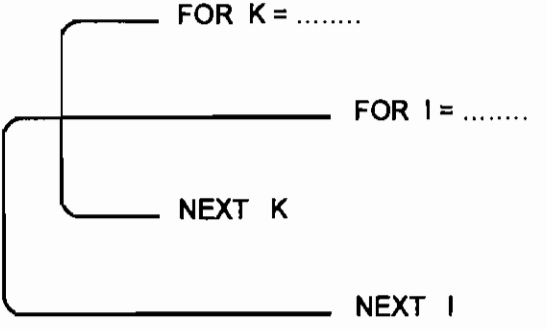
```
10      FOR I = 1 TO 50
20      READ A ,B
30      M = (A + B) / (A -B)
40      PRINT M
50      NEXT I
60      DATA 5,10,50,30,90
70      END
        RUN
```

■ **الدوارات المتداخلة NESTED LOOPS :**

يمكن من خلال لغة البيزك إجراءات دوارات داخل دوارات بشرط أن يكون أسم المتغير الداخلى مختلف عن أسم المتغير للدوارة الخارجية.



وغير مسموح بتقاطع الدوارات .

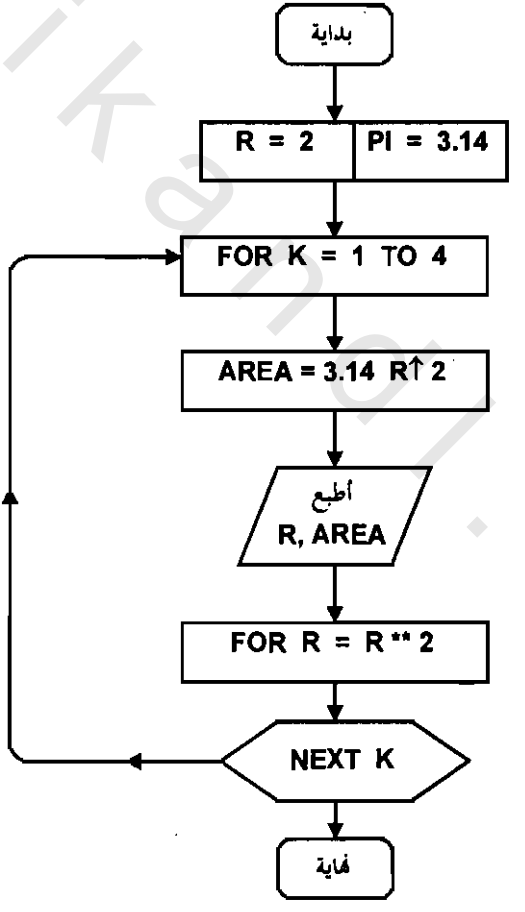


ويمكن الإنتقال من الدوارة الداخلية إلى أى أمر خارجى.

مثال:

أكتب برنامج لحساب مساحة الدوائر بأنصاف أقطار ٢ ، ٤ ، ٨ ، ٠٠٠٦ ، ٢٥٦ باستخدام الدوارة FOR ... NEXT ؟

```
10      LET R = 2
20      FOR K = 1 TO 4
30      AREA = 3.14 * R ^ 2
40      PRINT " R = " ; R , "AREA =" , A
50      LET R = R↑2
60      NEXT K
70      END
```



مثال:

اكتب برنامج بلغة البيزك لاعداد جدول ضرب الاعداد من ١ الى عشرة.

الحل:

```
10      FOR I = 1 TO 10
20      FOR K = 1 TO 10
30      LET M = I *K
40      LPRINT I , K "="; m
50      NEXT k
60      NEXT i
70      END
```

مثال:

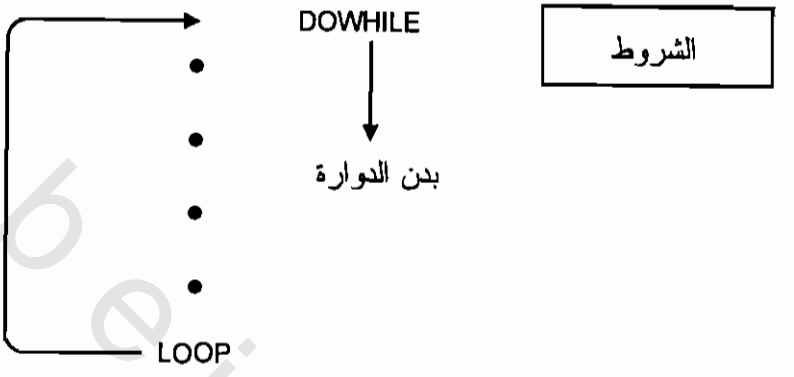
احسب حاصل ضرب ٥ ×٤×٣×٢×١ ، فيما يطلق عليه رياضيا مضروب العدد N1.

الحل:

```
10      F = 1
20      INPUT N
30      FOR I = 1 TO N
40      F = F * I
50      NEXT I
60      PRINT F
70      END
```


openkandl.com

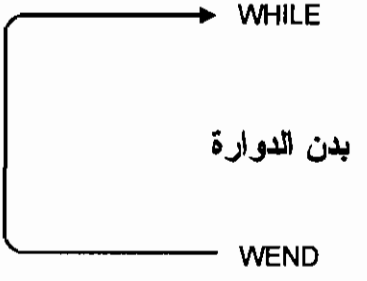
● DOWHILE LOOP :



مثال:

```
10      COUNT  = 1
20      DOWHILE  COUNT ≥  500
30      LPRINT  COUNT
40      COUNT = COUNT + 1
50      LOOP
60      END
```

● الامر WHILE ..... WEND :



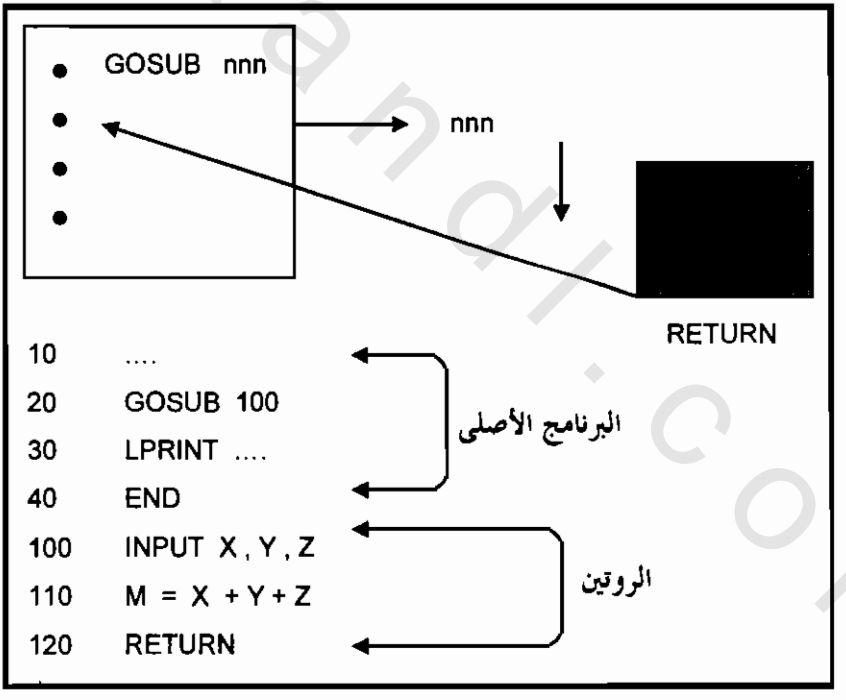

```
10      X  = 1
20      WHILE X  ≤  150
30      PRINT X
40      Y = SQR X  +  5*X  + 3
```


obeikandi.com

```
50      PRINT Y
60      WEND
70      END
RUN
```

البرامج الفرعية SUBROUTINES :

يعتبر استخدام الانماط SUBROUTINES أو الوحدات البنائيـــة MODULES احد الاساليب المتطورة فى تصميم البرامج بحيث يتكون البرنامج من عدد محـــدود من الإيعازات ويستعان بأنماط تؤدى وظائف محددة بالامر GOSUB والى جوارهـــا رقم الايعاز وعند تنفيذ النمط يبدأ التنفيذ من اول ايعاز فى النمط حتى ينتهى تنفيـــذه ويعود الى السطر التالى من الامر GOSUB بالامر RETURN .



المصفوفات ARRAYS :

يعتبر تخزين البيانات داخل ذاكرة الحاسب احد اهم الوسائل التـــى اتاحتـــها لغات البرامج بحجز مواقع تخزين بعدد البيانات المراد تخزينها لاعـــادة اســـتخدامها عند الحاجة اليها ، عكس ادخال البيانات باستخدام الامر DATA, READ أو الامـــر INPUT حيث يلغى البيان السابق عند ادخال بيان جديد فى موقع المتغيرات.

وتسمى مواقع تخزين البيانات فى الذاكرة المصفوفات ، وهـــى قـــد تكـــون مصفوفة احادية المواقع وفى اتجاه واحد وترتب البيانات داخل مواقع تخزين علـــى هيئة عمود واحد متعدد الصفوف.

المتغير X (I)	القيم المناظرة
X1	5
X2	3
X3	9
X4	10
X5	2
X6	6
X7	8

أو مصفوفة ذات بعدين.

X (I , K)		
X (1,1)	X (1,2)	X (1,3)
X (2,1)	X (2,2)	X (2,3)
X (3,1)	X (2,2)	X (3,3)
X (4,1)	X (4,2)	X (4,3)
X (5,1)	X (5,2)	X (5,3)

■ المصفوفة (القائمة) الاحادية:

يستخدم فى فتحها فى الذاكرة الامر DIM يليه اسم المصفوفة وطولها سيان كان رقمى أو هجائى.

```
60      DIM
60      DIM N ( 100 )
```

ويجب مايلى:

أ ــ اذا لم يذكر حيز المصفوفة يفتح النظام آليا عشر مواقع.

ب ــ يجب أن يكون الحيز مساو لعدد القيم فاذا استخدم حيز أقل يخطر الحاسب المستخدم بالرسالة التالية:

```
SUBSCRIPIT OUT OF RANGE
```

ج ــ اذا كان الحيز المستخدم اكبر من عدد القيم اهدرت مواقع تخزين فى الذاكرة دون داع.

■ ادخال البيانات إلى المصفوفة:

أ ــ بالامر LET :

```
10      LET A  (1) = 5
20      LET A  (2) = 7
•
•
•
900     LET A (99) = 3
1000    LET A (1000) = -3
```

ويعتبر استخدام الامر مضيعة للوقت.

ب ــ بالامر READ :

```
150     READ A (1) , A (2) , A (3) , .....A (100)
160     DATA
```

ج ــ بالامر INPUT :

```
100 INPUT A (I) , A (2) , A (3)  ..... A  (100)

.
?
```


obeikandl.com

د ـــ بالامر FOR: NEXT

فتح المصفوفة:

```
40      DIM A (100)
50      FOR I = 1 , 100
60        READ  A ( I )
70      NEXT I
80      DATA 5, 7, ....., -3, 3
```

نفس الامر يمكن استخدامه فى طباعة المصفوفة (القائمة):

```
110     FOR I = 1 , 100
120     PRINT A ( I )
130     NEXT I
```

■ **المصفوفة ذات البعدين :**

اذا كان عدد الاعمدة M وعدد الصفوف N تسمى مصفوفة N* M ويتم فتحها فى ذاكرة الحاسب وحجز مواقع القيم بالامر:

```
10      DIM X (2,3)
```

● تحميل البيانات:

```
300     FOR ROW = 1 TO  3
310     FOR COLUMN = 1 TO 2
320     READ A (ROW ,COLUMN)
330     NEXT COLUMN
340     NEXT ROW
```

● طبع المصفوفة:

```
A (1,1)    A (2,1)    A (3,1)    A (1,4)    ...
A (1,2)    A (2,2)    A (3,2)    A (2,4)    ...
A (1,3)    A (2,3)    A (3,3)    A (3,4)    ...
40      .............
50      FOR ROW
60      FOR COLUMN
```


obeikandl.com

```
70      PRINT A (ROW, COLUMN)
80      NEXT COLUMN
90      NEXT ROW
100     ..............
```

- **معالجة المصفوفات:**

(١) جمع كل عناصر المصفوفة السابقة:

```
100     LET TOTAL = 0
110     FOR ROW  = 1,3
120     FOR COLUMN = 1,5
130     TOTAL = TOTAL + A (ROW, COLUMN)
140     NEXT COLUMN
150     NEXT ROW
160     PRINT TOTAL
```

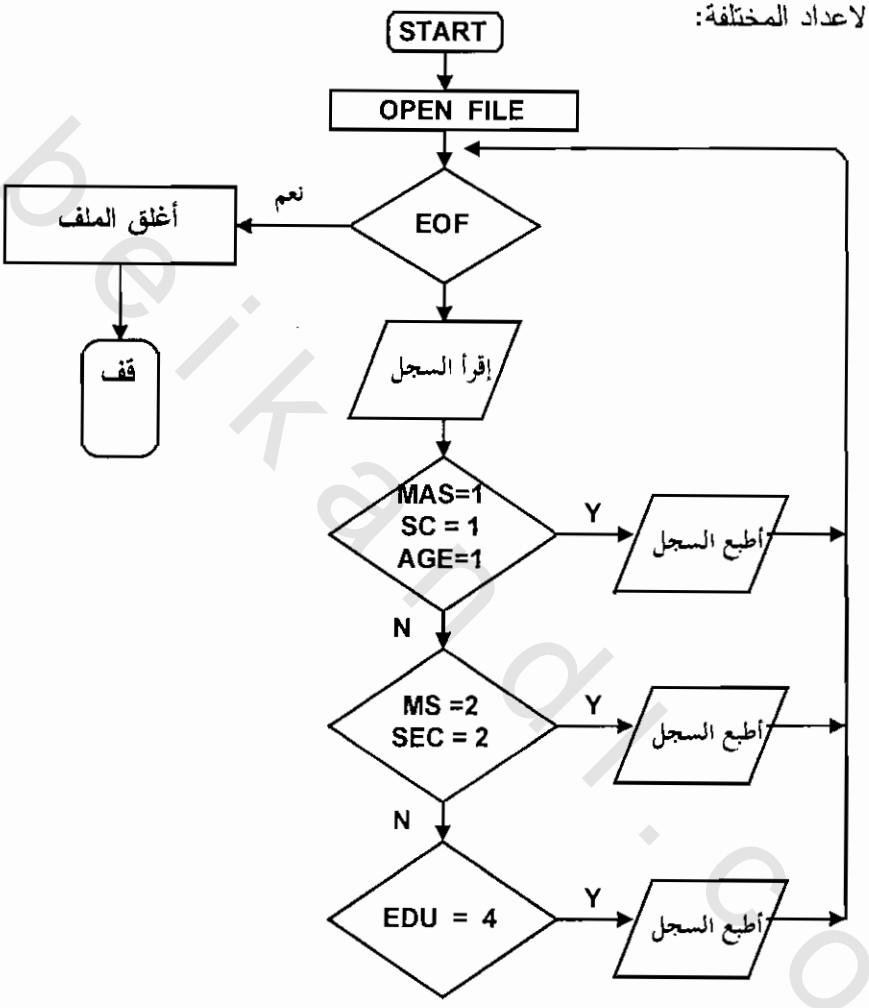
يفضل قبل ادخال بيانات أى مصفوفة التأكد من أن مواقع التخزين لا تضم أى قيم سابقة وذلك بتصفير القيم السابقة على النحو:

```
10      DIM X (4)
20      FOR X = 1,4
30        X  (I) = 0
40      NEXT X
```

```
10      FOR ROW = 1,3
20         FOR COLUMN = 1,5
30         A (ROW,COLUMN) = 0
40         NEXT COLUMN
50       NEXT ROW
```

مثال:

اكتب برنامج لايجاد اكبر عدد من بين مجموعة تحتـــوى علـــى ٢٥ مـــن الاعداد المختلفة:




```
10      REM PROGRAM TO FIND THE MAX VALUE
20      REM MAIN PROGRAM
30      DIM X (25)
40      INPUT N
```

```
50      GUSOG 90
60      GUSOG 140
70      END
80      REM SEARCHING FOR SUBROUTINES
90      FOR I = 1 OT 25
100     INPUT X (I)
110     LPRINT X (I)
120     NEXT I
130     RETURN
140     MAX = X (1)
150     FOR TO I = 2 TO 25
160     IF MAX > = X (I) THEN MAX = X (I)
170     NEXT I
180     LPRINT
190     LPRINT
200     LPRINT "MAXIMUM VALUE IS= "; MAX
210     RETURN
```

مثال:

المطلوب اعداد برنامج لحساب المتوسط والانحراف المعيارى لمجموعة من القياسات عددها ن .

الحل:

$$\text{المتوسط} = \bar{X} = \frac{\sum X}{N}$$

$$\text{الأنحراف المعيارى} = S^2 = \sum_{i=1}^{i=N} (xi - x)^2 / (N - 1)$$

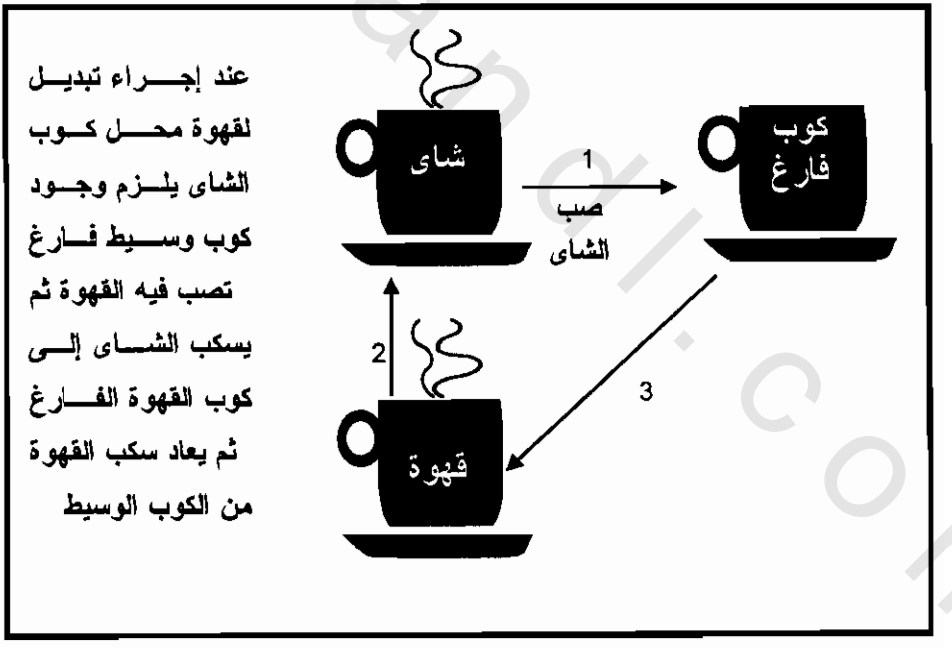
حيث $\bar{X}$ متوسط قيم الظاهرة ، N عدد هذه القيم.

```
10      DIM X (100)
20      S = 0
30      SUM = 0
40      READ N
50      FOR I = 1 TO N
60         READ X (I)
70         SUM = SUM + X (I)
```

```
80      NEXT I
90      MEAN =SUM / N
100     FOR I = 1 TO N
110     S = S + { X (1) - MEAN } ** 2
120     NEXT I
130     S = SQR ( S / (N-1) )
140     PRINT SUM, S
150     DATA 7, 8, 315, .....
160     END
```

مثال:

يراد فرز مجموعة من القيم باستخدام نظرية الكوب الفارغ (الفقاعة BUBBLE SORT) فإذا كان لدينا كوبين احدهما ممتلئ بالشاى والآخر بالقهوة ونريد أن نبدل الاكواب فاننا نلجأ الى كوب فارغ يستخدم وسيط فى التبديل.




```
10      REM BUBBLE SORT
15      DIM X (100)
```

```
20      READ N
30      FOR I = 1 TO N
40         READ X (I)                    ادخال البيانات
50      NEXT I
60      REM
70      FOR    K = 1 TO N - 1
80         FOR J = 1 TO N - K
90         IF X (J) X (1+J) THEN GOSUB 180          الفـــرز
100        NEXT J
110     NEXT K
120     REM
130     FOR I = 1 TO N
140        PRINT X (I)          الطباعـــة
150     NEXT I
160     DATA , . , .. , ...,
170     END
180     REM SORT
190     T = X (J)
200     X (J) = X (J+1)
210     X (1+J) = T
220     RETURN
```

أمثلة محلولة وتمارين :

مثال رقم (١) :

ايجاد مجموع وعدد كل من الارقام السالبة والارقام الموجبة لعـــدد ثلاثيـــن قيمة من قيم (A) .

الحل :

أ ــ افتح مجمع للقيم الموجبة بدايته صفر.

ب ــ افتح مجمع للقيم السالبة بدايته صفر.

ج ــ افتح عداد للقيم الموجبة.

د ــ افتح عداد للقيم السالبة.

هـ ــ اقرأ (A) .

و ــ قارن هل هى اكبر من صفر نعم:

(١) زود العداد للقيم الموجبة واحد.

(٢) اضف القيمة على المجمع الموجب.

استكمل الدوارة:

```
10      P = 0 : N = 0
20      COUNT-P = 0 : COUNT -N = 0
30      FOR I = 1,30
40         READ A
50         IF A O THEN GOSUB 100 ELSE
60         ELSE GOSUB 130
70      NEXT I
80      DATA
90      END
100     P = P + A
110     COUNT- P = COUNT - P + 1
120     RETURN
130     COUNT- N = COUNT - N + 1
140     N = N + A
150     RETURN
```

ملحوظة :

COUNT - P للقيم الموجبة

COUNT - N للقيم السالبة

مجمع القيم الموجبة P

مجمع القيم السالبة N

ملحوظة:

استخدم اسلوب آخر وليكن فتح عداد قيمته المبدئية ٣٠ يتناقص مع كل دورة قراءة واستخدم الامر GOTO فى صياغة البرنامج.

مثال رقم (٢):

المطلوب ايجاد عدد ومجموع القيم الفردية وعدد ومجموع القيم الزوجية لعدد ن من القيم والاجمالى الكلى للقيم.

الحل:

أ ــ افتح عداد للقيم الفردية T بدايته صفر J ومجمع للقيم الفردية SO .

ب ــ افتح عداد للقيم الزوجية K بدايته صفر K ومجمع زوجى S .

ج ــ اقرأ عدد القيم N .

د ــ اقرأ قيمه X .

هـ ــ افرض أن Y = X / 2 .

و ــ اذا كانت Y = INT (Y) تكون X عدد زوجى وبذلك تجمع القيمة على SE ويزاد العداد K بمقدار الوحدة أو تتجه للعداد والمجمع الآخر.

ز ــ اذا كانت J + K اصغر من N تعود الدوارة للخطوة (د).

ح ــ تطبع النتائج.

ط ــ النهاية.

```
10      J = 0 ; K = 0 : SE = 0 : SX
20      READ N
30      READ X
40      Y = X / 2
50      IF Y = INT (Y) GOTO  90
60      SO = SO+ X
70      J =  J  +  1
80      GOTO  110
90      SE = SE + X
100     K =  K  +  1
110     IF J  + K  N  GOTO  30
120     SUM  =  SO + SE
130     PRINT J, K ,SO, SE, SUM
140     DATA  ,    , , ,
150     END
```

مثال رقم (٣):

اكتب برنامج يحسب ارباح قدر من المال مودع فى البنك بسعر فائده مركبة.

حيث P = قيمة المبالغ الأساسية.

obeikandl.com

I = سعر الفائدة.

N = عدد السنوات.

F = الربح.

M = الإجمالى

```
10      FOR = 1 TO N
20      F = (1+I) ** N
30      M = F . P
40      PRINT F, P, N, M
50      NEXT N
60      END
```

مثال رقم (٤):

اكتب برنامج يحسب القيم المناظرة لـ Y للمتغير X فى المعادلة:

$$Y = 4X^{**}2 + 3 * X + 2$$

فى المجال فيما بين X = -2 TO 2

```
1       CLS
10      FOR X = 2 TO -2 BY 0.1
20      Y = 4 * X ** 2 + 3* X + 2
30      PRINT X,Y
40      NEXT X
50      END
```

مثال رقم (٥) :

أحسب ناتج خطوات هذا البرنامج :

```
10    K = 5
20    READ L,M
30    L = L + M
40    N = M / S + K
50    PRINT L , N
70    STOP
80    DATA 4,20
90    END
```

مثال رقم (٦) :

أحسب ناتج هذا البرنامج :

```
10    LET I = 1
```


obeikandl.com

```
20    PRINT I, I * I
30    LET I = I + 1
40    IF I = 10 THEN 20
50    END
```

مثال رقم (٧) :

حدد أخطاء أسماء المتغيرات والتعبيرات التالية :

```
BASIC / 32D / X1 / Z29 / $AH / 32D / READ REM / $3 / -371 + /
LET NAME = M.N.SWILAM
LET A$ = 10
LET X = YAND Y = Y + Z
LET SUM = SUM + B          READ 1X
LET A + B = C]    [PRINT X AND S
LET J = J + .N$]    [LET S$ = 1
```

مثال رقم (٨):

لديك البيانات التالية أجرى عليها الآتى:

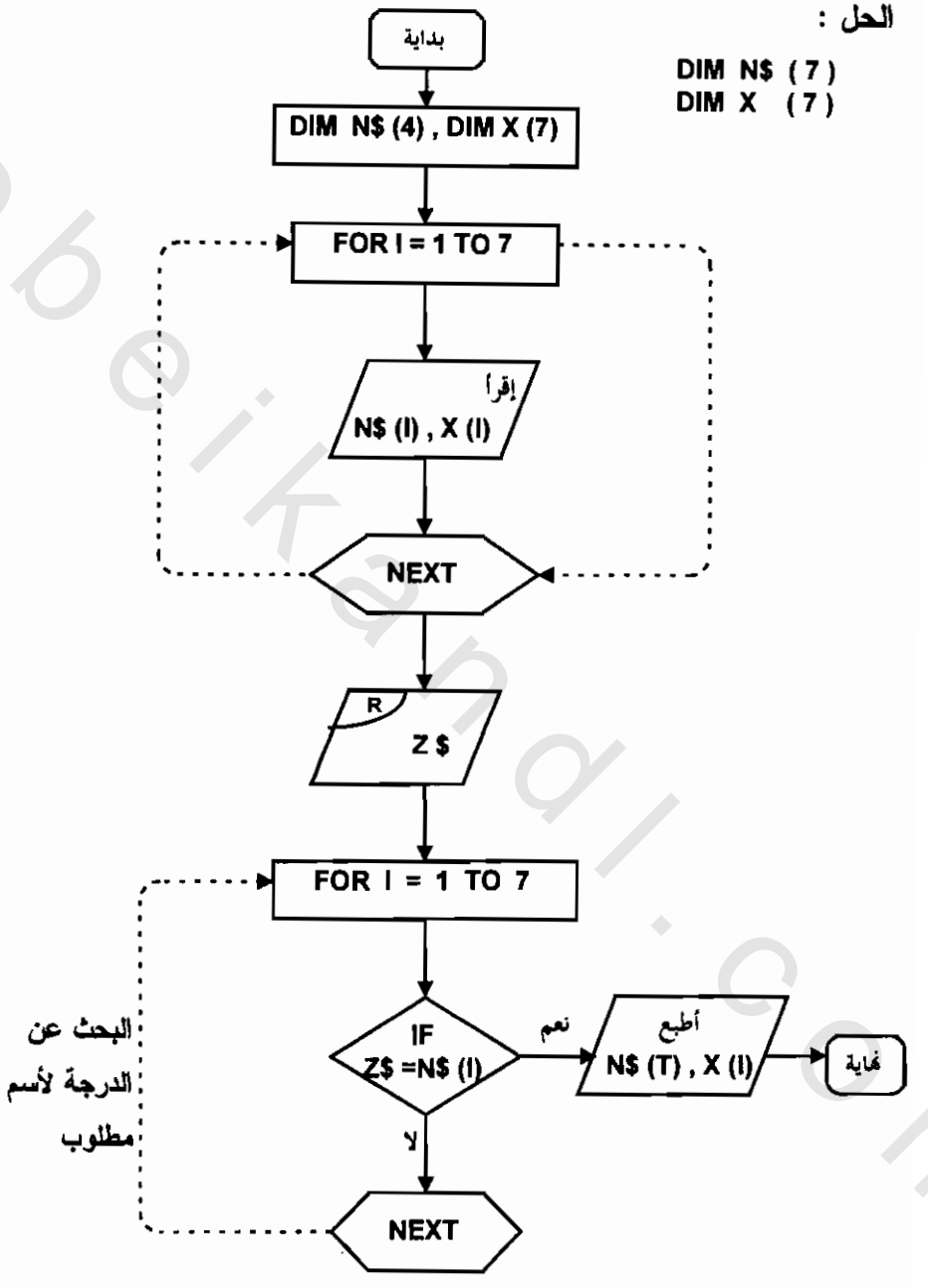
الأسم	الدرجة	
ش	٧٥	
ب	٦٠	
م	٥٠	
أ	٨٠	
أ س	٦٥	
س هـ	٤٥	
م ن	٦٠	

أ - إقرأ البيانات وخزنها فى مصفوفة N$ للأسماء ، X$ للدرجات.

ب - أبحث عن الدرجات لأى أسم.

الحل :

```
DIM N$ ( 7 )
DIM X  ( 7 )
```



البرنامج :

```
10    DIM N$(4),X(4)
20    FOR I = 1 TO 4
30    INPUT N$(I),X(I)
40    NEXT I
50    PRINT " أدخل الأسم المطلوب "
60    INPUT Z$
70    FOR I = 1 TO 4
80    IF Z$ = N$(I) THEN PRINT N$(I),X(I)
90    NEXT I
100   END
```

مثال رقم (٩):

أحسب المدى لمجموعة من الأرقام هى ١٣ ، ١٨ ، ١٦ ، ١٥ ، ١٢ ؟

الحل :

أسلوب حساب المدى :

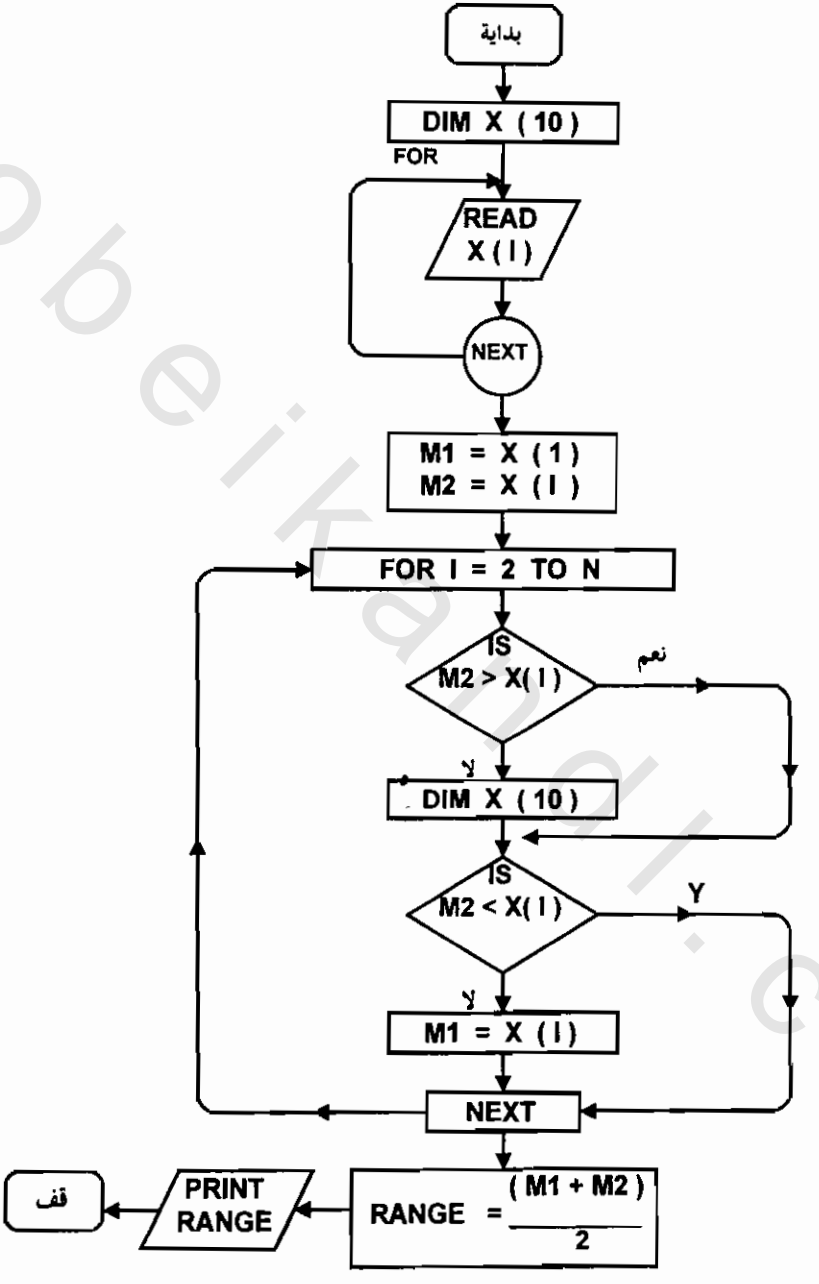
$$\text{المدى} = \frac{\text{أكبر قيمة} + \text{أصغر قيمة}}{٢}$$

ولذلك يجب على البرنامج مايلى:

أ - ترتيب الأرقام تصاعديا أو تنازليا.

ب - تعيين القيمة الأكبر والقيمة الأصغر.

ج - حساب المدى.



obeikandl.com

```
10      DIM X ( 10 )
20      READ N
30      FOR I = 1 TO 10
40      READ X ( I )
50      NEXT I
60      LET M1 = X ( 1 )
70      LET M2 = X ( 1 )
80      FOR I = 2 TO 10
90      IF M1 > X ( I ) THEN 110
100     LET M1 = X ( I )
110     IF M2 < X ( I ) THEN 130
120     LET M2 = X ( I )
130     NEXT I
140     LET RANGE = ( M1 + M2 ) / 2
150     PRINT " RANGE = " RANGE
160     DATA , , , , , , , , , , , ,
170     END
```

مثال رقم (١٠) :

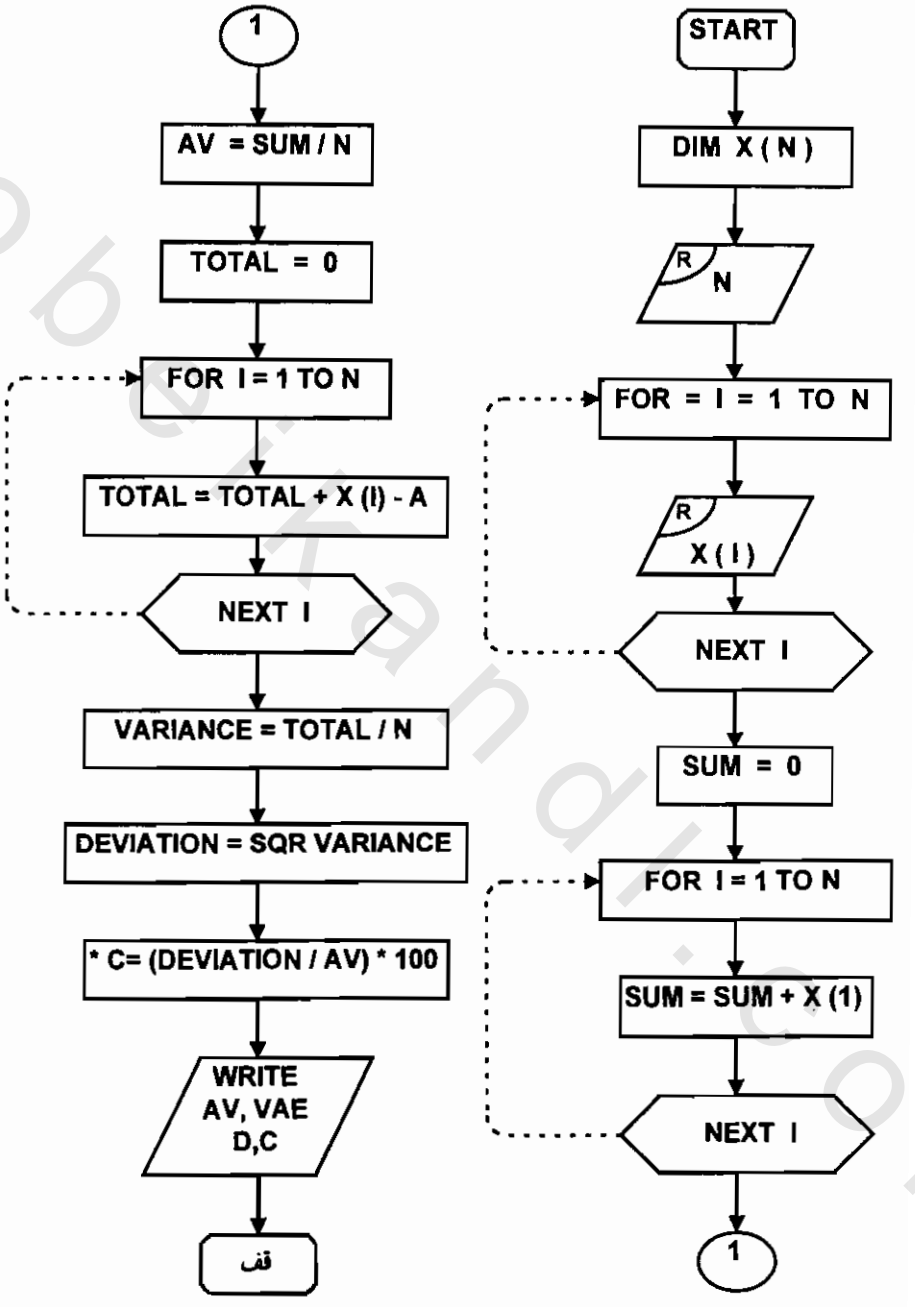
أكتب برنامج لتقدير وحساب المتوسط الحسابى ، التباين ، الأنحراف المعيارى ، معامل الأختلاف لمجموعة من القيم X عددها N علما بأن:

$$\text{المتوسط} \quad \bar{X} = \sum_{I=1}^{I=N} \Big/ N$$

$$\text{VARIANCE} \quad V = \frac{1}{N} * \left(\sum_{I=1}^{I=N} X(I) - \bar{X} \right)^2 \quad \text{التباين}$$

$$\text{الأنحراف المعيارى} \quad D = \text{SQR } V$$

$$\text{معامل الأختلاف} \quad C = \frac{D}{X} \times 100$$



* C = COFFICIENT OF VARIATION

```
10      DIM X ( N )
20      INPUT N
30      FOR I = 1 TO N
40      INPUT X ( 1 )
50      NEXT I
60      SUM = 0
70      FOR I = 1 TO N
80      SUM = SUM + X ( 1 )
90      NEXT I
100     AV = SUM / N
110     TOTAL = 0
120     FOR I = 1 TO N
130     TOTAL = TOTAL + ( X ( 1 ) - AV ) ↑ 2
140     NEXT I
150     VARIANCE = TOTAL / N
160     DEVIATION = SQR VARIANCE
170     COFFICIENT = ( DAVIATION / AV ) * 100
180     LPRINT
190     LPRINT "      THE RESULTS       "
200     LPRINT "  THE AVERAGE = " ; AV
210     LPRINT "  STANDARD DEVIATION = " ; DEVIATION
220     LPRINT "  THE VARIANCE    = ;  VARIANCE
230     LPRINT "  THE COFF OF VARIATION " = " ; COFFICIENT
240     STOP
250     END
```

الملفات :

عندما يتعامل الحاسب مع البيانات فإنه لايتعامل إلا مع السجل ، والسجل فى حد ذاته عبارة عن مجموعة مرتبطة منطقيا من البيانات [حقول البيانات] تعبر عن كيان واحد ومجموعة السجلات المتكررة هى الملف الذى يخزن سجلات عن كيان واحد متكرر والفرق بين الملفات وهياكل البيانات مثل المصفوفات فرق جوهرى فالملفات عبارة عن تخزين دائم للبيانات على وسائط التخزين الثانوية بينما هياكل البيانات تخزين مؤقت فى ذاكرة الحاسب ، وفى كلا الحالتين تستطيع البرامج التعامل مع البيانات وإجراء المعالجات التالية :

* إنشاء ملف جديد.
* دمج ملفين أو أكثر فى ملف واحد.
* إضافة سجلات.
* حذف سجلات.
* تحديث البيانات.

ويمكن تنظيم الملفات على الحاسبات الشخصية بأحد الأسلوبين :

(١) تنظيم مسلسل.

(٢) تنظيم مباشر.

ويرتكن التنظيم المسلسل على حقل مفتاح لا يتكرر فى باقى السجلات مثل رقم الطالب ــ رقم العامل ــ رقم قطعة الغيار ، ويتم فرز السجلات تصاعديا أو تنازليا على قيمة هذا الحقل ويطبق فى إسترجاع هذه السجلات القاعدة ، من سجل أولا يسترجع أولا ، ويتطلب ترتيب هذا النوع من الملفات مايلى:

(أ) أن ترتب السجلات التى سيتم إضافتها تسلسليا كما فى الملف الأصلى.

(ب) ينشأ ملف بيانات جديد.

وفى الملفات ذات التنظيم المباشر يمكن إسترجاع أى سجل مباشرة دون البحث فى باقى السجلات كما أن إضافة سجلات جديدة لا يستدعى إنشاء ملف جديد على وسيط تخزين آخر.

■ الملفات المسلسلة SEQUENTIAL FILES :

يجرى إنشاء الملف وفق الخطوات التالية:

(١) أفتح الملف بالأمر OPEN على النحو :

OPEN < أسم الملف > FOR < OUTPUT > AS # nn
< >

obeikandl.com

وتعنى كلمة OUTPUT أن الملف سيتم الكتابة عليه ، أما الرقم 1 # AS فهو رقم يقترحه منشأ الملف لسهولة أستدعائه بديلا عن أستخدام أسم الملف.

```
OPEN " INVEN " FOR OUTPUT AS $ 1
```

(٢) للكتابة على الملف أى تسجيل سجلات عليه يستخدم الأمر PRINT على النحو :

```
PRINT # 1 , NUMBER ; QUATITY
```

(٣) لقراءة أى سجل من الملف يستخدم الأمر :

```
READ NUMBER , QUANTITY
```

(٤) يستخدم أمر الدوارة WEND ..... WHILE لكتابة السجلات المناسبة مثل:

```
120     WHILE ITEM > 0
130     PRINT #1 , ITEM , QUANTITY
140     READ ITEM , QUANTITY
150     WEND
```

(٥) أغلق الملف بعد إدخال السجلات بالأمر CLOSE # 1 وهذا الأمر يسبب قيام نظام التشغيل بوضع نهاية للملف EOF بعد آخر سجل تم تسجيله.

مثال :

```
10      REM CREATE SEQUENTIAL FILE
20      OPEN " INVEN " FOR OUTPUT AS # 1
30      READ ITEM , QUANTITY
40      WHILE ITEM > 0
50      PRINT #1 , ITEM , QUANTITY
60      READ ITEM , QUANTITY
70      WEND
80      CLOSE # 1
DATA    5 , 2 , 14 , 4 , 15 , 7 , 0 , 0
```

إسترجاع البيانات من الملف المتتالى :

يتم إسترجاع البيانات من الملف وفق الخطوات التالية :

(أ) يستخدم الأمر OPEN افتح الملف لمدخلات ملف جديد :

```
OPEN " INVEN " FOR INPUT AS # 3
```

(ب) لقراءة سجل من على الملف رقم ٣ يستخدم الأمر INPUT :

```
INPUT # 3 , ITEM , QUANTITY
```

(ج) يستخدم الأمر WEND ..... WHILE لتكرار الخطوة (ب) حتى نهاية الملف.

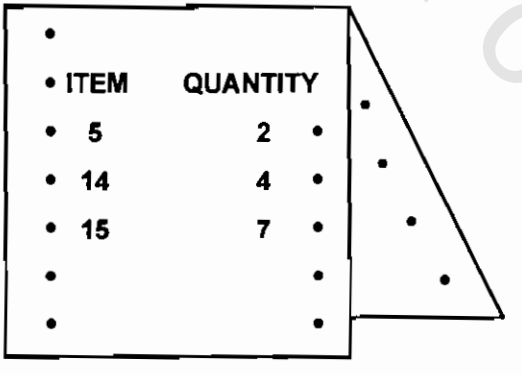
```
10      WHILE NOT EOF ( 3 )
20      INPUT # 3 . ITEM , QUANTITY
30      PRINT ITEM ; QUANTITY
40      WEND
```

(د) أغلق الملف بالأمر CLOSE # 3 .

البرنامج الكامل :

```
10      REM READ SEQUENTIAL FILE
20      OPEN " INVEN " FOR INPUT AS # 3
30      INPUT # 3 . ITEM , QUANTITY
40      PRINT " ITEM " , " QUANTITY "
50      WHILE NOT EOF ( 3 )
60      INPUT # 3 ITEM , QUANTITY
70      PRINT TAB 5 ITEM , TAB ( 20 ) QUANTITY
80      WEND
90      CLOSE # 3
100     END
```

النتيجة :



موجز أوامر التعامل مع الملفات المسلسلة :

الأمر	التفسير
OUTPUT	يحدد أن الملف جديد وينشأ لأول مرة
INPUT	يحدد أن الملف سوف تجرى قراءته
APPEND	إضافة سجلات جديدة بعد آخر سجل على الملف
# FILE	لتمييز ما هو الملف ولسهولة أستخدامه
أسم الملف	لتحديد أسم للملف مع أعتبار الأمتداد راجع DOS
WRITE	لكتابة سجلات على الملف

"راجع نسخة لغة البيزك التى تستخدمها الشركة المنتجة لها "

■ الملفات العشوائية RANDOM FILES :

يعرف ملف البيانات العشوائى بأنه عبارة عن مجموعة من السجلات ذات طول ثابت وكل سجل يتكون من مجموعة من حقول البيانات التى تخزن فى السجل كسلاسل حرفية. فيتم تحويل حقول البيانات التى تمثل قيما عددية إلى أعداد فعلية عند نقلها لذاكرة الحاسب ثم يتم تحويل هذه البيانات مرة أخرى إلى سلاسل حرفية قبل نقلها إلى الملف.

ويتم أستخدام الحاسب مخزن مؤقت أو وسيط BUFFER يحتوى على سجل واحد فى ذاكرة الحاسب ويقسم هذا المخزن المؤقت إلى حقول حيث يحتوى كل حقل على بيان واحد.

مزايا الملفات العشوائية RANDOM ACCESS FILES:

البرامج التى تستخدم فى بناء الملفات العشوائية وإسترجاع البيانات معها تتكون من خطوات أكثر تعقيداً من البرامج المستخدمة مع الملفات المتتابعة ولكن للملفات العشوائية مميزات كثيرة نذكر منها:

١ - الملفات العشوائية تتطلب مساحات تخزين على القرص أقل من الملفات المتتابعة وذلك لأن الأعداد فى الملف العشوائى تخزن على القرص فى الشكل الثنائى BINARY FORMAT بينما تخزن الأعداد فى الملف المتتابع بنظام التكويد ASCII .

٢ - يمكن إسترجاع البيانات عشوائياً من أى مكان على القرص بدون الحاجة إلى قراءة سجلات الملف بدءاً من السجل الأول وحتى نصل إلى السجل المطلوب كما هو متبع وضرورى فى الملفات المتتابعة.

٣ - زمن تداول المعلومات فى الملف العشوائى أقل من زمن تداول المعلومات من الملف المتتابع ، وذلك لأن البيانات تخزن فى وحدات مستقلة تسمى السجلات وكل سجل له رقم فريد خاص بهUNIQUE NUMBER أو مفتاح KEY.

والجدول التالى يوضح الفرق بين الملفات المتتابعة والملفات العشوائية:

	الملفات المتتابعة	الملفات العشوائية
١ - البرامج اللازمة لتشغيل الملفات.	سهلة الكتابة EASY TO WRITE .	أكثر تعقيداً COMPLEX.
٢ - طريقة التداول ACCESSING MODE.	طريقة تتابعية SEQUENTIAL .	يمكن إسترجاع البيانات عشوائياً أى بأى ترتيب ANY ORDER .
٣ - زمن التداول ACCESSING TIME.	أطول LONGER .	أقصر SHORTER .

إنشاء ملف عشوائى CREATING A RANDOM FILE :

توضح خطوات البرنامج التالى مثالاً لبناء ملف عشوائى يحتوى على بيانات طلبة فى إحدى المدارس وتشمل رقم الطالب وأسمه وتاريخ ميلاده .

وسيرمز لرقم الطالب بالرمز S$.

ولأسم الطالب بالرمز N$.

ولتاريخ ميلاد الطالب D$.

```
10    REM CREATION OF STUDENTS - FILES
20    OPEN "R" , #1, "STUDENTS" , 34
30    FIELD #1, 4 AS S$, 20 AS N$, 10 AS D$
40    INPUT "STUDENT. NO :" ; X
50    IF X = 99 THEN COLSE : END
60    INPUT "NAME" ; Y$
70    INPUT "DATE OF BIRTH" : Z$
80    LSET S$ = MKI$ (X)
90    LEST N$ = Y$
100   LSET D$ = Z$
110   PUT #1, X
120   GOTO 40
```

وعند إعطاء أمر التنفيذ RUN يعمل البرنامج كمايلى:

```
STUDENT NO .: 1
NAME: SOHER
DATE OF BIRTH: 26/6/60
STUDENT NO .: 2
NAME: ZENAB
DATE OF BIRTH: 5/9/71
STUDENT NO .: 5
NAME: MOHAMED SOWEL
DATE OF BIRTH: 27/5/76
STUDENT NO .: 20
NAME: MANAL
DATE OF BIRTH: 21/8/82
STUDENT NO .: 29
NAME: ADEL
DATE OF BIRTH: 10/10/86
STUDENT NO .: 99
OK
```

```
1  SOHER 26 / 6 / 60
2  ZENAB 5 / 9 / 71
.
.
5  MOHAMED SOWEL 27/ 5 /76
.
.
20 MANAL 21 / 8 / 82
.
.
29 ADEL 10 / 10 / 86
.
.
99
```

ومن دراسة خطوات هذا البرنامج يمكن التعرف على الأوامر المستخدمة لبناء الملف العشوائى ويتم تلخيصها فيمايلى:

١ - **فتح الملف OPEN:**

قبل البدء فى أستخدام أى ملف عشوائى وتنفيذ أى عمليات خاصة به سواء كانت قراءة سجلات منه أو كتابة سجلات فيه يجب فتح الملف بأستخدام الأمر OPEN والصورة العامة له هى :

OPEN "R" , # FILE - NUMBER , FILE - NAME , RECORD - LENGTH.

حيث (R) تشير إلى أن التعامل مع الملف سيكون عشوائياً ويستخدم لفتح الملف سواء لإسترجاع بيانات منه أو الكتابة فيه ، (# FILE NUMBER) هو رقم الملف ويمثل خط إتصال مفتوح "أحد المخازن المؤقتة فى ذاكرة الحاسب (BUFFER) " لتبادل البيانات بين الذاكرة الداخلية والمخزن الخارجى وتأخذ الأرقام من ١ إلى ١٥ ، (FILE NAME) أسم الملف ويتكون من مجموعة من الحروف والأرقام لا يزيد عددها عن ٨ على أن يبدأ الأسم بحرف كما يمكن إضافة أمتداد (EXTENSION) لا يزيد عن ٣ حروف وهو إختيارى ،) (RECORD LENGTH ويحدد طول السجل الواحد فى الملف وإذا لم يتم تحديد طول السجل فى جملة فتح الملف فإنه يعتبر ١٢٨ بايت ، وفى أمر فتح الملف فى السطر رقم ٢٠ من البرنامج السابق (بيانات الطلبة) نلاحظ :

(أ) إستخدام R لأن الملف مفتوح إدخال بيانات للكتابة فيه.

(ب) إستخدام (#1) كرقم للملف.

(ج) إستخدام الأسم "STUDENTS" كأسم للملف.

(د) طول السجل ٣٤ بايت.

٢ - **توصيف الحقول FIELDS :**

يتم حجز مخزن وسيط RANDOM BUFFER يحتوى على سجل واحد فى ذاكرة الحاسب ووظيفة الأمر FIELD هو تقسيم المخزن الوسيط إلى حقول بحيث

يحتوى كل حقل على بيان واحد من مجموعة بيانات السجل ، وجملـــة FIELD فى السطر ٣٠ من البرنامج السابق توضح أن السجل يتكون من ٣ حقول وهى:

رقم الطالب وطول الحقل المخصص له = ٤ بايت

أسم الطالب وطول الحقل المخصص له = ٢٠ بايت

تاريخ ميلاد الطالب وطول الحقل المخصص له = ١٠ بايت

ونلاحظ أن مجموع أطوال هذه الحقول الثلاثة يجب أن يساوى طـــول الســـجل وهو ٣٤ المبين فى جملة فتح الملف.

ملاحظات هامة :

١ - يجب عدم أستخدام الأسماء المستخدمة فى جملة FIELD كأسماء لمتغيرات فى داخل البرنامج فى جمل مثل جملة INPUT لأن هذه الأسماء مخصصة لحقـــول المخزن الوسيط فقط.

٢ - يمكن تنفيذ أى عدد من جمل FIELD مع الملف الواحد.

٣ - إدخال البيانات INPUT :

بعد نسخ الملف وتوصيف الحقول يتم إدخال البيانات بأستخدام لوحـــة المفـــاتيح وفى البرنامج السابق يتم ذلك من خلال السطور ٤٠ ، ٦٠ ، ٧٠.

٤ - التحويل والنقل :

حيث أنه فى الملفات العشوائية يخزن كل حقل من حقول البيانات فـــى الســـجل كسلسلة حرفية STRING فلذلك يجب تحويل حقول البيانات التـــى تمثـــل قيمـــا عددية إلى سلاسل حرفية قبل نقلها إلى المخزن الوسيط :

(أ) التحويل بأستخدام الدوال MKI$ أو MKS$ أو MKD$:

ويتم التحويل بأستخدام الدوال MKI$ أو MKS$ أو MKD$ ووظيفتها تحويل القيم العددية إلى سلاسل حرفية.

فتستخدم الدالة MKI$ لتحويل العدد الصحيح INTEGER إلى سلسلة حرفية مكونة من ٢ بايت.

وتستخدم الدالة MKS$ لتحويل العدد الحقيقى REAL NUMBER OR SINGLE - PRECISION NUMBER إلى سلسلة حرفية مكونة من ٤ بايت.

وتستخدم الدالة MKD$ لتحويل العدد المزدوج الدقة - DOUBLE PRECISION - NUMBER إلى سلسلة حرفية مكونة من ٨ بايت.

(ب) النقل إلى المخزن الوسيط بأستخدام جملة LSET أو جملة RSET :

بعد تحويل القيم العددية إلى سلاسل حرفية يستخدم الأمر LSET أو الأمر RSET فى تحريك أو نقل البيانات من الذاكرة إلى المخزن الوسيط العشوائى RANDOM FILE BUFFER أستعداداً لكتابتها فى الملف بأستخدام الأمر PUT # .

والأسطر ٨٠ ، ٩٠ ، ١٠٠ من البرنامج السابق توضح عمليتى التحويل والنقل ، ففى السطر ٨٠ يتم تحويل رقم الطالب (X) وهو عدد صحيح إلى سلسلة حرفية بواسطة الدالة MKI$ ثم يخزن فى الحقل S$ فى المخزن الوسيط بواسطة الأمر LSET (LEFT - JUSTIFIES) ، أو الأمر (RIGHT - JUSTIFIES) RSET ، ووظيفة كل منهما نقل السلسلة الحرفية من الذاكرة وتخزينها فى حقل المخزن الوسيط وإذا كان طول السلسلة الحرفية أصغر من طول الحقل المميز لها فتكمل بمسافات خالية ، أما إذا كان طول السلسلة الحرفية أكبر من طول الحقل المحجوز لها فتهمل الحروف الزائدة من جهة اليمين.

وفى السطر ٩٠ يتم نقل أسم الطالب وهو سلسلة حرفية إلى الحقل D$ فى المخزن الوسيط بواسطة الأمر LSET .

٥ - كتابة البيانات فى الملف PUT :

بعد إدخال بيانات السجل فى المخزن الوسيط يتم كتابتها فى الملف العشوائى بأستخدام الأمر #PUT والصورة العامة له هى :

PUT (#) FILE - NUMBER , (RECORD NUMBER)

ونجد هذا الأمر فى السطر ١١٠ من البرنامج السابق فى الصورة :

PUT # 1, X

(أ) تم أستخدام نفس رقم الملف (1 #) المستخدم فى جملة فتح الملف.

(ب) الرمز X وهو رقم الطالب الذى تم إدخاله فى جملة INPUT فى السطر رقم ٤٠ والذى سيعتبر رقما للسجل.

وإذا تم حذف RECORD - NUMBER من جملة PUT # فإن السجل الحالى يعطى رقما أوتوماتيكياً يزيد عن رقم السجل السابق له بمقدار ١.

وفى كل مرة تنفذ جملة PUT يتم كتابة سجل فى الملف ويأخذ السجل رقم الطالب الذى تم إدخاله فى السطر رقم ٤٠ بواسطة أمر INPUT .

٦ - إغلاق الملف CLOSE :

عند الإنتهاء من كتابة البيانات المدخلة فى الملف يتم إغلاق الملف بأستخدام الأمر CLOSE الذى ينهى الأتصال بين الملف وذاكرة الحاسب وتكون البيانات التى أدخلت قد تم حفظها على القرص.

والصورة العامة لأمر CLOSE هى :

CLOSE (# LIST OF FILE - NUMBERS)

وتتبع كلمة CLOSE العلامة (#) ثم قائمة بأرقام الملفات التى قد تكون رقم ملف أو أكثر والقائمة أختيارية.

ملاحظة :

إذا لم يتم إدخال أى أرقام ملفات بعد CLOSE يتم إغلاق جميع الملفات التى سبق فتحها ، وعلى ذلك عندما يكون المطلوب إغلاق عدد معين من الملفات

المفتوحة وليس كلها يجب أن يلى أمر CLOSE أرقام هذه الملفات ، وعند تنفيذ البرنامج يكون قد تم بناء الملف STUDENTS .

إسترجاع (قراءة البيانات من الملف)

ACCESSING A RANDOM FILE :

المقصود بإسترجاع البيانات هو نقل البيانات من الملف الموجود فى المخزن المساعد إلى الذاكرة الداخلية والتعامل مع هذه البيانات ، ويمكن توضيح ذلك عن طريق البرنامج التالى الذى يقوم بإسترجاع البيانات التى تم إدخالها فى الملف السابق (بيانات الطلبة):

```
10      REM READ DATA FROM A FLILE
20      OPEN "R", #1 "STUDENTS", 34
30      FIELD #1, 4 AS S$, 20 AS N$, 10 AS D$
40      PRINT TAB(4); "STUDENT NO"; TAB(25); "NAME";
50      PRINT TAB(40); "BIRTH DATE"
60      PRINT
70      INPUT "STUDENT NO."; X
80      IF X = 99 THEN CLOSE #1 : END
90      GET #1, X
100     PRINT
110     PRINT TAB(4); CVI (S$); TAB(25); N$; TAB(40); D$
120     GOTO 70
```

ومن دراسة هذا البرنامج يمكن التعرف على الخطوات التى تتبع لإسترجاع بيانات من ملف عشوائى والتى يتم تلخيصها فيمايلى:

١ - أمر فتح الملف OPEN :

فى السطر رقم ٢٠ والصورة العامة له هى نفسها التى شرحها فى حالة بناء الملف العشوائى.

٢ - توصيف الحقول FIELD :

فى السطر ٣٠ تم أستخدام جملة FIELD لتقسيم المخزن الوسيط RANDOM BUFFER إلى حقول البيانات التى ستقرأ من الملف.

ملاحظة :

إذا أستخدم البرنامج لأداء عمليتى إدخال بيانات فى ملف عشوائى وإسترجاعها من نفس الملف يمكن أستخدام جملة OPEN واحدة وكذلك جملة FIELD واحدة.

٣ - قراءة أو إسترجاع البيانات من الملف GET :

يتم قراءة السجل المطلوب من الملف العشوائى ونقله إلى المخزن الوسيط RANDOM BUFFER بأستخدام جملة GET والصورة العامة لها هى:

GET (#) FILE - NUMBER (RECORD NUMBER)

حيث (#) إختيارية و FILE - NUMBER هو رقم الملف الذى أستخدم فى فتح الملف و RECORD - NUMBER هو رقم السجل المطلوب قراءته وهو إختيارى ، وإذا لم يحدد رقم السجل يتم أستدعاء السجل التالى لآخر سجل تم أستدعاؤه عن طريق آخر جملة GET سابقة ، وفى السطر ٩٠ من البرنامج يتم قراءة السجل رقم (X) من الملف رقم (١).

٤ - بعد ذلك يتم عرض البيانات على الشاشة بأستخدام أمر PRINT أو يمكن عرضها بأستخدام الطابعة مع ملاحظة أن القيم التى سبق تخزينها كسلاسل حرفية والتى يتم نقلها من المخزن الوسيط RANDOM BUFFER يجب أن تحول من صورة السلاسل الحرفية إلى قيم عددية قبل طبعها وتستخدم فى ذلك الدوال CVI, CVS , CVD وتستخدم الدالة CVI لتحويل السلسلة المكونة من ٢ بايت إلى عدد صحيح ، وتستخدم الدالة CVS لتحويل السلسلة المكونة من ٤ بايت إلى عدد حقيقى REAL NUMBER OR SINGLE - PRECISSION NUMBER وتستخدم الدالة CVD فى تحويل السلسلة الحرفية المكونة من ٨

بايت إلى عدد مزدوج الدقة DOUBLE - PRECISSION NUMBER ، والسطر ١١٠ من البرنامج السابق يوضح ذلك.

ملاحظات هامة :

فيما يلى بعض الملاحظات التى يجب مراعاتها عند كتابة برنامج يستخدم فى بناء ملف وإسترجاعه وتعديله :

أ - يمكن حجز أول سجل فى الملف ليكتب فيه عدد السجلات التى يتكون منها الملف فإذا كان عدد السجلات المطلوب كتابتها فى الملـــف = (N) ففى هذه الحالة يكون عدد سجلات الملف = (N + 1) .

ب - قبل الكتابة فى أى ملف يفضل إدخال حرف معين فى الحقل الأول فى جميع السجلات وذلك حتى يمكن أستخدامه بعد ذلك فى معرفة هل تـــم أستخدام سجلات الملف بإدخال بيانات فيها أم لم يستخدم بعد.

*

obeikandl.com

١٠ - الملفات

عرضنا خلال الفصل السابق لمحة سريعة عن الملفات ، وفى هذا الفصل نعرض إليها بقدر أكبر من التفصيل.

ولان الحاسب ليس صندوقا سحريا تصب فيه البيانات كيفما اتفق وتتولى آلاته وبرامجه استكمال معالجة ماصب فى جوفه وتقدم المخرجات لمستخدم الحاسب ، لانه ليس كذلك يجب تنظيم البيانات بما يلائم قدراته وامكانياته ، وهذا الاعداد يشمل تنظيم البيانات على هيئة ملفات كل منها يختص بموضوع معين ويعتبر وحدة واحدة ويمكن تخزينة تخزينا دائما على وسائط التخزين الثانوية مثل الشرائط المغناطيسية والاقراص حتى يمكن الرجوع اليه عند الحاجة.

وكشأن كل عناصر منظومة الحاسب من ازدواجية الكيان يمكن النظر للملفات من منظورين:

المنظور المنطقى LOGICAL VIEW:

ويركز هذا المنظور على خصائص حقول بيانات السجلاتRECORDS المكونة للملف ، كما يركز على المعالجات اللازمة على البيانات ، ونلحظ أن النظرة المنطقية لاتلقى بالا لوسيط التخزين المحمل عليه الملف.

منظور فيزيائى LOGICAL :

ومحور هذا المنظور معرفة الوسيط الحامل للبيانات وقدراته واسلوب تنظيم واستعادة البيانات.

وعلى هذا يمكن تعريف الملف على انه مجموعة من السجلات المرتبطة منطقيا فليس مقبولا أن ننشئ سجلات للعاملين أو سجلات لكتب احدى المكتبات ونعثر داخل الملف على سجل يتعلق بسيارة أو قطعة غيار ، فوجود هذا السجل الغريب يكسر قاعدة الارتباط المنطقى والموضوع الواحد للملف ، كما يمكن تعريف السجل بأنه مجموعة مرتبطة منطقيا من عناصر البيانات أو حقول البيانات ، وكل حقل يتكون من مجموعة متناغمة من الحروف الابجدية ذات معنى ودلالة ومضمون.

ويمكن تلخيص ما اسلفنا على النحو:

الحرف : هو اصغر مكون منطقى فى الملف وقد يكون ابجدى أو رقمى.

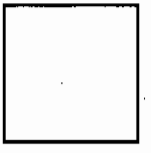
الحقل : وهو اصغر وحدة بيان ويتشكل من مجموعه متناغمة من الحروف.

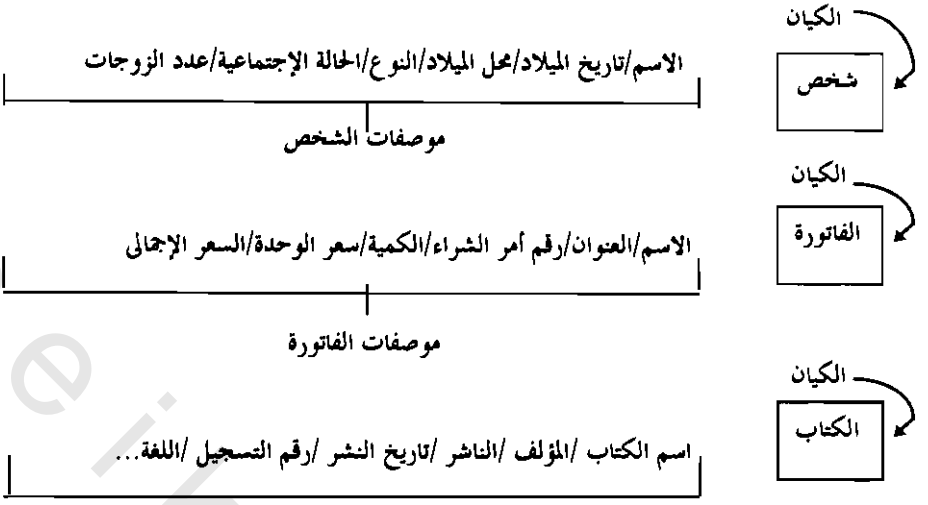
السجل : مجموعه مترابطة منطقيا من الحقول توصف كيان بذات توصيفا يتلائم مع طبيعة ومتطلبات الملف ، وقد تسمى الحقول بالموصفات للكيان ATTRIBUTES .

الكيان Entity :

هو شيء محدد مثل الانسان ــ السلعة ــ التاجر ــ الموظف ــ أو نظام المعلومات ، أو هو شيء نهتم بتسجيل بيانات عنه أو بمعنى آخر الكيان هو المصدر أو الباعث للموصفات فيما توضحه المخططات التاليه فى شكل (١٠/١) ، وشكل (١٠/٢).

ويرمز لأى كيان
برسم شكل المربع
ويكتب داخله أسم الكيان



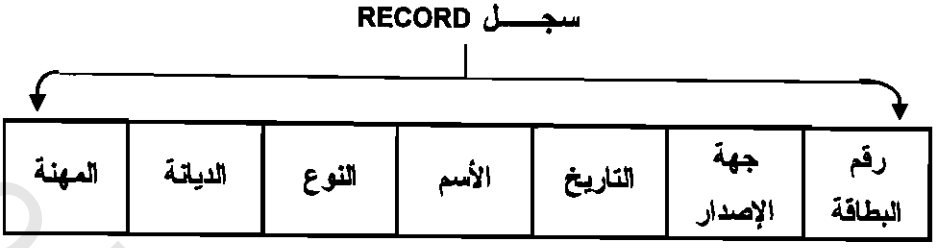


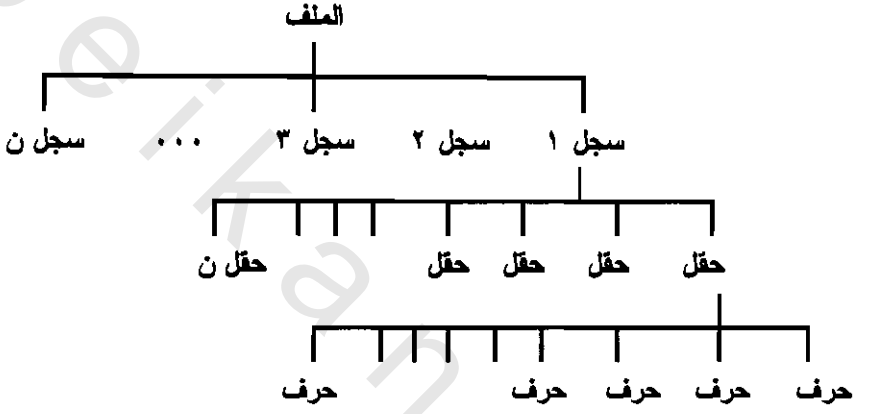
ويمكن تلخيص ما سبق فى الجدول التالى :

المسمى	الشـــــــــــرح	عدد الموصفات
كيـــان	هى أغراض مطلوب تسجيل بيانـات عنها.	غير محدود حسب طبيعـة كـل كيان.
السجل المنطقى	سجل منطقى واحد لكل كيان .	كل سجل يحتوى على عدد مـن عناصر البيانات أو الموصفات.
السجل الفيزيائى	قد يحتوى على أكـثر مـن سـجل منطقى.	كل سجل منطقى يحتـوى علـى عدد من الحقول.

الشكلان (١ ، ٢/١٠)

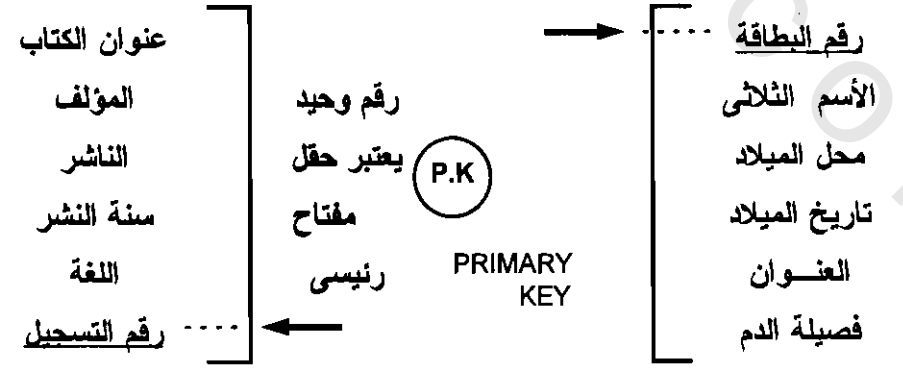
obeikandl.com





أنواع حقول السجل:

إذا تدارسنا بيانات البطاقة الشخصية أو بيانات تسجيل كتاب بأحد المكتبـــات لوجدنا أن موصفات صاحب البطاقة أو موصفات الكتاب كما هى موضحـــة علـــى النحو:



وسوف نلاحظ ان هناك نوعين من الحقول حقل وحيد له قيمة لاتتكرر على مستوى السجل وعلى مستوى الملف كله مثل رقم البطاقة / الرقم القومى / رقم تسجيل الكتاب ، وكلاهما رقم فريد Unique ولذلك اذا سألت عن بيانات صاحب البطاقة رقم (كذا) أو ما هو عنوان واسم مؤلف الكتاب الذى رقم تسجيله (كذا) فما إيسر الوصول إلى البيانات المطلوبة لإن هذا النوع من الحقول يشير مباشرة الى باقى حقول السجل ويحدد الكيان المعبر عنه تحديداً دقيقا.

مثل هذه الحقول الخاصة تسمى حقل مفتاح رئيسى Primary Key ولاينفى وجود هذا النوع من الحقول وجود حقول اخرى تصلح للاشارة للسجل ويطلق عليها المفاتيح الثانوية Secondary Key ، كما يمكن الدمج بين اكثر من حقل غير مفتاحى لتكوين حقل مفتاحى Unique ، أما باقى الحقول فهى حقول يحتمل تكرارها فى أى عدد من السجلات فى الملف الواحد ، ففى ملف البطاقات الشخصية قد يتواجد عدد "ن" من السجلات لها نفس أسماء أصحاب البطاقات ، وربما نفس محل الميلاد ويحتمل نفس تاريخ الميلاد ولاتنفى احتمال وجود ولو عدد محدود جدا من الاشخاص لهم نفس العنوان وهكذا .

الرقم القومى	الأسم	محل الميلاد	تاريخ الميلاد	العنوان	فصيلة الدم

والحقل الفريد أو المفتاح الرئيسى هو احد الادوات المنطقية فى إسترجاع البيانات من الملفات كما سنأتى إلى ذلك تفصيلا.

أنواع الملفات:

تقسم الملفات إلى عدة أنواع أبرزها:

١ ــ **الملف الرئيسى Master File :**

وهو ملف ذا بيانات شبه مستديمة (المرتبات) ويتم تحديثه على فترات متباعدة (شهريا ــ نصف سنوى ــ سنوى) وبالتالى فان الملف الرئيسى يحتوى على بيانات ذات طبيعة استاتيكية (الاسم ــ العنوان ــ المرتب الاساسى...).

٢ ــ **ملفات المتغيرات Transaction File :**

وهى ملفات تضم المتغيرات اليومية عى البيانات ــ الحركة ــ مثل اوامر الشراء الواردة للمؤسسة خلال فترة معينة وتستخدم هذه الملفات فى تحديث الملف الرئيسى.

٣ ــ **ملفات جامدة أو مرجعية Reference File :**

وهى الملفات التى تضم بيانات لاتتغير مثل الاسم ــ تاريخ الميلاد ــ محل الميلاد ــ النوع ــ الديانة ــ اسماء الكتب وسنه النشر ــ الموردون لمكتبة الجامعة.

٤ ــ **ملفات البرامج وبياناتها:**

وهى ملفات تضم بعض البرامج أو البيانات المرتبطة بالبرامج ، وقد تكون البرامج والبيانات المرتبطة على نفس الملف أو على ملفين مستقلين.

٥ ــ **ملف نصوص Text File :**

ويضم نص وثيقة أو عدة وثائق مرتبطة منطقيا.

٦ ــ **ملف شغل Work File :**

وهو ملف ينتج خلال مراحل المعالجة ولا يجرى تخزينه شأنه شأن المسودات فى الأعمال الإدارية والمكتبية.

٧ ــ **ملف تجريب Scratch File :**

وهو ملف مؤقت يمحى فور انتهاء المعالجة بإن يعاد الكتابة عليه.

العمليات على الملفات:

تتم جملة عمليات على الملفات هى على النحو:

١ ــ التحديث Updating :

عندما تتغير البيانات المتاحة عن البيانات المسجلة يتم تحديث الملف بالبيانات الجديدة.

٢ ــ المرجعية Referncing :

لتأكيد معلومة من معلومات مسجلة ولايجرى خلالها أى تعديل أو تحديث.

٣ ــ صيانة الملفات:

وهى عملية تشمل إضافة سجلات جديدة وحذف سجلات قديمة مثل تغيير الثمن ، حذف أصناف لم تعد تنتج ــ عناوين العملاء ــ أرقام التليفونات.

أطوال السجلات:

تقسم السجلات من حيث الطول (الحيز مقدر بوحدة البايت) إلى نوعين أساسيين هما:

أ ــ طول ثابت Fixed :

أى أن حيز كل سجل على الملف مساو تماما لحيز أى سجل آخر ، ويعتبر هذا النوع من افضل انواع السجلات لأنه يساعد على سهولة تصميم البرامج وإن كان لايحقق كفاءة عالية فى استغلال وسائط التخزين اذ يعتبر الحيز الذى لم يستخدم مجرد فراغات محجوزة Blank يستحيل إستغلالها.

ب ــ الطول المتغير Variable Length :

يتميز بأن اطوال السجلات ليست موحدة ، بعض السجلات قصيرة وبعضها طويل ويعود هذا الى ان بعض السجلات بها عدد أكبر من الحقول ، فأن تساوت فى عدد الحقول فالسبب يعزى إلى إختلاف حيز الحقول ذاتها، وتعتبر

obeikandl.com

سجلات الطول المتغير من أفضل أنواع السجلات إستغلالا لحيز التخزين على الوسائط المختلفة.

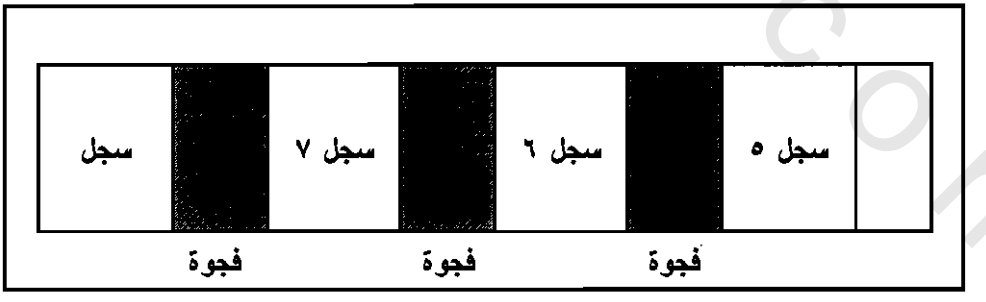
نشاط الملفات:

يعتبر نشاط الملف أو حجم الحركة عليه احد المعايير الهامة Terms في تصميم الملف من حيث اختيار وسائط التخزين والتنظيم واسلوب الاسترجاع ، كما أن درجة ثبات البيانات وعدم تغيرها أو تحديثها يعتبر من المعايير التى تؤخذ في الاعتبار فإذا كان تغير البيانات يتصف بالسرعة دعيت الملفات بأنها ملفات متطايرة Volatile .

تنظيم الملفات على الوسائط التخزينية:

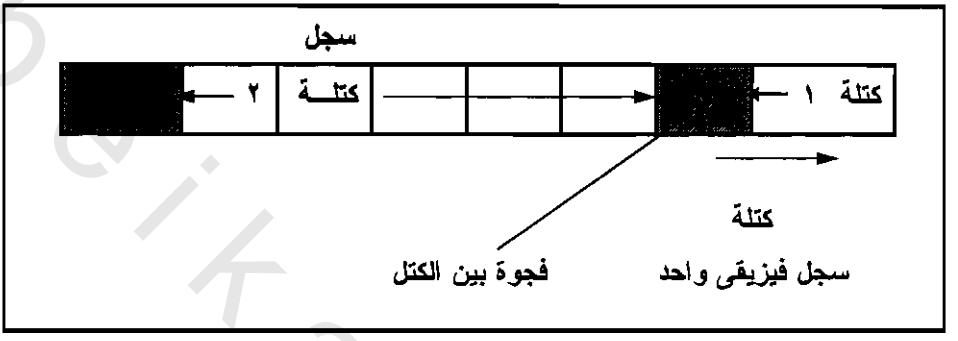
أولا: الملفات على الشرائط المغناطيسية:

تكتب السجلات من الذاكرة على الشرائط بالامر "أكتب" وكل سجل يتم كتابتة يتلوه سجل آخر حتى نهاية تسجيل السجلات ، ونلحظ من ميكانيكية الكتابة على الشريط أن دوران الشريط يأخذ أقصى سرعة منتظمة من بداية الكتابة حتى نهايتها ثم يبطىء الدوران لدرجة التوقف بعدها يعاود إكتساب السرعة اللازمة مما يستغرق وقتا يسمح بمرور حيز من الشريط دون الكتابة عليه مسببا فجوة بين السجلات Inter Record Gab كما هو مبين بالشكل (١٠/٣).



شكل (١٠/٣)

وحتى نقلل الفجوة لادنى حيز ممكن تخزن السجلات على هيئة مكتلة Blocked ويضم السجل الفيزيائى الواحد عدة سجلات منطقية يتراوح عددها بين سجل واحد وعدد {ن} من السجلات ويوضح الشكل (١٠/٤) الفجوة البينية بين الكتل الفيزيائية للسجلات.



شكل (١٠/٤)

الفجوة بين الكتل

والتكتيل يرفع من كفاءة نقل البيانات ويساعد على تحسين إداء وحدة التشغيل المركزية علاوة على الاستغلال الجيد للشرائط .

معامل التكتيل:

هو عدد السجلات المنطقية فى السجل الفيزيائى (الكتلة) ويتراوح بين سجل واحد ، وعدد "ن" من السجلات فاذا كان معامل التكتيل يساوى واحد فانه لاتكتيل ، وإذا تم النص على إجراء تكتيتل فإن الامر "اكتب" يكتب كتلة كاملة وليس سجلا مفردا وعندما يصدر الأمر بالقراءة من الملف فإن الحاسب يقرأ كتلة كاملة ويفككها نظام التشغيل إلى سجلات مفردا.

تنظيم الملفات على الشرائط المغناطيسية:

١ ــ تنظم الملفات وفق المعيار الزمنى لورود السجلات فمن يإتى اولا يسجل اولا ويسمى هذا التنظيم Pile كومة أو Serial متوالى أو متسلسل.

٢ ــ إذا أعيد ترتيب سجلات الملف المسلسل وفق قيمة حقل المفتاح سمى الملف فى هذه الحالة ملف متتالى Sequential .

وكما نظمت الملفات مسلسلة أو متوالية فإن استرجاع السجلات يكون بذات ترتيب وتنظيم الملف وهو اسلوب فرضته تقنية الشرائط مما حد من استخدامها فى الانظمة النشطة أو الملفات المتطايرة ، حتى عند تحديث بيانات سجل يعاد كتابته فى موقعة الاول مما يستدعى عند تحديث الملفات ان يكون شريط ملف المتغيرات له نفس تنظيم الملف الرئيسى ، وفى حالة حذف أى سجل يبقى مكانه خاليا على الشريط حتى يتم كتابة الشريط (الملف) مرة اخرى.

توصيف وتسمية الملفات:

إلى جانب سجلات البيانات يتم تعريف وتوصيف الملف بإستخدام سجلين احدهما فى بداية الملف والآخر فى نهايتة ويكتبا مغناطيسيا ويشملا اسم الملف ــ تاريخ فتحه ــ الصلاحية ــ تاريخ إنتهاء صلاحيته ، ويتولى برنامج التطبيقات اختبار البيانات والتأكد من مطابقة الشريط للبرنامج ، ويدون فى سجل نهاية الملف ، إشارة نهاية الملف ــ عدد السجلات ــ رقم الشريط.

ثانيا: الملفات على الاقراص المغناطيسية:

ناقشنا فى الباب الثانى (وسائط التخزين الثانوية) طريقة التسجيل على مجموعة الأقراص الصلبة وكيفية إعداد الأسطوانات المكونة من المسارات المتماثلة على الأسطح المغناطيسية حيث يقسم كل مسار إلى عدد من القطاعات ويتراوح بين

obeikandi.com

٨ ، ١٢٨ قطاعا ويسمى القطاع الواحد كتلة البيانات لان بيانات كل قطاع تقـــرأ أو تكتب دفعة واحدة فيما بين القرص والذاكرة فيما يناظر كتلة منطقية على الشريـــط المغناطيسيى ، وقد يتم جمع الكتل المنطقية إلى بعضها البعض مكونة Buckets فيما يوضحه الشكل (١٠/٥) مما يسرع عمليات القراءة الكتابة على الاقـــراص وتنظـم الملفات على الاقراص بأيا من اساليب التنظيم الآتية:

أ ــ تنظيم مسلسل كما فى الشرائط.

ب ــ تنظيم متتالى كما فى الشرائط .

وكلا الاسلوبين يعتبر إهدارا لقدرات الحاسب وسرعة التعـــامل مـــع الاقراص سيان فى تسجيل البيانات أو إسترجاعها.

ج ــ تنظيم متتالى مفهرس ويتطلب انشاء ملف فهرس إلى جـــانب ملـــف البيانات.

ولايحقق هذا التنظيم سوى سرعة إسترجاع البيانات ويحتـــاج حـــيز تخزين أكبر من حيز ملف البيانات.

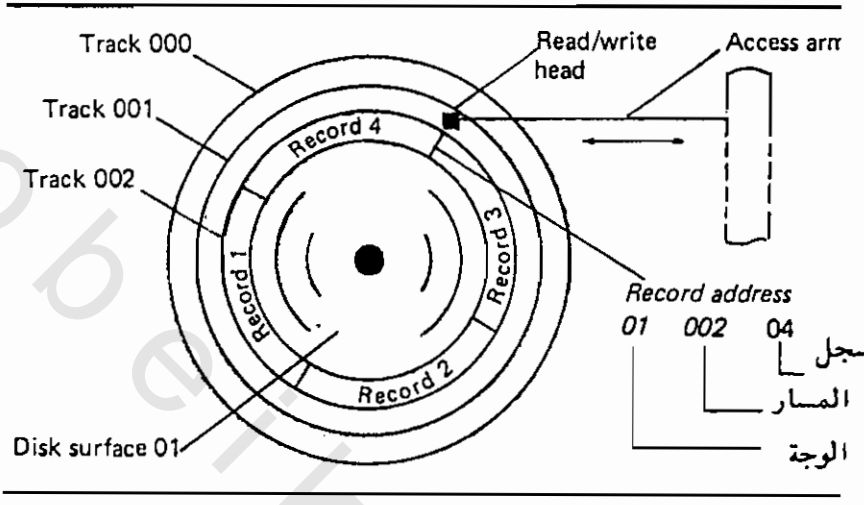
د ــ تنظيم عشوائـــــى (مباشر).

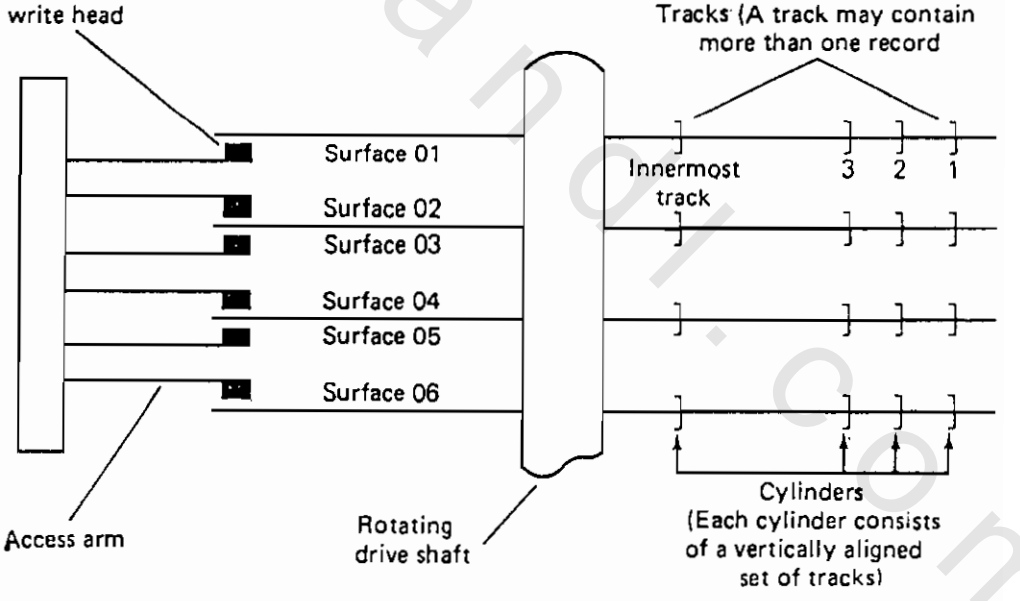
وإن كنا سنتناول هذه التنظيمات تفصيلا فى صفحات لاحقة فلابد من التأكيد هلى أن تقنية الاقراص تتيح الاسترجاع الفورى لأى ســـجل علـــى أى موقـــع مـــن الأسطوانات الوهمية للبيانات مما يتطلب عنونـــة الســـجلات Addressing وتقســـيم مواقع التخزين إلى أجزاء يسهل الوصول إليها وفق هذا الترتيب الهرمى:

١ ــ **الأسطوانة:** عدد الأسطوانات يساو عدد المسارات على وجه أى قرص.

٢ ــ **المسار** : عدد المسارات فى الأسطوانة الواحـــدة يســـاوى عـــدد أوجهـــه الأقراص المغناطيسية ناقص ٢.

٣ ــ **القطاع:** وهو اقل كتلة منطقية يمكن عنونتها وتقرأ أو تكتب دفعة واحدة.





شكل (١٠/٥)

القطاعات والمسارات وفهرسة السجلات

وبذلك يكون عنوان السجل على النحو من اليسار إلى اليمين كما فى الإطار التالى:

الموقع النسبى	=	رقم السطح	رقم المسار	رقم القطاع

تنظيم الملفات على الاقراص المغناطيسية:

١ ـ التنظيم المباشر Direct Organization :

يعتمد هذا التنظيم على تخزين السجلات واسترجاعها وفق عنوان لكل سجل يحسب من معادلة رياضية تطبق على حقل المفتاح الرئيسى فان كان الحقل ابجدى فيجب تحويله الى قيمة رقمية بعدها يتم حساب قيمة العنوان.

٢ ـ التنظيم المفهرس Indexed Organization :

يعتمد هذا التنظيم على انشاء ملف صغير يسمى ملف الفهرس Index File إلى جوار ملف البيانات Data File ويعتبر ملف الفهرس المدخل الاساسى لملف البيانات ، ويتكون سجل ملف الفهرس من حقلين،الاول المفتاح الرئيسى أو حقل المفتاح لسجل البيانات الى جانب حقل الفهرس المحدد لموضع السجل على الوسيط المغناطيسى ، مما يعنى أن كل سجل بيانات له مدخل وحيد فى ملف الفهرس ، ونؤكد مرة اخرى ملف مفهرس يعنى ملفين ، ملف بيانات وجواره ولصيق به ملف فهرس ويمكن إتباع التنظيم المفهرس فى حالتين:

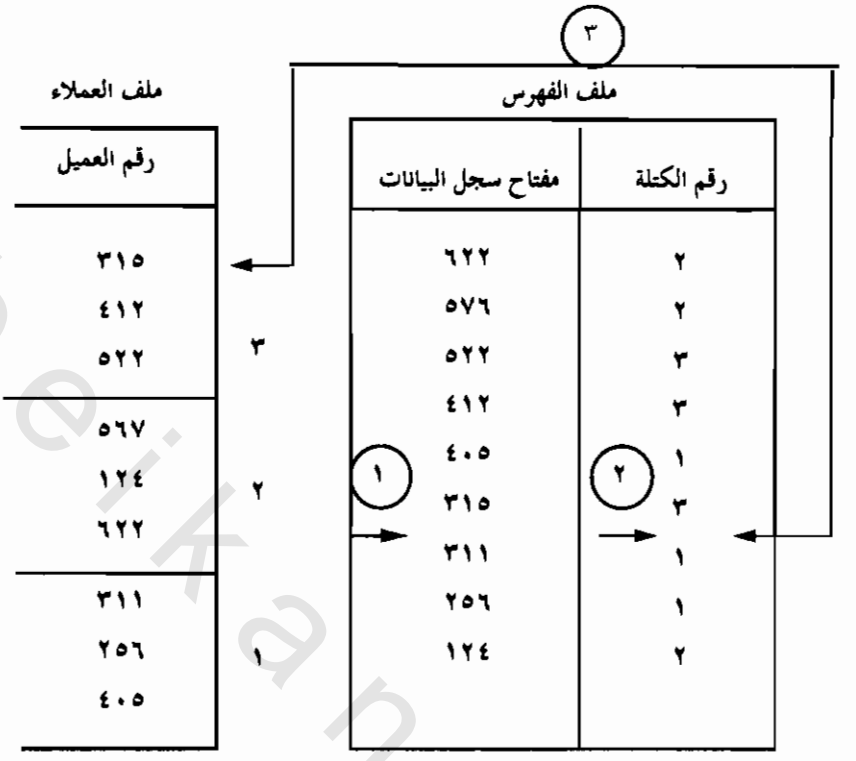
الأولى: تنظيم عشوائى مفهرس Indexed Random Organization :

عندما يتم كتابة سجلات ملف البيانات دون ترتيب أو تحديد على وسائط التخزين فهذا مانعنيه تماما بمفهوم التنظيم العشوائى المفهرس

جواره سجل الفهرس الذى يضم حقلين كما هو موضح مما دفع إلى عدة تساؤلات عن الفائدة المحققة من استخدام الفهارس خاصة وهو ملف سوف يشغل حيزاً لايستهان به من وسائط التخزين ويتطلب انشاء وصيانة وتحديث ومعالجة شأن كل الملفات؟

هل التكلفة الاضافية لانشاء الفهرس تعطى مردودا يستدعى هذا الجهد والتكلفة؟

ومجموعه الاسئلة المثارة توضح مدى اهمية الفهارس ، فهذا التنظيم يتيح الاسترجاع المتتالى رغم أن سجلات البيانات موزعة عشوائيا اثناء التسجيل.. كيف؟ بإعادة فرز وترتيب ملف الفهارس وسلسله مفتاح سجل البيانات تصاعدياً يتيح هذا النوع من الاسترجاع ، ايضا يمكن استرجاع سجلات البيانات عشوائيا دون جهد باجراء البحث فى ملف الفهرس وصولا الى القطاع ــ الكتلة ــ التى تضم السجل المنشود مباشرة مما يقلل من زمن الاسترجاع والذى ينعكس بدوره على كفاءة النظام لان التعامل مع ملف سجلاته لا تتعدى ٨ بايت أبسط كثيراً من التعامل مع سجلات قد يصل حيز السجل الواحد فى ملف البيانات مئات أو عده آلاف من البايت.



التنظيم المتتالى المفهرس Indexed Sequential :

ويضم مزايا ملف الفهارس الى جانب أن سجلات البيانات مرتبة ترتيبا متتاليا وفق حقل المفتاح ، وفى هذه الحالة ليس هناك ادنى حاجة الى فهرس كثيف يضم مدخلات لجميع السجلات ويكتفى بفهرس مختصر مما يساعد على سرعة البحث وتقليل زمن استجابة النظام مع استخدام حيز تخزين محدود وقد يوضع الفهرس المختصر ذاته فى الذاكرة مما يرفع من كفاءة واستجابة النظام.

مزايا نظم الملفات:

١ ــ الملفات المتتالية تستهلك من حيز التخزين ادنى قدر متاح خصوصا إذا كانت مكتلة ، لكن الاسترجاع المتتالى يبرز عدم مرونة هذا التنظيم.

٢ ــ تمتاز الملفات المباشرة بأقصى سرعة استرجاع لكنها غير مناسبة من حيث الاسترجاع المتتالى.

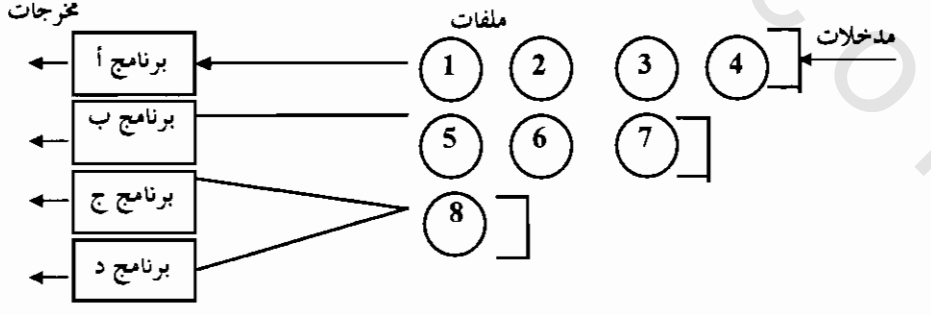
٣ ــ التنظيم المفهرس يتيح للمستخدم الاسترجاع المتتالى والمباشر رغم أن الاخير ليس بالسرعة المناسبة وهنا يتفوق التنظيم المتتالى المفهرس.

٤ ــ معظم نظم الملفات تلبى متطلبات النظم غير النشطة مثل انظمة المرتبات والمخزون مما يستدعى أستخدام تكلفة إضافية.

٥ ــ من ابسط الاساليب فى ادارة البيانات رغم العيوب الكثيرة التى تعانى منها.

عيوب نظم الملفات:

حتى السبعينات ونظرا للقصور الكبير فى معدات الحاسبات صممت معظم التطبيقات على مبدأ الارتباط المباشر بين البرنامج والبيانات الخاصة بها فيما عرف باسم Data-Program-Dependence ، فعندما كان يكتب برنامج بلغة الكوبول فأنه يجرى إنشاء ملف يضم البيانات اللازمة لهذا البرنامج ، وعندما يكتب برنامج بلغة البيزيك ينشأ معه ملف آخر تطابق صياغة بياناته برامج البيزيك وهكذا.



حقيقة أن كل برنامج يمكنه التعامل مع أكثر من ملف لكنه يتعامل معها تتابعيا لأنه لايمكنه التعامل مع أكثر من ملف واحد فى الوقت الواحد ، شريطة أن تكون الملفات مكتوب بياناتها بطريقة تلائم لغة البرمجة المستخدمة كما أسلفنا.

هذه الحقيقة كان لها نتائجها السلبية التى نلخصها على النحو:

١ ــ حدت من مرونة النظام.

٢ ــ سببت نقص كفاءة النظام.

٣ ــ أتاحت وجود تكرارية فى البيانات الواحدة.

٤ ــ رغم الارتباط المباشر بين البيانات ولغة البرمجة فليس متاحا استخدام بيانات برنامج لبرنامج آخر رغم أنهما مكتوبان بنفس لغة البرمجة مما حتم استخدام نسخ جديدة من ملف البيانات ليلائم البرنامج الآخر.

٥ ــ تسببت تكرارية البيانات فى عدم تحقيق التكاملية بينها لان تحديث احد الملفات لايعنى بالضرورة تحديث الآخر.

٦ ــ انعكس كل هذا على زيادة كبيرة فى وسائط تخزين البرامج والبيانات.

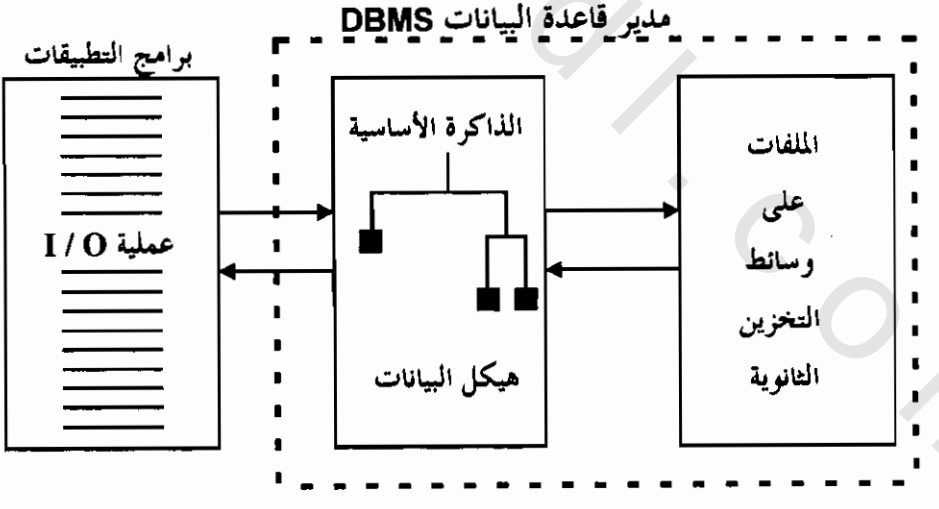
كل هذه المشاكل قادت الى التفكير نحو إيجاد حلول مناسبة لأحداث تكاملية بين مختلف الملفات وإنهاء مشكلة تكرارية البيانات... إلخ.

وقد سميت هذه الفكرة تكاملية الملفات Integrated Files ثم شاع مسمى قواعد البيانات Data Base وتتلخص الفكرة فى وضع البيانات ضمن إطار موحد دعنا نطلق عليه للتبسيط حوض البيانات Data Pool أو قاعدة بيانات Data Base بحيث يستطيع مختلف مستخدمى النظام التعامل معها بشكل سهل ومبسط.

*

١١ ـ هياكل البيانات

تستخدم كلمة هيكل وهيكلة فى مجالات عديدة لتوصيف اغراض كبيرة تـــم بنائها من وحدات بنائية صغيرة باسلوب تكرارى ، كأن نبنى سورا ضخما من وضع قوالب الطوب جوار بعضها البعض ونلصقها بالأسمنت. ونلمس الهيكلة فـــى أمـــور شتى ، قطعة الفلز عبارة عن بناء مهيكل من وحدة بنائيـــة أصغـــر هـــى الـــذرة ، والبلورات بناء مهيكل من الايونات ، وهياكل البيانات هى كتلة منطقية نجمت عـــن تكرارية عناصر البيانات وفق ترتيب محدد وعلاقات مثل الأسمنت مـــع الطـــوب، وتعتبر هياكل البيانات مرحلة وسيطة بين الملفات على وسائط التخزين وبين برامج التطبيقات فيما يوضحه الشكل (١١/١).



الشكل (١١/١)

تعريف هياكل البيانات:

يمكن تعريف هياكل البيانات على أنها (بناء منطقى كبير من عناصر البيانات أو مجموعات منظمة من عناصر البيانات تعامل كوحدة واحدة ، مثل السجلات أو المصفوفات) . كما نستطيع تعريفها بأنها (مجموعة من عناصر البيانات المنظمة وفق أسلوب يوحى بكيفية تخزينها واسترجاعها كعناصر بيانات مستقلة).

والتعريف الأخير ليس على اطلاقه ففى استرجاع البيانات من جداول قواعد البيانات العلاقية لايستطيع المستخدم غير المتخصص تحديد أسلوب الاسترجاع ، لكن فى بعض هياكل البيانات مثل المصفوفات يجب على المستخدم فهم طبيعة المصفوفة وكتابة البرامج اللازمة والمحققة لاهدافه مما يستدعى فهم هيكل البيانات ومناظرة الشكل المنطقى بالوضع التخزينى للبيانات داخل الذاكرة الأساسية.

أنواع هياكل البيانات:

تقسم هياكل البيانات الى عدة أنواع يناظر كل منها اسلوبا خاصا فى تصميم البرامج والمعالجة الالكترونية ويمكن إدراجها تحت ثلاثة أنواع:

١ ــ هياكل بيانات خطية :

أى ترص البيانات داخل الذاكرة على خط واحد وتشمل :

أ - المصفوفة ARRAY .

ب - سلسلة الحروف STRING (السلاسل).

ج - السجلات RECORDS والملفات (الجداول).

د - القوائم LISTS .

هـ - الكومه STACK .

و - قوائم الانتظار QUE .

٢ ــ **هياكل بيانات شجرية :**

وتعنى شجرة البيانات Tree (الهياكل الهرمية).

٣ ــ **هياكل بيانات شبكية :**

وتعنى Plex الشبكات (الرسوم Graphs) كما تسمى بيانات مضفرة.

أولا : هياكل البيانات الخطية :

١ - المصفوفة:

دعنا نتفهم طبيعة المصفوفة من مثالنا التالى ، اذا كانت درجات مادة قواعد البيانات لفئة من الطلبة على النحو (٤٥ ، ١٥ ، ٣٠ ، ٣٢ ، ٤٨ ، ٥٥ ، ١٢) فأنه يمكن القول أن درجة الطالب الأول تساوى ٤٥ ، ودرجة الثانى ١٥ والثالث ٣٠ وهكذا ، فاذا رمزنا لعنصر البيانات بالرمز (ع) فإن ع (١) = ٤٥ ، ع (٢) = ١٥ ، ع (٣) = ٣٠ ، ع (٧) = ١٢ وهكذا ، ويسمى الرقم الى جوار [ع] بالرقم الدليل ويشير الى رقم العنصر أى ترتيبه فى ذات الفئة ، فاذا اعتبرنا أن الرقم الدليل له قيمة عامة هى (I) فإنه يتيح التعامل المتكرر ومعرفة أى عنصر فى فئة الدرجات بتغير قيمة (I). ورص الدرجات على النحو السابق يسمى مصفوفة ، واستخدام الرقم الدليل SUBSCRIPT يسمح بمعالجة ومعرفة كل عنصر من عناصرها على حده.

خصائص المصفوفة:

نلحظ مما سبق أن للمصفوفة عدة خصائص أبرزها:

أ - تحتوى على عدد محدد من عناصر البيانات.

ب - تحتوى على نوع واحد من عناصر البيانات.

ج - ذات حيز ثابت لايتغير ولا يتبدل فى المصفوفة الواحدة.

د - عندما يتم تخزين بيانات المصفوفة فى ذاكرة الحاسب فإن هناك تناظر وتماثل بين شكل المصفوفة المنطقية (المكتوبة على الورق) وبين شكلها وهيئتها فى ذاكرة الحاسب مما يساعد على انشاء منطق برنامج يتعامل مع عناصرها باستخدام الرقم الدليل (I) .

هـ - يتطلب تخزين المصفوفة فى ذاكرة الحاسب واجراء معالجة بياناتها عدة خطوات على النحو التالى:

(١) فتح مخازن فى ذاكرة الحاسب للتخزين المؤقت للبيانات بأمر التخصيص LET كما فى لغة البيزك وذلك وفق منطق البرنامج الذى سيتعامل مع البيانات. ويعيب هذه الطريقة البطىء وإحتمالات الخطأ.

(٢) أو تخصيص مواقع متلاصقة فى ذاكرة الحاسب لعناصر البيانات ، وهذه وظيفة تؤدى فى كل لغات البرمجة بإمر ذاتى BUILT IN FUNCTION مثل الأمر "(الطول) (الاسم) DIM " . كما فى لغة البيزك مثل DIM X (100).

(٣) استخدام الرقم الدليل فى الاشارة لعناصر البيانات المطلوب التعامل معها.

(٤) إذا تعامل البرنامج مع بعض عناصر البيانات فإن البرنامج لايمكنه التعامل مع نفس العناصر مرة اخرى رغم بقاء البيانات فى مواقع التخزين داخل الذاكرة وقد حلت لغة البيزك هذه المشكلة بالأمر Restore مع إتاحة إعادة تسمية المتغيرات والتعامل معها على إعتبارها مدخلات جديدة.

المصفوفات ذات البعدين:

إن معالجة المصفوفة الآحادية يوضح مدى التطابق بين المصفوفة المنطقية والأخرى المخزنة فى ذاكرة الحاسب ويوضح كذلك ضرورة معرفة البرنامج عنوان

أول خلية تخزين داخل الذاكرة مما يساعد على تحديد باقى مواقع تخزين عناصر المصفوفة ، لكن هذه البساطة فى معالجة المصفوفة ذات البعد الواحد ليس شرطا فى المصفوفات المتعددة التى تتركب من عدد من الأعمدة وعدد آخر من الصفوف مثل مصفوفة لنفس الطلبة لعدد من المواد الدراسية حيث يختلف الشكل المنطقى لها عن الشكل التخزينى فيما يوضحه الجدول.

اسم الطالب	درجة المادة الأولى	درجة المادة الثانية	درجة المادة الثالثة
أحمد أحمد أحمد	١٥	٢٠	١٣
سناء أحمد أحمد	١٨	١٢	١٦
زكى رمزى	١٤	١٠	١٨
فيليب	١٨	١٦	١٩
رمزى	١٣	١٤	١١
نجاة	١٠	١٥	١٤
هديل	١٦	١٢	١٠

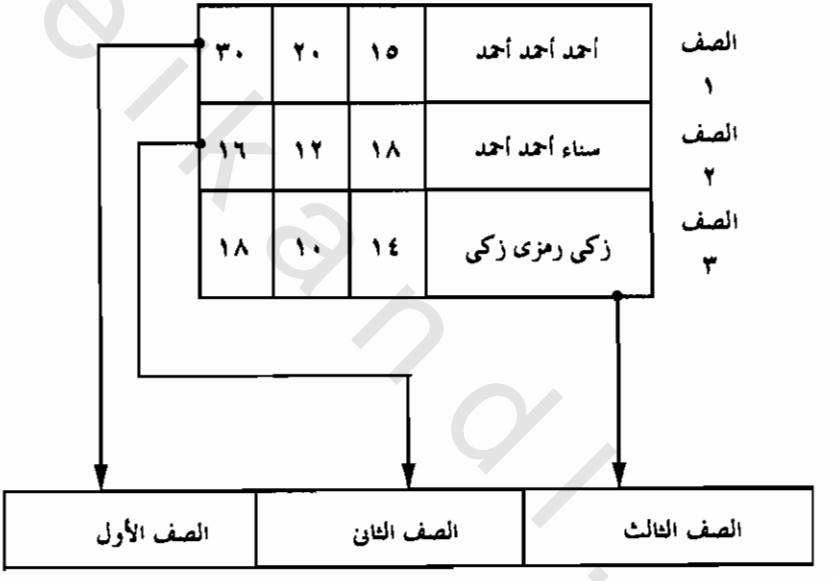
فإذا كان العمود الأول من جهة اليسار يرصد درجات الطلبة فى الكيمياء والذى يليه هو درجات الفيزياء والعمود الأخير يمثل درجات الطلبة فى الاحياء ، فإن مثل هذه المصفوفات تستدعى أن نتذكر مايلى:

١ - أن ذاكرة الحاسب ليست على هيئة جدول بل هى خلايا تخزين متراصة متجاورة على مستوى واحد.

٢ - ضرورة برمجة المستويين س ، ص على الورق داخل الذاكرة على مستوى واحد وواحد فقط.

وحتى يجرى ذلك لابد أن نتأكد أن حيز المصفوفة لن يتغير عند اجراء تحديث على البيانات بعدها يتم ادخال البيانات الخاصة بالصف الأول ثم الصف الثانى ثم الصف الثالث وهكذا على النحو الموضح فى الشكل (١١/٢) مما يستدعى:

شكل الجدول على الورق



الشكل داخل ذاكرة الحاسب

شكل (١١/٢)

أ - التحديد الدقيق لعدد وحدات التخزين اللازمة للمصفوفة.

ب - حجز مواقع التخزين فى الذاكرة متلاصقة ما أمكن.

ج - تخزين البيانات فى الذاكرة صفا صفا.

د - تحديد المداخل الصحيحة للمصفوفة.

٢ - السلاسل الهجائية والرقمية:

تسمى مجموعات الحروف الابجدية أو الابجدية الرقمية ، التى تعالج على الحاسب كوحدة واحدة سلسلة STRING.

أمثلة:

" أحمد ذهب الى المدرسة "

" مرفت حصلت على الليسانس "

" + أ ب ٦٥٣ % س ص ع "

ولتمييز هذه السلاسل تفترض لغة مثل البيزيك استخدام علامة الدولار $ فى نهاية اسم المتغير المناظر لعنصر البيانات ، كما تستخدم لغات البرمجة الأخرى اساليب مختلفة فى هذا الشأن.

وقد يجرى ضم هذه المسلسلات بعملية ضم CONCATENATED مثل:

A $ = " زواج "

B $ = "... "

C $ = " سعيد "

A$ + B$ + C$ = " زواج .... سعيد "

وبهذا تعامل سلاسل الحروف باعتبارها مصفوفة ذات طول متغير شأنها شأن السجلات فيما يوضحه شكل (١١/٣).

وقد تكون سلاسل الحروف ذات طول ثابت أو ذات طول متغير:

أ - طول ثابت يعنى أن طول السلسلة ثابت وتشغل حيز محدد من مواقع التخزين.

ب - ذات طول متغير فيما يوضحه الشكل (١١/٤) وتعنى أن لكل سلسلة طول مختلف عن الأخرى وكلا النوعان يستخدما كثيرا فى نظم المعلومات.

عنوان الموقع	١	٢	٣	٤	٥	٦	٧	٨	٩
المحتوى	ع	ل	ى		م	ح	م	د	
الشرح	السلسلة الأولى			السلسلة الثانية					

سلسلة ذات طول ثابت

الشكل (١١/٣)

عنوان الموقع	
المحتوى	

عنوان الموقع	
المحتوى	

سلسلة ذات طول متغير

الشكل (١١/٤)

obeikandl.com

٣ - السجلات:

تعتبر السجلات وحدة منطقية متكاملة تتركب من عدد من حقول البيانات المترابطة منطقيا وهى تشبه المصفوفة الاحادية وان اختلفت عنها فى عدة نواح أبرزها أن المصفوفات تضم عناصر بيانات ذات نوع واحد بينما تضم السجلات عناصر بيانات مختلفة وبذلك لايمكن فتح مصفوفة لاختزان سجلات بل تستخدم الجداول TABLES أو الملفات فى تخزينها.

٤ - القوائم:

سبق وأوضحنا أن للمصفوفات حيز ثابت لايتغير أو يتبدل مما يسهل مناظرة المصفوفة المنطقية بنظيراتها التخزينية ، ويعود هذا إلى جمود المصفوفات ، وعلى العكس منها تكون القوائم التى تتصف بالديناميكية ويتغير حيزها باستمرار ، ففى قائمة أعضاء نادى رياضى يضاف أعضاء جدد ويحذف أعضاء إنضموا الى نادى آخر أو ماتوا أو هاجروا الى أرض الله الواسعة ، وهذه الديناميكية تستدعى تسجيل القوائم فى الحاسب بأقل فاقد فى وسائط التخزين.

وتختلف القوائم عن مسلسلة الحروف فى امكان التعامل مع عناصرها كلا على حده وليس كتلة واحدة مما يستدعى استخدام المؤشرات POINTERS فى ادارتها.

وتنقسم القوائم إلى نوعين :

أ ـ قائمة ذات إتجاه واحد.

ب ـ قائمة دائرية (ذات الإتجاهين).

المؤشرات :

حتى نمهد للمؤشرات نسترجع الحقائق التالية:

أ - أن كل عنصر بيان يخزن فى موقع له عنوان.

ب - انه لاستعادة أى بيان يجب معرفة عنوانه.

ج - أن العنوان - هو الآخر - يخزن فى موقع آخر بالذاكرة يسمى المؤشر.

د - تتم استعادة البيان بالوصول الى عنوان البيان (المؤشر) ومنه نصل إلى البيان.

مثال:

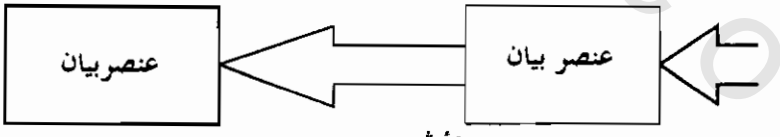
ناقش أساليب معالجة العبارة التالية:

(ايهاب لا يحب أكل الدجاج)

أ - يمكن أعتبارها هيئة مسلسلة ابجدية كما مسلسلة الحروف.

ب - يمكن معالجتها على هيئة قائمة LIST كل عنصر منطقى فيها يسمى DATUM ويتصل كل عنصر DATUM بالذى يليه بواسطة مؤشر POINTER نرمز له بسهم ←.

ج - يعتبر عنصر البيان DATUM والمؤشر POINTER مكملان لبعضهما الآخر أو وحده واحدة هى وحدة بناء القوائم.

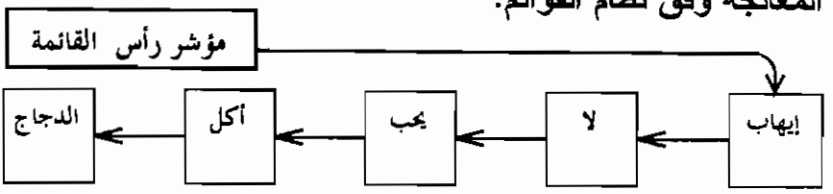


المعالجة على هيئة مسلسلة:

" إيهاب لايحب أكل الدجاج " = A$

obeikandl.com

المعالجة وفق نظام القوائم:



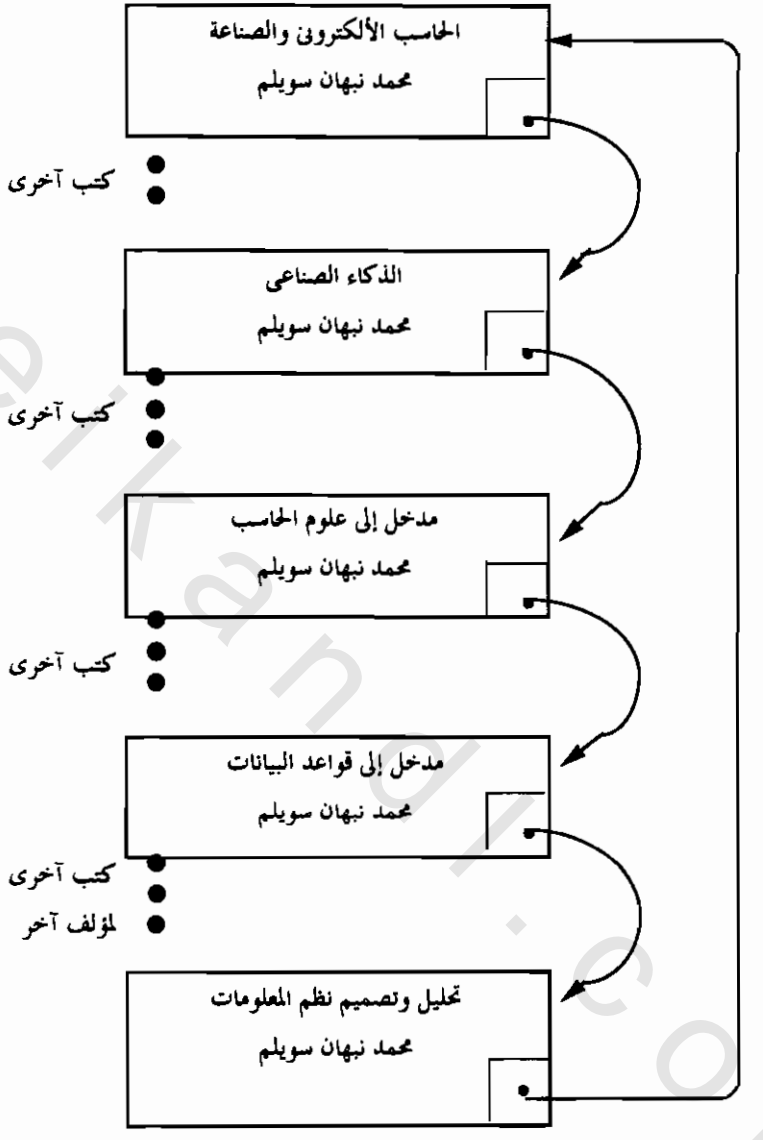
ونلحظ أن عنصر البيان (الدجاج) لايصحبه مؤشر مما يعتبر إشارة نهاية القائمة ، وبذلك لايمكن الرجوع إلى بداية القائمة ، وهذا يوضح لنا مفهوم القائمة ذات الإتجاه الواحد ، أما إذا إدرج مؤشر مع عنصر بيان الدجاج يشير إلى موقع كلمة إيهاب فإنه يمكن العودة مرة أخرى فى الاتجاه المضاد وينتج عن ذلك قائمة ذات إتجاهين (دائرية) الإتجاه.

أنواع القوائم واستخدامها :

أ ـ القوائم المتصلة Linked List :

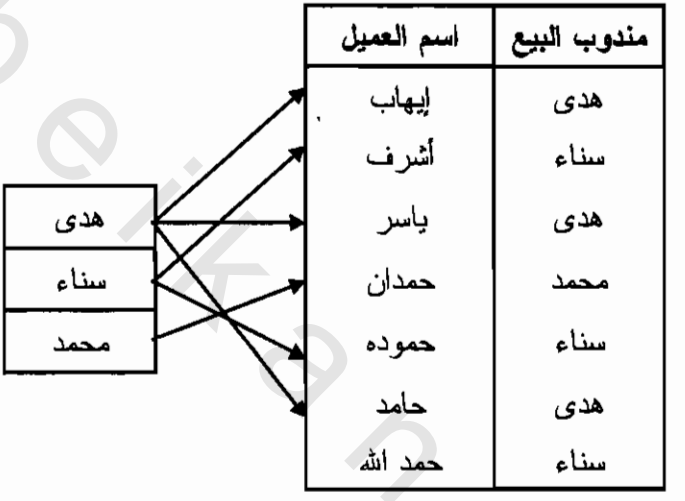
تستخدم المؤشرات فى بناء نظم معلومات أفضل ، فإذا افترضنا أن احدى المكتبات العامة تستخدم نظام القوائم لرصد بيانات مقتنياتها وحتى ترتقى المكتبة بأدائها تم فرز قائمة المقتنيات ابجديا حسب عنوان الكتاب ، وهذا النوع من القوائم لا يتيح تجميع كل مؤلفات كاتب بذاته لأن كتبه موزعة ابجديا على امتداد القائمة كلها ، مما يشكل صعوبة أمام المتردد على المكتبة إذا كان لايعرف عن الكتاب سوى أسم مؤلفه.

ولحل مثل هذه المشكلة وسواها يتم إضافة خلية تخزين اضافية (بايت واحدة) من نوع المؤشرات إلى سجل الكتاب ، يخزن بها عنوان الكتاب التالى لنفس المؤلف فيما يوضحه الشكل (١١/٥) ، ويطلق على هذا النوع من القوائم القوائم المرتبطة أو القوائم المتصلة LINKED LIST وهى تحقق رصد جميع عناصر أو سجلات البيانات المرتبطة منطقيا متى تم العثور على أول عنصر أو سجل.

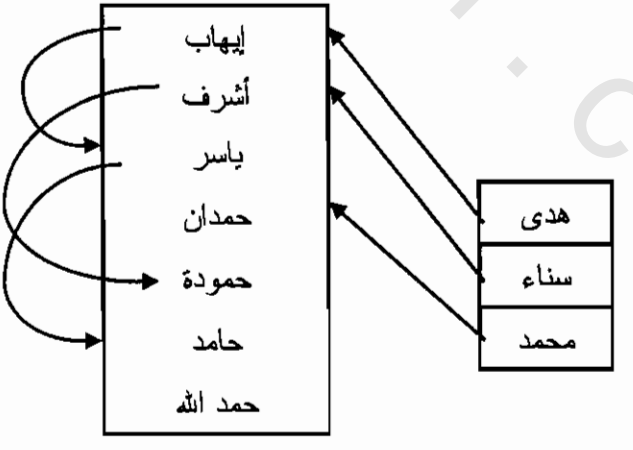

الحاسب الألكترونى والصناعة
محمد نبهان سويلم
كتب آخرى
الذكاء الصناعى
محمد نبهان سويلم
كتب آخرى
مدخل إلى علوم الحاسب
محمد نبهان سويلم
كتب آخرى
مدخل إلى قواعد البيانات
محمد نبهان سويلم
كتب آخرى
لمؤلف آخر
تحليل وتصميم نظم المعلومات
محمد نبهان سويلم

شكل (١١/٥)

كما يمكن استخدام القوائم فى الأعمال التجارية والإدارية على غرار مثـــال المكتبة فإذا فرضنا أن هناك شركة أدوات تجميل تبيع السلع عـــن طريـــق مندوبـــى المبيعات وكان الملف كما هو موضح بالجدول فيمكن إنشاء قوائم ومؤشرات تربـــط بين أسم مندوب البيع وعملائه مما يوفر حيز فى الذاكرة ويرفع كفاءة الإسترجـــاع



والوصول إلى البيانات ويمكن إجراء الربط بين عملاء كل مندوب بيع لزيادة مرونة النظام على النحو الموضح :



ب – قوائم كثيفة DENSE LISTS :

وهى التى تضم جميع عناصر القائمة فى مواقع تخزين متجاورة داخل الذاكرة ، فيما يشبه مصفوفة احادية ضخمة مما يشكل صعوبة فى التعامل معها خصوصا فى عمليات الحذف والاضافة . فإذا فرضنا أن لدينا قائمة بعشرة أسماء ونريد حذف الاسم الاول فيجب بعد الحذف تحريك جميع الاسماء للامام حتى يملىء الفراغ الذى نجم عن الحذف ، اما إذا كان المطلوب اضافة اسم للقائمة فإن الاضافة تواجه عدة صعوبات منها التأكد من وجود مكان خال والثانى تحريك الاسماء للخلف.

العمليات على القوائم:

تسمح القوائم المتصلة والتى سبق وعرضنا اليها فى مثالى المكتبة وشركة أدوات التجميل تسمح بإجراء العمليات التالية عليها بمرونة ويسر:

(١) إضافة عنصر بيان.
(٢) حذف عنصر بيان.
(٣) تحديد موقع (أو تواجد) عنصر بيان.
(٤) تحديد عنصر البيان التالى.
(٥) فرز القوائم وفق أى عنصر بيان.
(٦) إنشاء قوائم خالية.

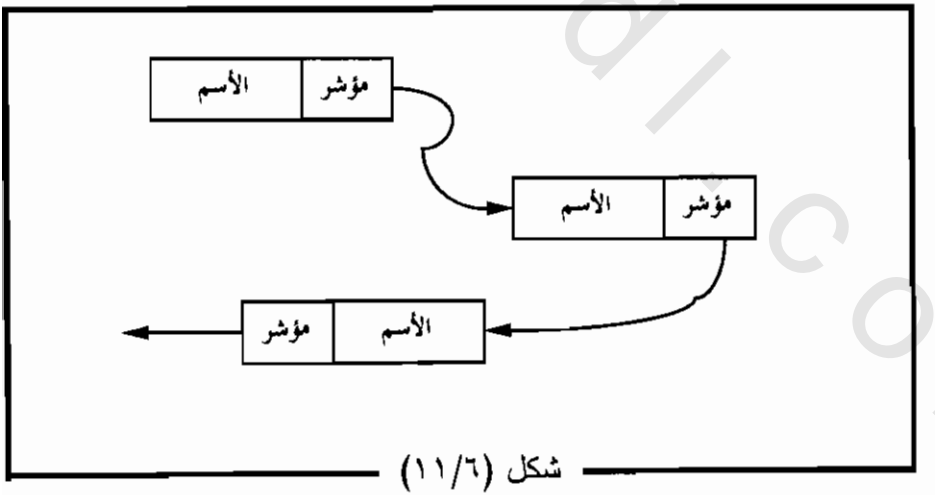
وترجع هذه المرونة إلى أن القوائم المرتبطة لاتحتل مواقع متلامسة فى داخل الذاكرة كما فى المصفوفات أو القوائم الكثيفة ، ويجرى اعداد هذا النوع من القوائم باضافة حقل تخزين جديد إلى كل سجل – يعمل كمؤشر – يشير الى عنوان تخزين السجل اللاحق ، وحتى يتم الوصول الى القائمة ذاتها فإنه يتم تخزين عنوان أول سجل فى

القائمة فى خلية تخزين ويسمى هذا الموقع مؤشر رأس القائمة أو مؤشر القائمة ويوضح الشكلان (١١/٦) (١١/٧) تلك المفاهيم.

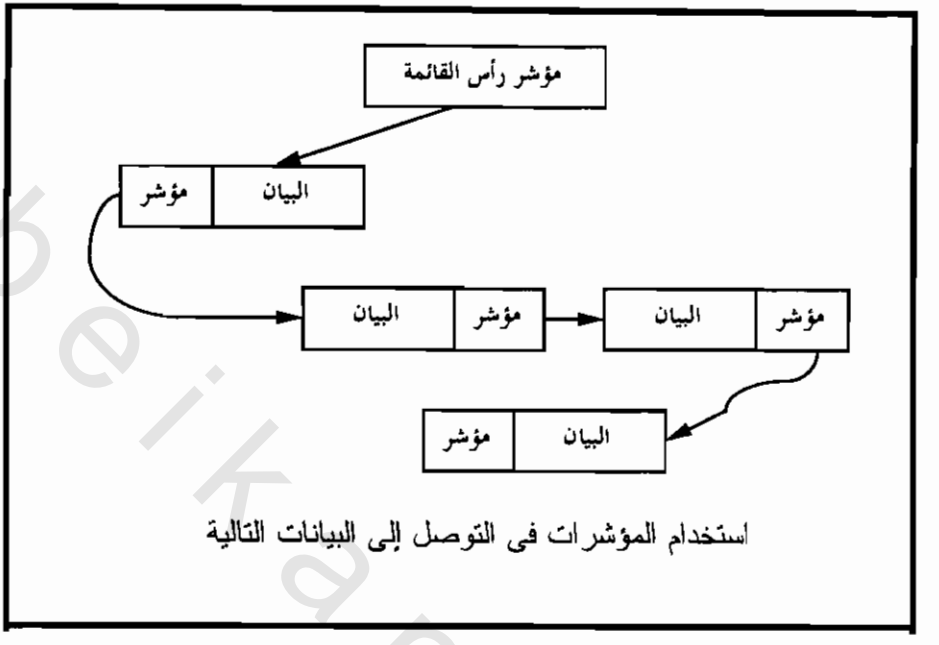
هنا يتبادر الى الذهن سؤال كيف يعرف الحاسب نهاية القائمة أو لازال هناك مزيدا من السجلات ، ويحل هذا الاشكال تخصيص موقع تخزين ثنائى ضمن المؤشر فإذا كانت قيمة البت (واحد) تعنى لازال هناك بيانات لاحقة ، اما اذا كانت البت صفرية القيمة دل ذلك على انتهاء القائمة فيما يوضحه الشكل (١١/٨).

الاضافة والحذف من القوائم:

يمكن حذف أى بيان من القوائم بتعديل قيمة المؤشر القديم الى قيمة جديدة على النحو الموضح فى الشكل (١١/٩) وبالتالى لاتظهر البيانات عند استرجاع أو طبع القائمة . فى حين تتم الاضافة على القائمة بالعثور أولا على موقع خال مع اضافة عنوان (مؤشر) يوضح البيانات السابقة للبيانات المضافة مما يجعل هناك مدخل جديد للحصول على البيانات فيما يوضح فى الشكل (١١/١٠).



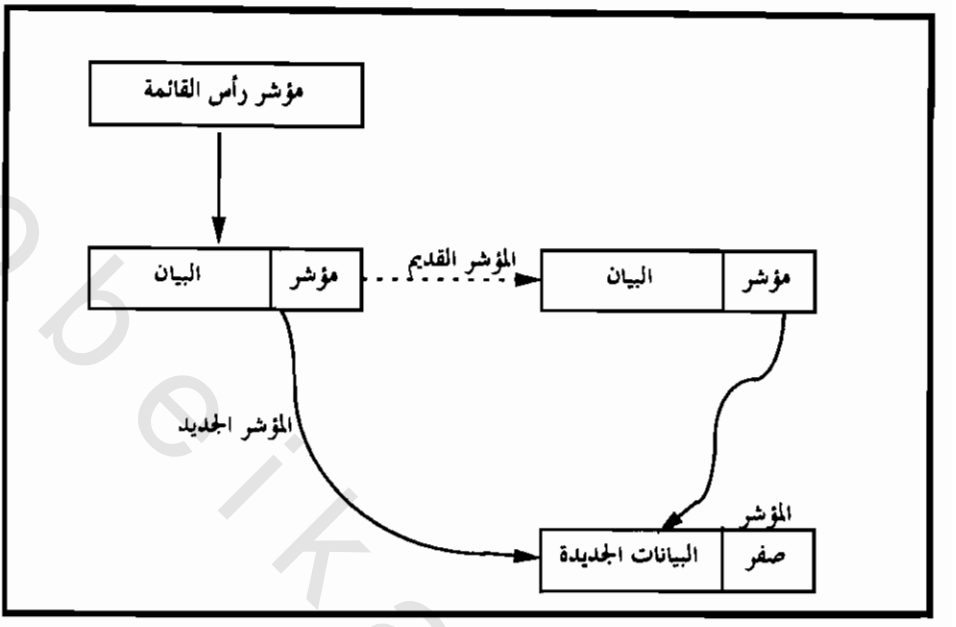
شكل (١١/٦)



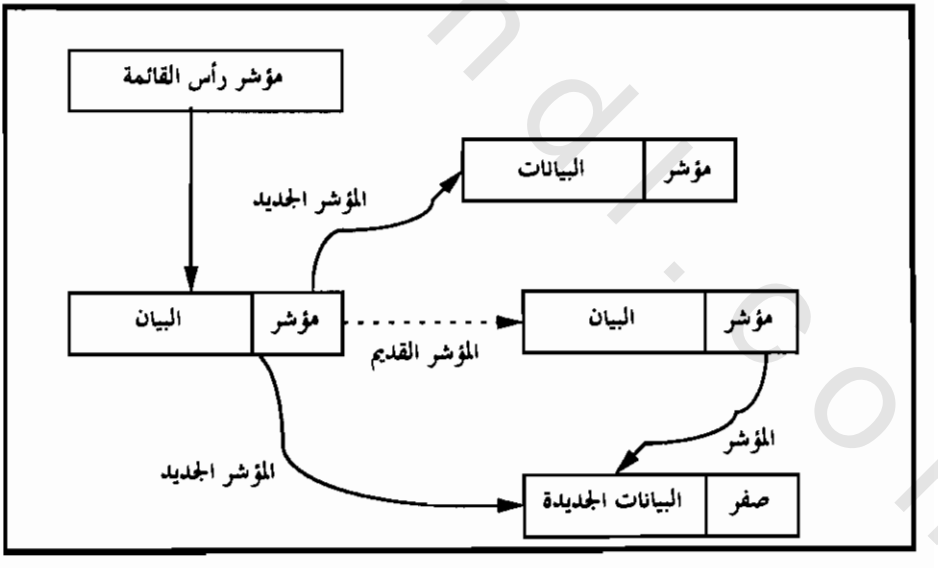
شكل (١١/٧)

شكل (١١/٨)

obeikandl.com



شكل (١١/٩) يوضح حذف سجل أو بيان من القائمة



شكل (١١/١٠) تحديث بيانات أو سجلات قائمة

٥ - الكومة STACK :

يأتى مفهوم الكومة من ذات مفهومها فى الحياة اليومية ، فإن كانت خزينة الكتب مملوءة (كومة) فأى كتاب يوضع أعلاها سوف يكون أول كتاب يؤخذ منها عند البحث عن كتاب آخر ، بمعنى من يدخل أخيرا يخرج أولا LAST IN FIRST OUT (LIFO) وتناظر الكومة القوائم فيما عدا أن اضافة أو حذف أى سجل بيانات يتم عند طرف واحد منها مما يسهل عمليات الاضافة والحذف وتكاد تلغى تماما صعوبة التعامل مع القوائم ، ويحجز للكومة حيز متصل داخل الذاكرة أكبر قليلا من الحيز المطلوب حتى تستوعب المواقع الزائدة عمليات الاضافة أو الحذف وهذا التمدد يحتاج مؤشر يشير الى الموقع الخالى عند احد طرفيها.

٦ - قوائم الانتظار QUE :

تشبه قوائم الانتظار فى الحاسبات قوائم الانتظار فى الحياة اليومية عندما يقف الناس على هيئة طابور لشراء سلعة أو انجاز عمل ، ويطبق فى كلاهما مبدأ من يأتى أولا يخدم أولا ، وفى الحاسب يطبق نفس المبدأ (من يدخل أولا يخرج أولا) ، ففى نظم الحاسبات قد تقذف وحدة التشغيل المركزية بمخرجات متدفقة أسرع ملايين المرات من سرعة وحدات الطباعة مما يستدعى تخزين المخرجات على الأقراص المغناطيسية (مكان انتظار) حتى يأتى دورها فى الطباعة ، وهنا يختلف الأمر فعناصر البيانات لاتتحرك مثلما يتحرك الناس فى طابور السينما بل يجب تحريك مؤشرات على قائمة الانتظار.

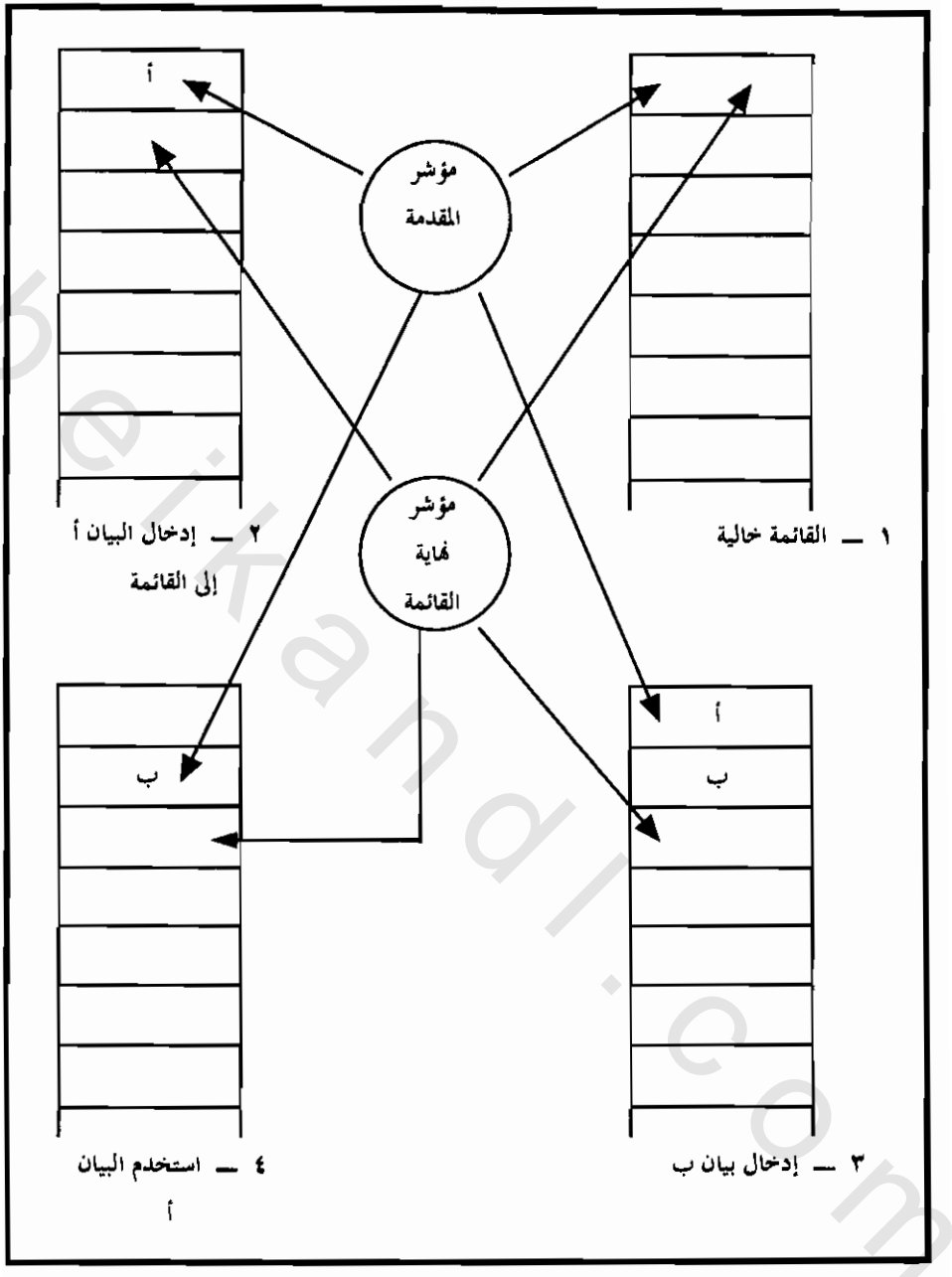
تنفيذ قوائم الانتظار:

يتطلب تنفيذ قوائم الانتظار على الحاسب حجز حيز فى الذاكرة يستوعب البيانات مع وجود مؤشرين ، يتولى الأول الاشارة الى رأس موقع الانتظار بينما

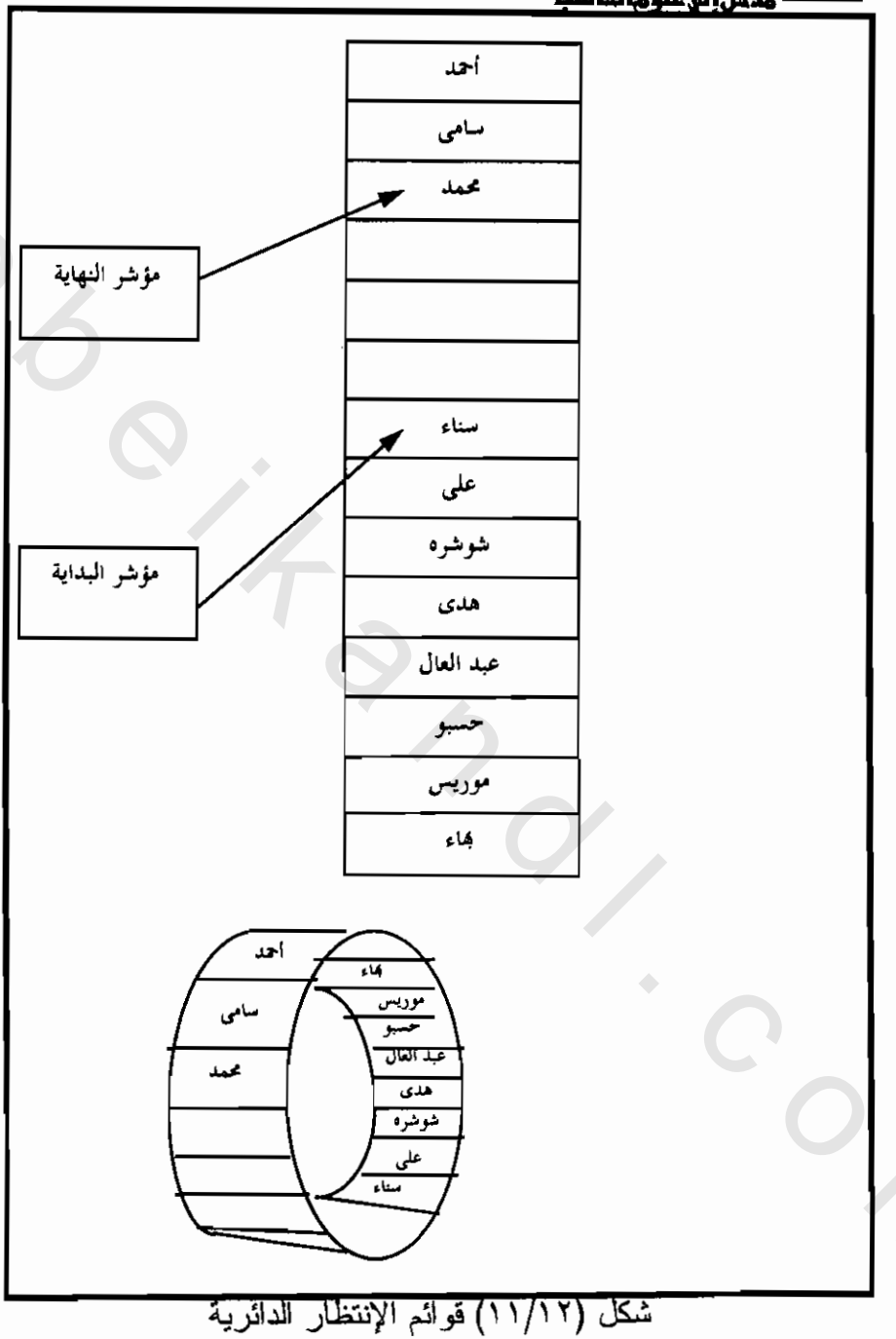
يشير الآخر إلى نهاية موقع الانتظار TALL & HEAD POINTERS ويشير مؤشر رأس القائمة الى عنوان أول موقع مشغول بينما يشير مؤشر ذيل القائمة الى عنوان أول موقع خال فى كتلة المواقع المعنونة المخصصة للقائمة . ويوضح الشكل (١١/١١) عن حركة التعامل مع قوائم الانتظار.

ونلحظ فى الشكل الفرعى [١] أن القائمة لازالت خالية من البيانات مما ترتب عليه أن مؤشرى مقدمة ونهاية القائمة يشيران إلى موقع التخزين الأول كبداية ونهاية للبيانات ، بينما فى الشكل الفرعى [٢] دخل عنصر البيان [أ] ونلحظ أن مؤشر نهاية القائمة تزحزح إلى الموقع الفارغ التالى بعد موقع [أ] ، وفى الشكل الفرعى رقم [٣] تم إدخال عنصر البيان [ب] وتحرك مؤشر نهاية القائمة إلى الموقع رقم [٣] وفى الشكل الفرعى [٤] تم خروج عنصر البيان [أ] مما أستوجب من مؤشر رأس القائمة الإشارة إلى موقع البيان [ب] كبداية لقائمة الإنتظار.

وهذا التحرك المستمر فى عناوين مواقع الانتظار قد يجعل كتلة القائمة تنتقل عبر الذاكرة مدمرة فى طريقها كل البيانات وهذا الزحف لايعود إلى حيز القائمة لكن إلى تغير قيم المؤشرات باستمرار مما دفع الى التفكير فى قائمة انتظار حلقية أو دواره هى أقرب ما تكون إلى دورات البرامج LOOPS بأن تتحرك البيانات المخزنة فى القائمة ناحية راس القائمة باستمرار كلما خرج بيان منها مثلما نتحرك للامام فى طابور تذاكر السينما كلما خرج احد المنتظرين من الصف بعد حصوله على التذاكر ، وفى هذه الحالة فإنه اقتربت البيانات المخزنة فى القائمة من نهايتها دفعت بيانات جديدة من ذيل القائمة فيما يوضحه الشكل (١١/١٢) ، ومثل هذه القوائم الانتظارية الدوارة تصلح فى انشاء المناطق العازلة للمدخلات والمخرجات.


مؤشر
المقدمة
مؤشر
نهاية
القائمة
١ ــ القائمة خالية
٢ ــ إدخال البيان أ
إلى القائمة
٣ ــ إدخال بيان ب
٤ ــ استخدم البيان
أ
أ
ب
ب

شكل (١١/١١)



شكل (١١/١٢) قوائم الإنتظار الدائرية

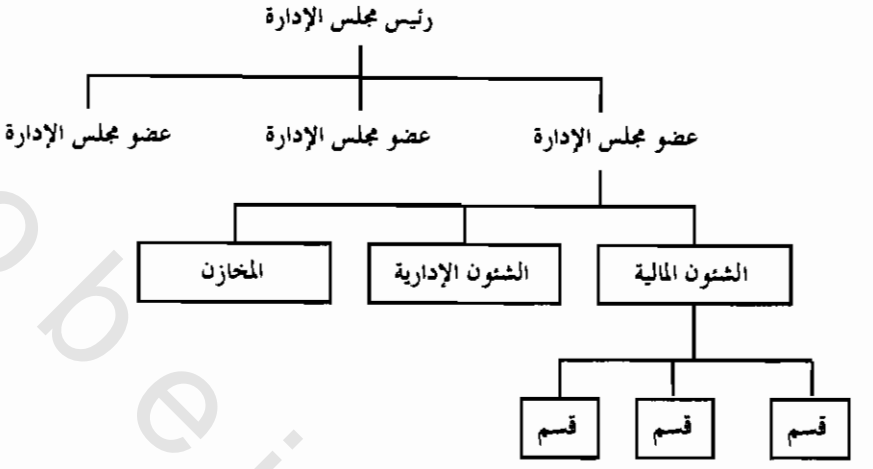
تطبيقات قوائم الانتظـــار:

تستخدم قوائم الانتظار فى عدة تطبيقات داخل نظام التشغيل خاصـــة فـــى ادارة المشغل كما تطبق فى برامج الفرز ، فاذا فرضنا أن طلبة كلية التجارة مسجلين فـــى الحاسب ضمن قوائم موحدة سواء كانوا فى الدراسات العليا أو مرحلة البكـــالوريوس ويراد اصدار طباعة مستقلة لكلا القسمين فإن هناك عدة حلول لتنفيذ المطلوب منـــها فرز القائمة مرتين ، مرة على حقل الدراسات العليا ومرة أخرى على حقل مرحلـــة البكالوريوس ، وفى هذا مضيعة شديدة لطاقـــة الحاســب والأفضــل فــرز طلبــة البكالوريوس واصدار طباعة فورية لهم مع تخزين بيانات طلبة الدراسات العليا فـــى قائمة انتظار لحين طباعتها.

ثانيا : شجرة البيانات TREE :

تشبه هياكل البيانات الشجرية التمثيل المرسوم لشجرة عائلة ضخمة أو الـــهيكل التنظيمى لاحدى الشركات أو المؤسسات حيث يوجد مدير الشركة على راس التنظيم ثم يمتد خط الى النائب بعدها يتفرع التنظيم الى الادارات والأفرع والأقسام... وهكذا فيما يوضحه الشكل (١١/١٣) لهيكل تنظيمى لاحدى الشركات.

ويشترط فى هذا الشكل الهرمى عدم وجود مستوى أدنى يعمل تحت قيـــادة رئيسين فى وقت واحد ، كما فى شجرة العائلة حيث لايوجد ابن له والدان (عـــدد ٢ والد).



شكل (١١/١٣)

والعناصر البنائية فى التركيب الشجرى تسمى NODES وكل عنصر يتكون من وحدة بيان ومشيرين على الأقل ويسمى NODE الأعلى ROOT NIDE ، الأخـــر LEAF والخط الواصل بينها هى الأقواس ARC.

مؤشر يمين	عنصر بيان	مؤشر يسار

عنصر بيان الهيكل الشجرى

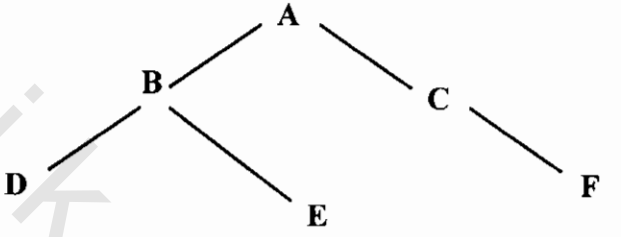
وغير خاف أنه لايوجد داخل الذاكرة يمين أو يسار ، انمـــا تســـتخدم هـــذه الطريقة لتوضيح شكل التخزين على الورق وبالتالى فإن كل عقدة سوف تمثـــل فـــى الذاكرة بواسطة مجموعة من خلايا التخزين متجاورة واذا كان قيمة أحد المؤشـــرات صفرا NIL فهذا يعنى عدم وجود امتداد اما اذا كان المؤشرين NIL فهذا يـــدل علـــى

obeikandl.com

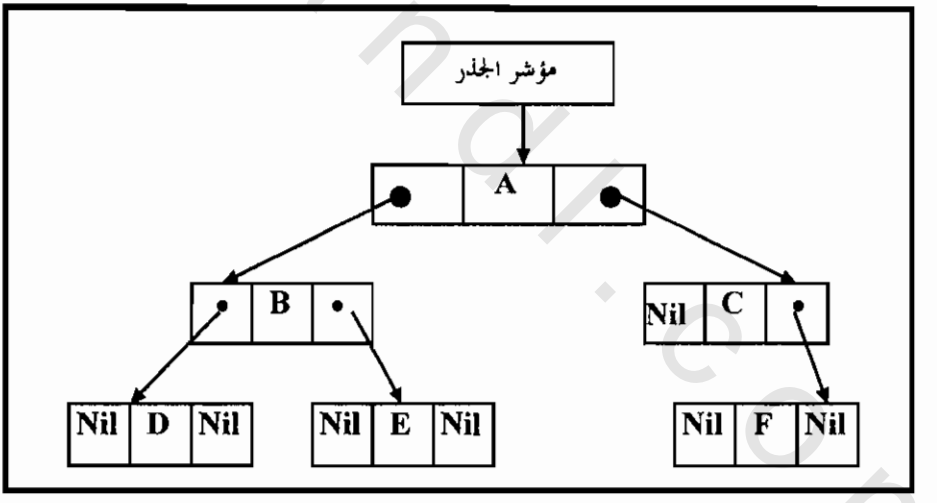
نهاية تركيب البيانات ، اما المدخل الرئيسى للشجرة فيخزن فى مؤشر الجذر ROOT POINTER .

مثال:

المطلوب تمثيل العلاقة التالية فى هيئة شكل تخزين داخل الحاسب؟



الحل: يوضحه الشكل (١١/١٤).



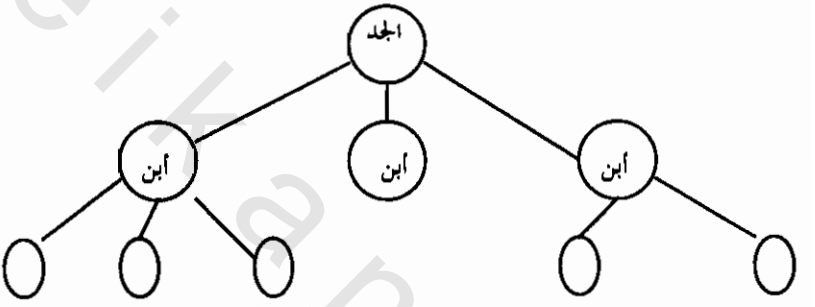
شكل (١١/١٤)

وهذا التركيب يتيح تحديد جذر السجلات ROOT لكل فرع LEEF لكن على حساب استخدام مواقع تخزين أكثر للمؤشرات.

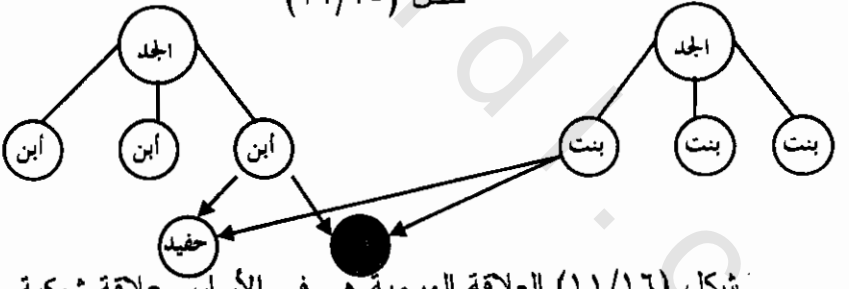
obeikandl.com

ثالثا : هياكل البيانات الشبكية :

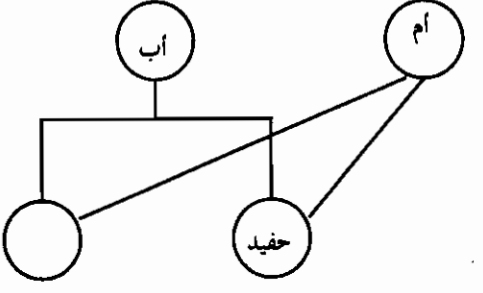
إذا أتصل أى عنصر بيان فى المستوى الإدنى من هياكل البيانات الشجرية بأكثر من عنصر فى مستوى أعلى فيطلق مسمى هياكل بيانات شبكية ، حتى شجرة العائلة من النوع الشبكى وليست من النوع الهرمى فالشكل (١١/١٥) يوضح العلاقات الهرمية لأسرة تتكون من جد ــ أبناء ــ أحفاد ، شكل غير حقيقى لأننا أهملنا الأمهات منذ زمن بعيد لكن الشكل (١١/١٦) وهى علاقة يمكن توضيحها أكثر كما فى شكل (١١/١٧).



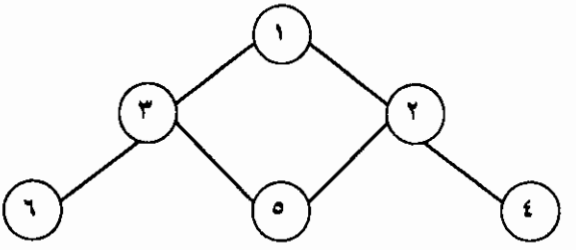
شكل (١١/١٥)



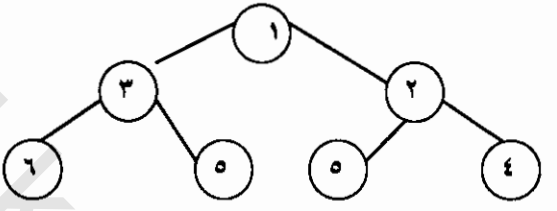
شكل (١١/١٦) العلاقة الهرمية هى فى الأساس علاقة شبكية



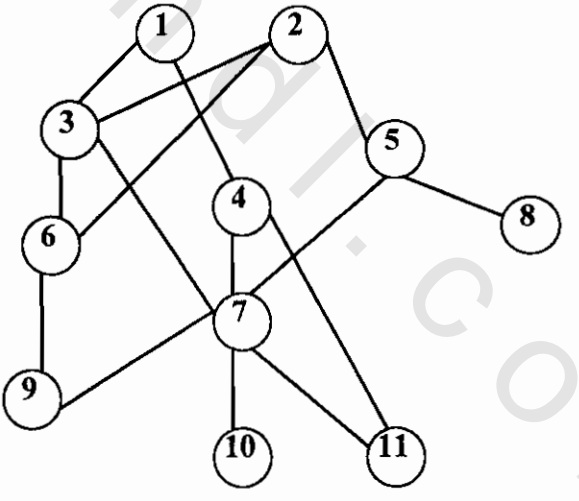
شكل (١١/١٧)



تكافئ الشكل الشجرى التالى



وتنقسم هياكل البيانات الشجرية إلى نوعين ، بسيط ومعقد ، ففى النوع البسيط يمكن تحديد مستويات الهيكل البنائى للبيانات أما فى النوع المعقد فيصعب ذلك كثيرا كما فى الشكل (١١/١٨).



شكل (١١/١٨)

١٢ - قواعد البيانات

تعتبر قواعد البيانات احد الحلول الجيدة والمكلفة لمشاكل بيئة نظم الملفات وابرزها مشكلة تكرارية البيانات وانعكاساتها السلبية على استخدام وسائط التخزين وتضارب المعلومات وما يستتبع ذلك من تكاليف لحفظها وتشغيلها وصيانتها ناهيك عن أن تحديث أى عنصر بيان لا يعنى تحديثه على مستوى النظام بل يقتصر على الملف المعنى بهذا التحديث مما يسبب عدم تكاملية البيانات وعدم امكان فرض ادارة مركزية وسيطرة امنية تقى المعلومات من مخاطر التدخل فيها أو الاخلال بها أو سرقتها.

وعلى ضوء هذا المنظور يمكن تعريف نظام قواعد البيانات على انه "تجمع منظم لسجلات البيانات والبرامج وذلك بأقل قدر ممكن من التكرارية وأكبر قدر متاح من التكاملية مع اتاحة المشاركة على البيانات لمختلف المستخدمين دون أدنى ارتباط بين البيانات وبين برامج التطبيقات".

مميزات قواعد البيانات:

١ - امكان اضافة ملفات جديدة.

٢ - اضافة بيانات جديدة على الملفات الموجودة فى القاعدة.

٣ - استرجاع بيانات من الملفات المكونة لقاعدة البيانات.

٤ - تحديث البيانات.

٥ - حذف البيانات من الملفات.

٦ - ازاحة ملفات خالة أو مكتوب عليها مسجلات.

٧ - يمكن تعديل البرامج دون تعديل البيانات والعكس صحيح.

٨ - يمكن للمستخدم النظر اليها على أنها ملفات متكاملة.

٩ - تلبى حاجات كافة (معظم) المستخدمين للبيانات.

١٠ - يمكن فرض قيود التأمين والسرية على بعض البيانات الهامة.

١١ - تحقق المرجعية على الملفات.

١٢ - امكان تخليق بيانات جديدة من البيانات الموجودة على الملفات.

وبذا تتلافى معظم عيوب بيئة نظم المعلومات المرتكنة على الملفات.

مكونات نظام قاعدة البيانات :

يتكون نظام قاعدة البيانات من اربعة مكونات اساسية هى:

١ - **البيانات :**

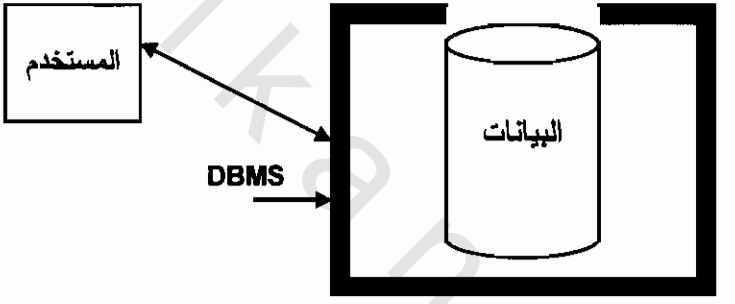
تتوافر قواعد البيانات على الحاسبات الصغيرة والشخصية كما تتوافر على الحاسبات الكبيرة وتعتمد كفاءة النظام على قدرة وامكانيات الكيان الآلى للحاسب ، فالحاسبات الشخصية توفر قاعدة بيانات لمستخدم واحد بينما الحاسبات الكبيرة توفر قاعدة بيانات لعدد من المستخدمين يشاركون على البيانات المتاحة ، يجب أن تتصف البيانات بالتكاملية وعدم التكرارية وامكانية المشاركة عليها ، وعموما دون بيانات دقيقة ومنطقية وصحيحة فلا معنى لقاعدة البيانات.

٢ - **المعدات :**

ترتكن قواعد البيانات على الاقراص المغناطيسية ارتكانا كبيرا علاوة على وحدات من الشرائط الكثيفة كوحدات BACK UP احتياطية لتخزين البيانات للظروف الطارئه.

٣ - **البرامج :**

وهى الطبقة الوسيطة بين البيانات المخزنة فى الملفات أعلى الاقراص وبين مستخدمى قاعدة البيانات ، وابرز هذه البرامج برنامج مدير قاعدة البيانات DBMS وهو عبارة عن برنامج بالغ التعقيد باهظ الثمن ويحتاج من مختص قاعدة البيانات قدرا عليا ومعرفيا كبيرا للتعامل اذ انه يتولى السيطرة على العناصر الآلية والبرمجية للقاعدة بالتعاون مع نظام التشغيل فيما هو مبين بالشكل (١٢/١)٠



شكل (١٢/١)

٤ - **مستخدمو قواعد البيانات :**

وينقسمون الى ثلاث فئات :

أ ــ مخططو البرامج الذين يكتب برامجهم ويستخدمون امكانيات قاعدة البيانات.

ب ــ مختصو قواعد البيانات ، وهم المسئولون عن صيانة وتشغيل قاعدة البيانات٠

ج ــ المستخدمون لقواعد البيانات الذين يتعاملون مع قاعدة البيانات عبر النهايات الطرفية٠

واجبات برنامج

مدير قواعد البيانات DBMS

DATA BASE MANAGEMET SYSTEM

مدير قواعد البيانات هو احد البرامج المعقدة التى تنتجها شركات الحاسبات ويتولى انشاء وتوسيع وصيانة قاعدة البيانات ويقوم بدور الوسيط بين البيانات ومستخدميها ويشرف على ادارتها ، ويوزع مناطق التخزين على البيانات وينشأ لها الفهارس والمؤشرات اللازمة لاسترجاعها ، كما يمكن تغيير الشكل البنائى لها ، كما يتيح عمليات اضافة أو حذف أو تعديل السجلات ، كما يقوم بدور الوسيط بين البرامج وبين البيانات ٠

الى جانب ذلك يمكنه القيام بالاعمال التالية :

أ ــ انشاء قاعدة بيانات جديدة وادارة قواعد البيانات الحالية ٠

ب ــ معالجة السجل المطلوب بمفرده ضمن أى ملف ٠

ج ــ استرجاع السجلات سجلا سجلا ٠

د ــ يقى البيانات من الدخول عليها من شخص غير ذى صلاحية ٠

هـ ــ حماية البيانات ضد التخريب ٠

و ــ وضع نقاط ارشادية تستخدم فى حالة عطل الآلات أو عطل البرامج مما يساعد على سهولة استئناف العمل دون العودة الى بداية الملف ٠

ز ــ رصد الحركة على البيانات احصائيا ٠

ح ــ تسجيل كل تعامل على البيانات ٠

ط ــ وضع البيانات الهامة التى يشتد عليها الطلب فى مواقع ذات اسبقيات ٠

ى ــ يحتفظ بقاموس بيانات شامل أى بيانات عن البيانات، والمستخدمين ، والصلاحيات المتاحة لكل مستخدم ٠

واجبات مختص قواعد البيانات :

DATA BASE ADMINSTRATOR

الاشراف على قاعدة البيانات وظيفة هامة فى مراكز المعلومات ويتولاها أفراد ذوى كفاءة ومقدرة اذ عليهم يتوقف مايلى:

١ - ضبط البيانات المخزنة فى قواعد البيانات بحيث تلبى حاجات مستخدمى المعلومات.

٢ - يتولى الاشراف على اصدار التقارير المطلوبة فى النظام.

٣ - تحقيق أمن وسلامة البيانات وقواعد البيانات .

٤ - الاشراف على اضافة البيانات الجديدة وتحديث البيانات القديمه.

٥ - التحقق من عدم تكرارية البيانات.

٦ - التحقق من تكاملية البيانات.

مزايا قواعد البيانات فى اتخاذ القرار:

تغطى قواعد البيانات عدة مزايا لا يحققها نظام الملفات لمتخذ القرار منها:

أ - تقدم للادارة تقاريراً مبنية على معلومات محدثة شاملة مما يساعد على اتخاذ قرارات سليمة عكس نظام الملفات المرتبط ببيانات محددة.

ب ـ تقدم للادارة الوسطى تقاريراً مفصلة جيدة يصعب الحصول عليها من نظام الملفات.

ج ـ تحقق وفر فى التكلفة نتيجة عدم تكرارية البيانات.

د ـ توفر الجهد المبذول فى ادخال البيانات نتيجة توحيد المدخلات لكل نظام فرعى.

هـ ـ البساطة الشديدة فى استخدام لغة الاستفسار لان مدير قاعدة البيانات يتولى مهام التعامل مع البيانات.

و ـ الاستجابة السريعة لاحتياجات المستخدمين .

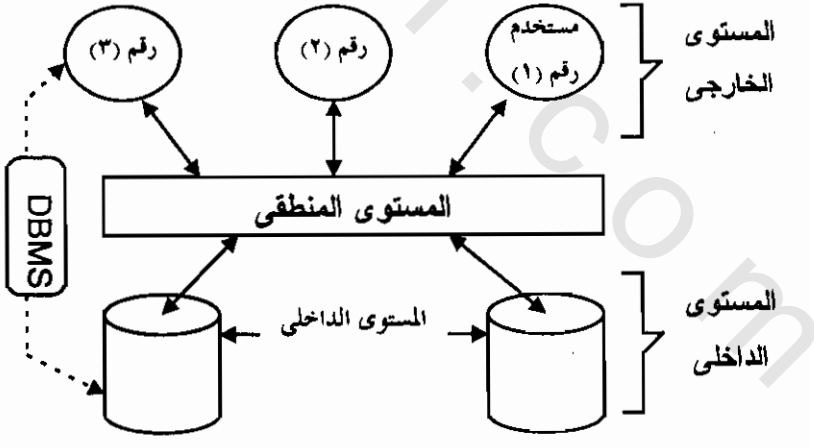
ز ــ الاقلال من عدد الأفراد العاملين فى مركز المعلومات ٠

ح ــ ادارة جيدة للبيانات حيث تحفظ البيانات فى مكـــان مركـــزى موحـــد لكـــل المؤسسة أو المنظمة٠

ط ــ الاسترجاع المتعدد MULTIPLE ACCESS باستخدام أساليب بسيطة نسبيا ومن خلال استخدام مفاتيح (حقول خاصة)٠

التركيب البنائى لقاعدة البيانات :

يمكن تصور التركيب البنائى لقواعد البيانات على ثلاثة مستويات ، مستوى اقـــرب الى المستخدم هو المستوى الخارجى حيث يتعامل المستخدم مع البيانات المخزنة، لذا يركز هذا المستوى على كيفية عرض البيانات للمستخدم ، اما المستوى الثانى فـــهو المستوى الداخلى ويركز على كيفية تخزين البيانات على الاقـــراص٠ وفيمـــا بيـــن مستوى المستخدم ومستوى الملفات يوجد مستوى ثالث هو المستوى المنطقـــى فيمـــا يوضحه الشكل (١٢/٢) ويتضح من الشكل عدم وجود ارتباط مباشر بين المستويات الثلاث فى تركيب قاعدة البيانات مما جعل الفصل بين هذه المستويات امـــرا ممكنـــا واكد للمستخدم ان بامكانه تغيير برامجه دون تغيير قاعدة البيانات واكد أن العكـــس ممكن فلا علاقة بين هذا وذاك.



شكل (١٢/٢) المستويات الثلاث لقاعدة البيانات

وفيما بين كل مستوى والذى يليه يوجد نوع من التطابقMAPPING لتوصيف البيانات بين المستوين للتأكد من وجود البيانات وامكان العزل بين البيانات المخزنة وبين مستوى المستخدم وفك أى ارتباط بينهما وهذه احدى مميزات نظم قواعد البيانات.

انواع قواعد البيانات :

توجد أنواعا كثيرة من قواعد البيانات لكن أبرزها واظهرها على الساحة ثلاثة أنواع رئيسية هى :

أ ــ قواعد بيانات هرمية التركيب HIERARCHICL DATA BASE.

ب ــ قواعد بيانات شبكية NETWORK DATA BASE.

ج ــ قواعد بيانات علائقية RELATIONAL DATA BASE.

والنوع الأخير أصبح الاكثر استخدما وشيوعا وتعتبر قاعدة البيانات , DBIV +DB 111 أقرب قواعد بيانات الحاسبات الصغيرة الى قواعد البيانات الكبيرة ، وسوف نركز دراستنا على هذا النوع.

قواعد البيانات العلاقية :

تعتبر قواعد البيانات العلاقية من أهم أنواع قواعد البيانات وتتصف بأنها قاعدة بيانات يستقبلها المستخدمون على هيئة جداول وليس شيئا آخر سوى الجداول فيما يوضحه الجدول (الملف) المعبر عن بيانات الاشخاص المسموح لهم بالاستعارة (ملف الاستعارة).

الرقم القومى	الأسم الأول	أسم الوالد	أسم العائلة	تاريخ الميلاد
٦٦٨٧٩٤	محمد	محمود	عليوه	١٩٤٠٠١٠٤
٢٥٦٦٧١٨	أيهاب	محمد	سويلم	١٩٧٦٠٧١٧
١٧٣٠٠٢٠	منال	إبراهيم	يسرى	١٩٧٠٠٧١٣
٧١٩٠٠٠	بطرس	سعد	مليكه	١٩٠٢٠٨٠٤

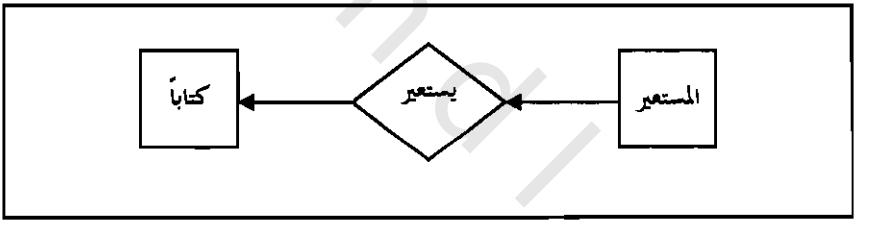
ملحوظة:

البيانات فى الجدول السابق بيانات افتراضية وليس لها أصل من الواقع.

ملف أوعية مكتبية:

رقم الإيداع	المؤلف	أسم الكتاب	الناشر	سنة النشر
٩١٨٤٠٦	جمال الغيطانى	الأدب والثقافة	الأهرام	١٩٧٣
٣٤٢٣١	أنيس منصور	حول العالم	النهضة	١٩٦٢
٣٥٦٢٣١	أنيس منصور	الذين هبطوا من السماء	الشروق	١٩٦٧

ويفرض أن شخصا له سجل فى ملف البطاقات الموضح فى الجدول الاول استعار كتابا من مقتنيات المكتبة فان العلاقة الناشئة عن هذا الاجراء يمكن توضيحها فى الاطار التالى:



وتعتبر قواعد البيانات العلاقية هذه العلاقة ملف جديد ـ جدول جديد ـ موصفاتة أى حقوله كالتالى :

* رقم المستعير (الرقم القومى).
* رقم ايداع الكتاب.
* تاريخ الاستعارة.

(يمكن الاستغناء عن حقل تاريخ الاستعارة فى هذا الملف)

وفيما يلى بعض الملاحظات :

١ - الجدول فى قواعد البيانات العلاقية يعادل الملف.

٢ - الاعمدة تناظر الحقول.

٣ - السطر يعادل السجل.

٤ - لكل جدول مسمى وحيد.

٥ - أن الجدول المسمى الاعارات نشأ بين العلاقة بين الفرد والكتاب.

٦ - أن لكل مسجل فى جدول المستعرين حقل مفتاح (الرقم القومى) كذلك فى جدول الكتب فان حقل المفتاح هو (رقم التسجيل).

٧ - أن رقم المستعير ورقم تسجيل الكتاب يمكن استخدامهما كحقل مفتاحى فى الجدول (الملف) المسمى الاعارات.

٨ - أن كل البيانات فى الجداول الثلاث بيانات ذرية لا يمكن تفتيتها لادنى من ذلك.

٩ - كل القيم معبر عنها صراحة وليس ضمنا.

١٠ - لا تستخدم مؤشرات احاله فيما بين الملفات.

لغة قواعد البيانات STRUCTURED QUERY LANGUAGE SQL:

تسمى لغة الاستعلام أو لغة البحث وتستخدم غالبا فى برامج مدير قواعد البيانات فى إنشاء أو تدمير الملفات كما تستخدم فى البحث الانتقائى لسجلات دون غيرها بدلا من استعراض كل الملفات للوصول الى السجلات المطلوبة ، وفى الحاسبات الشخصية تستخدم لغات مطورة من لغة البحث فى حزم البرامج المعروفة باسم قواعد البيانات مثل DB3+ , DB4 وما شابه من حزم برامج.

ويمكن استخدام لغة الاستفسار باسلوبين :

أ ــ اسلوب التعامل المباشر مع البيانات INTERACTIVE باستخدام التعليمات التالية:

- أنشأ جدول CREATE TABLE .
- أنشأ مشهد CREATE VIEW .
- أنشأ فهرس CREATE INDEX .

دمر / أسقط جدول / مشهد / فهرس DROP TABLE
DROP VIEW
DROP INDEX

ب ــ كما تستخدم عناصر لغة الاستعلام ضمن احدى لغات البرمجة الاجرائية مثل الكوبول ، وهذا يوضح لنا أن لغة الاستعلام ليست لغة اجرائية فهى توصف ماذا نريد WHAT وليس كيف ننجز ما نريد (HOW) حيث يتولى مدير قاعدة البيانات الخطوات الاجرائية دونما تدخل من المستخدم فيما يسمى السياحة الآلية فى النظام.

١ - لغة تعريف البيانات DDL :

وتنقسم لغة الاستعلام الى قسمين ..... قسم أو لغة تعريف البيانات (DDL) DATA DEFINITION LANGUAGE وهى التى اشرنا الى بعض اوامرها فى الفقرة السابقة.

أمثله:

أ ــ **إنشاء جدول** CREATE TABLE {TABLE NAME}

الرقم القومى حرفى ١٨ و ID CHAR (18) NOT NULL

الاسم الاول حرفى ١٢ و FNAME CHAR (12)

الاسم الثانى حرفى ١٢ و SNAME CHAR (12)

اسم العائلة حرفى ١٢ و TNAME CHAR

obeikandl.com

تاريخ الميلاد حرفى ٨ و BDATE CHAR (8)

راجع جدول بيانات المستعارين .

ب ــ غير الجدول

```
ALTER TABLE { TABE NAME }
ADD {COLUMN - NAME DATA - TYPE }
ALTER TABLE F
ADD PHONE CHAR 10.
```

فيضاف حقل جديد الى الملف (الجدول) عن رقم التليفون.

ج ــ تستخدم باقى الاوامر على النحو السابق مثل:

دمر الجدول DROP TABLE F

أنشأ فهرس فريد على حقل

الرقم القومى و CREATE UNIQUE INDEX XID CN ID (S#)

٢ - لغة تحريك وتدوال البيانات DML

DATA MANUPILATION LANGUAGE

وهى القسم الثانى من لغة الاستفسار ــ الاستعلام ــ وتستخدم اربعة اوامر لاغير ، ورغم محدودية عددها الا انها تحقق كفاءة عالية فى استرجاع وتحديث وتعديل البيان:

أ ــ الأمر	اختر	SELECT
	حدث	UP DATE
	أحذف	DELETE
	احشر	INSERT

امثله

عن استخدامات الاوامر على الملف المستعرين:

```
SELECT ID, FNAME , TNAME
MORF S
WHERE        DATE "19400104"
```


obeikandi.com

الإجابة :

٦٦٨٧٩٤ محمد محمود عليوه

```
UPDATE  { NAME TABLE }
SET  ID =  3668794
WHERE ID =  668794
```

إضافة سجل جديد

```
INSERT INTO  { TABLE NAME }
        VALUEES  (..... , ..... ,..... , ..... )
```

```
DELETE FORM { TABLE NAME }
WHERE  ID = "          "
       AND          ----------
       AND          ----------
                    ----------
       OR           ----------
```

خصائص قواعد البيانات العلاقية:

أ ــ كل ملف فى قاعدة البيانات العلائقية يضم نوع واحد متكرر من السجلات.

ب ــ ليس هناك ترتيب محدد للحقول.

ج ــ ليس هناك ترتيب محدد للسجلات ــ سيان فى قمة الجدول أو فى أى مكـــان آخر منه.

د ــ لكل حقل قيمة واحده فقط (لا تكرارية).

هـ ــ لكل سجل حقل مفتاح.

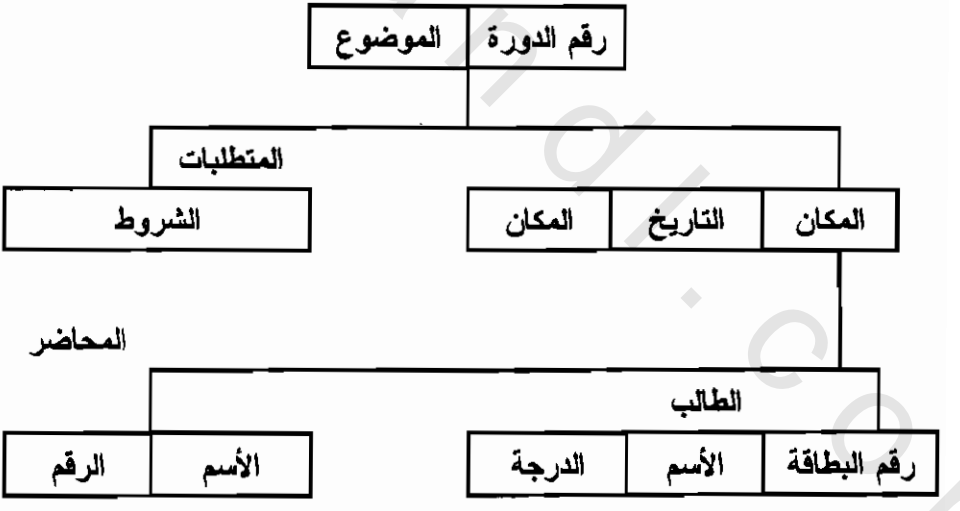
و ــ اوامر التعامل مع قاعدة البيانات لا تقتصر على الأربعة (اختار ــ حدث ــ أحذف ــ أحشر) انما تستخدم تعليمات اخرى مثل JOIN "صل " "أربط" وله شروط واساس رياضى فى التعامل مع هذا الامر.

ز ــ المنظر VIEW ليس ملفا مخزنا على وسائط التخزين بل هو ملـــف وهمـــى يتخلق من الملفات الكائنة ولا يمكن تحديثة أو انشاء فهارس عليه (الحقـــول)

{VIEW NAME} CREATE VIEW (الحقول) AS SELECT (أسم الجـــدول) FORM (شروط) WHERE .

قواعد البيانات الهرمية HIERARCHICAL D . B :

هى عبارة عن مجموعة مرتبة ومتكررة من نوع واحـــد مـــن السـجلات المركبة على هيئة شجرة ، أى أن لكل سجل جذر ROOT واحـد أى سـجل واحـد تتفرع منه سجلات أخرى وقد تتفرع هذه الفروع الى سجلات وهكـذا فيمـا يمكـن تشبيهه بشجرة العائلة (الجد ــ الابن ــ الابناء) ولا يسمح فى قواعد البيانات الهرمية بأن يكون لاى سجل اكثر من واحد مثال قاعدة بيانات مركز تدريب فيما يوضحـــه الشكل (١٢/٣).



الشكل (١٢/٣)

قواعد البيانات الهرمية

ونلحظ فى هذا النوع من قواعد البيانات أن بعض البيانات التى يمكن ابرازها فى قواعد البيانات العلائقية باستخدام حقول مدمجة تظهر فى هذه القاعدة بالروابط بين الاب والابن أى المستوى الاعلى والمستوى الادنى ٠ ويختلف تحريك أو تدوال معالجة البيانات في هذه القواعد عن العلاقية في أن تدوالها يتطلب:

١ - تسجيل جذر السجل ٠

٢ - وظيفة لتحريك البيانات من شجرة الى شجرة ٠

٣ - معامل للحركة بين كل سجلات هذا التكون الشجرى ٠

٤ - وظيفة لإضافة السجلات ٠

٥ - وظيفة لحذف السجلات٠

مثال عن اسلوب توصيف البيانات :

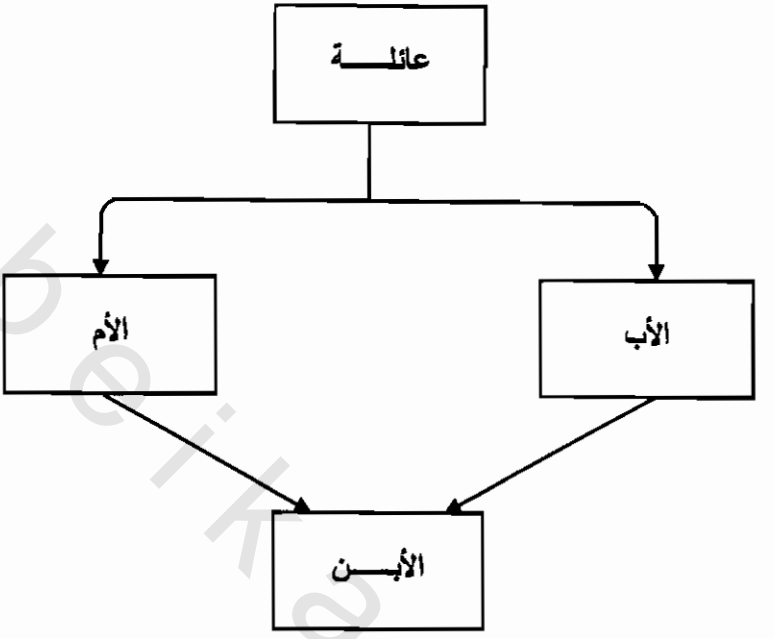
```
1 - DBD  NAME  =  EDUCP  DBD
2 - SEG  NAME  =  COURSE , BYTES  =  36
3 - FIELD NAME  = COURSE #, QES , BYTES =  3

4 - FIELD NAME = TITLE , BYTES = 33 START 4
5 - SEG
```

ويبدأ فى توصيف سجلاً جديد وهكذا.

قواعد البيانات الشبكية NET WORK :

وهى برامج DBMS نتعامل مع السجلات ذات الارتباط المتعدد وهى اقرب قواعد البيانات للواقع اذ أنه من الصعب أن تكون العلاقات الطبيعية فى الحياة على النظام الهرمى فقط ويمكن تلخيص هذا النوع من العلاقات بأن المستوى الأدنى قد يكون له أكثر من اتصال بالمستوى الأعلى فيما يوضحة الشكل (١٢/٤)٠

obeikandl.com



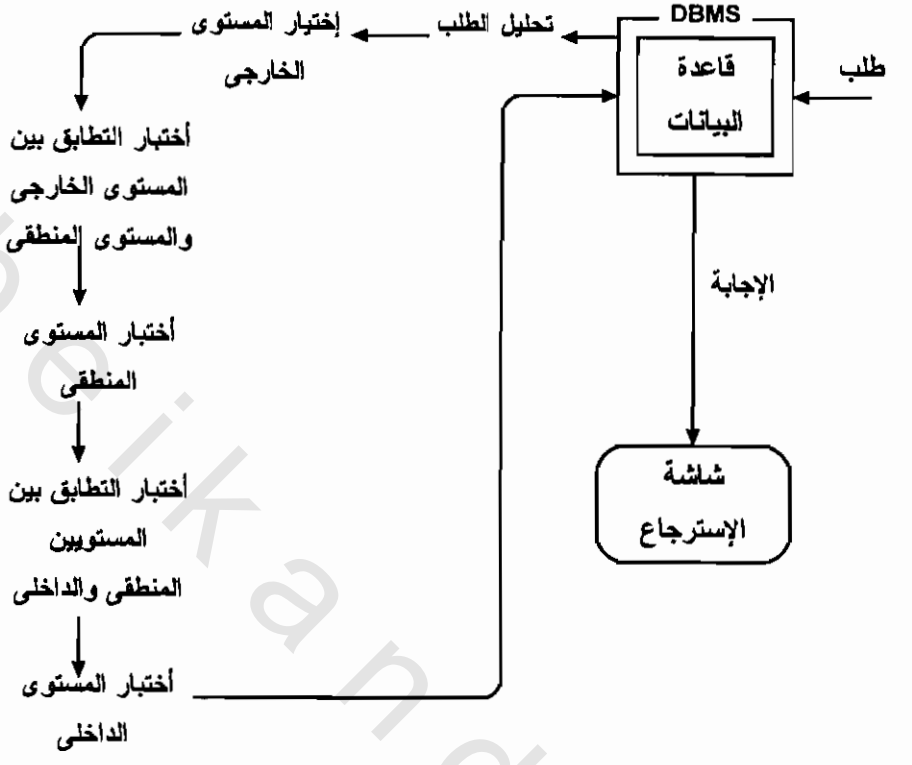
شكل (١٢/٤)

قواعد البيانات الشبكية

دور DBMS عند طلب الاسترجاع :

سبق وعرفنا أن DBMS أنه عبارة مجموعة برامج أو حزمة برامـــج يتـم اعدادها وكتابتها بواسطة الشركات المنتجة للحاسـبات أو شركـات كتابـة الكيـان البرمجي للحاسبات وتعفى مستخدمى قواعد البيانات من مهام معقـدة للولـوج الـى البيانات وتوصيفها وتحريكها وتخليق المستويات المختلفة ٠

ويتولى مدير قواعد البيانات وتحت اشرافه ادارة خطوات كثيرة ومعقدة فيما يوضحه الشكل (١٢/٥)٠



الشكل (١٢/٥)

يوضح خطوات الدور الذى تقوم به برنامج ادارة قاعدة البيانات

عيوب قواعد البيانات :

ان القائمة الطويلة التى عرضناها عبر هذا الفصل عن مزايا قواعد البيانات لا ينفى بحال من الاحوال أن انشاء قواعد البيانات له عدة عيوب نخلص اليها فـــى النقاط التالية :

أ ــ الحيز :

حتى تقدم قواعد البيانات كل خدماتها المميزة الى مختلف مستخدميها فان هذا يتطلب حيز هائل من وسائط التخزين الثانوية وذاكرة اساسية ذات حيز ضخم

فيما يضيف تكلفة مادية اضافية الى جانب جهد صيانة وتعديل وتحديث الملفات كما تتطلب معدات اضافية كثيرة ٠

ب ــ مشاكل الكيان البرمجى للقاعدة:

يتصف برنامج مدير قاعدة البيانات بأنه برنامج معقد يتطلب جهدا كبيرا فى استيعابه وفهمة من مسئولى نظام المعلومات حتى يستفاد بما عرضنا من مزايا٠

ج ــ التكلفة :

يعتبر برنامج مدير قاعدة البيانات مكلف فى حد ذاته فاذا اضفنا تكلفته الى باقى عناصر التكلفة (وسائط التخزين ــ الذاكرة ــ تدريب مسئولى النظام ٠٠٠) لاتضح ارتفاع ثمن مكونات قاعدة البيانات٠

د ــ توقف قاعدة البيانات:

عن العمل نتيجة مشاكل الكيان الى أو البرمجى لها فانها تؤثر على قطاع عريض من المستخدمين لا يسهل درأ اضراره المادية والمعنوية٠

هـ ــ برامج التأمين والدعم واستعادة التشغيل:

برامج مكلفة ماديا وصعبه التصميم وتحتاج افرادا على مستوى تعليمى وتدريبى مرتفع٠

على ضوء تلك الاعتبارات فللقراء أن يفكروا هل تنشأ قواعد بيانات ضخمه أم تنشأ قواعد بيانات تخصصية داخل المؤسسة الواحدة ؟

١٣ - قاعدة البيانات
DB3 +

يتطلب استخدام حزمه قاعدة البيانات +DB3 عددا من المتطلبات الفنية للحاسب الشخصى بأن يكون من انتاج IBM أو من الانواع المتوافقة معها بشرط أن يتوفر به مايلى:

أ ــ ذاكرة أساسية سعة ٢٥٦ كيلوبايت.

ب ــ نظام تشغيل DOS أو MS - DOS أو النوافذ.

ج ــ وحدتى قرص مرن أو وحدة مرنة وأخرى صلبة.

حدود قاعدة البيانات +DB3:

أ ــ تستطيع تشغيل حتى عشرة ملفات DBF ، فى نفس الوقت وحتى سبعة ملفات مفهرسة NDX ، يمكنها تشكيلها من ملف واحد ، فى حين يعد حقلMEMO بمثابة ملفين.

ب ــ فى كل ملف DBF واحد يمكن تخزين ٢ مليون حرف ، كل سجل يمكن تشكيله من ١٢٨ حقل بما لا يزيد عن ٤٠٠ حرف فى السجل الواحد.

أهم حقول قاعدة البيانات +DB3:

أ ــ الحقل الحرفى بحد أقصى طوله ٢٥٦ حرف.

ب ــ الحقل الرقمى بحد أقصى ١٩ حرف.

obeikandl.com

ج ــ حقل تاريخ بحد أقصى ٨ حرف.

د ــ حقل منطقى بطول حرف واحد.

هـ ــ حقل الملاحظاتMEMO FIELD بحد أقصى ٥٠٠٠ حرف.

أنواع ملفات قاعدة البيانات:

* اسم الملــف: يختاره المستخدم بأقصى طول ٨ حروف،
* امتداد الملف: EXTENSION ويتكون من ثلاثة حروف مسبوقة بنقطة على نمط يشبه نظام تسمية ملفاتDOS .
* أهم أنواع الملفات.

ملفDBF :

ملف يظهر على هيئة جدول

ملف MEM :

مساحة محجوزة توضع به ثوابت تستخدمها أثناء تشغيل البرامج.

ملفNDX :

ملف يحتوى على نفس البيانات فى ملف DBF لكنه مرتب حســـب حاجــة المستخدم.

ملفPRG :

يشتمل على مجموعة أوامر قاعدة بيانات والتى تكون برنامجا معينـــا لأداء عملية معينة.

ملفTXT :

يستخدم فقط فى حالة طلب البيانات على هيئة ASCIIكود لاستخدامها هـــذه البيانات فى ملف آخر.

obeikandi.com

ملف SCR :

ملف يحتوى بيانات تستخدم فى اظهار الشاشات.

ملف VUE :

ملف يحوى أسماء الملفات NDX, DBF المتصلة بها وأسماء الحقول والخيارات المطلوبة عليها ، أى يمكنك رؤية حقول مختارة من ملفات مختارة.

أنواع الحقول فى DB3 + :

حرفى: أقصى طول له ٢٥٤ حرف ولا يمكن اجراء عمليات حسابية عليه.

تاريخ: طوله ٨ حرف ويأخذ شكل MM / DD / YY ويسجل وفق نظام التاريخ الامريكى ويمكن اجراء عمليات حسابية عليه.

منطقى: طوله حرف واحد T ، F أو N ، Y ويمكن تغيير الترتيب(نعم أو لا).

رقمى: يحوى الارقام التى تجرى عليها عمليات حسابية.

مفكرة MEMO : يستخدم لتسجيل كمية كبيرة من النصوص باسلوب بعيد عن قيود قاعدة البيانات وبمجرد اختيارك للكتابة فى حقل MEMO تفتح لك مدير قاعدة البيانات منسق الكلمات لتكتب به البيانات.

التعامل مع قاعدة البيانات (*) :

بافتراض أن برامج القاعدة مسجلة على القرص المغناطيسى الصلب تحت فهرس فرعى هو DBASE ولتحميلها فى الذاكرة يتم الآتى:

```
C : \> CD \ DBASE
C : \ DBASE >
```

(*) بدءا من هذا العنوان ومابلى من عرض سوف نشير بكلمة "القاعدة" إلى قاعدة البيانات +DB3.

obeikandl.com

يتحول المؤشر الى :

```
C : \DBASE > DBASE
```

والنتيجة :

أ ــ تظهر الشاشة التعريفية للقاعدة.

ب ــ على يسار اسفل الشاشة تظهر اشارة استعداد القاعدة وهى عبارة عن نقطة سوف نطلق عليها النقطة المضيئة وبعدها بمسافة واحدة تظهر الشرطة المضيئة.

ج ــ عند هذا الحد يمكن استخدام احد امرين:

(١) ASSIST وبذلك تنقل القاعدة الى MODE ASSIST أو الشاشة المساندة.

(٢) كتابه الاوامر مباشرة بجوار النقطة المضيئة وادخال الامر.

لتوضيح مفهوم الشاشات المساندة باستخدام الامر ASSIST تظهر شاشة المساندة الاولى شكل (١٣/١) ومن الشكل يتضح مايلى:

١ - يشمل أول سطر ثمان بدائل هى من اليسار لليمين:

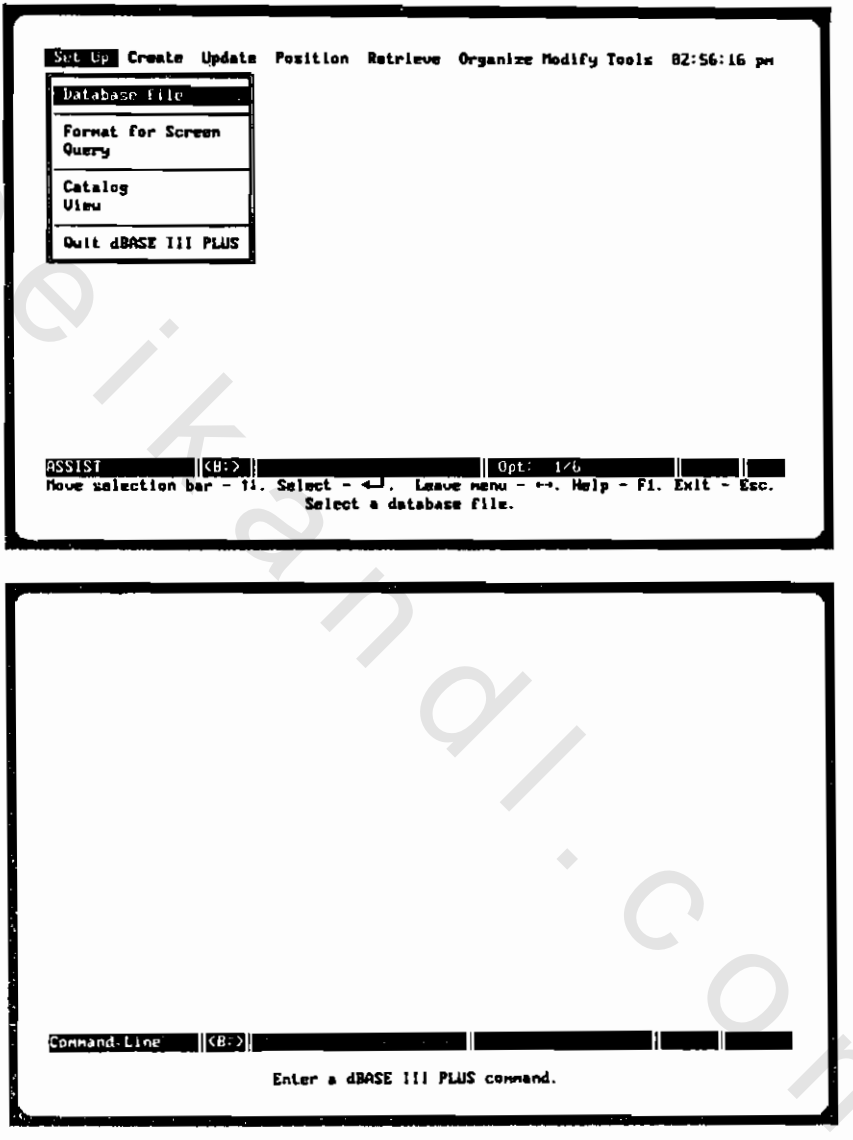
SETUP CREATE

٢ - اسفل اول اختيار على اليسار يظهر مستطيل مضىء به قائمة اختيارات فرعية لمجموعة SETUP وبالمثل يظهر جدول اسفل كل اختيار رئيسى اذا استخدمت المفتاح فى تحريك اضاءه الاختيارات.

٣ - سطر الاوامر على هيئه مستطيلات مضيئة ويظهر به ترجمة الاختيارات الى اوامر القاعدة.

٤ - سطر يوضح تعليمات الانتقال عبر الشاشة أو الخروج منها والعودة الى نظام التشغيل.

٥ - سطر ارشادى يساعد المستخدم على توصيف العمل الذى يقوم به.


Set Up Create Update Position Retrieve Organize Modify Tools 02:56:16 pm
Database file
Format for Screen
Query
Catalog
View
Quit dBASE III PLUS
ASSIST
<B:>
Opt: 1/6
Move selection bar - ↑↓. Select - ↵. Leave menu - ↔. Help - F1. Exit - Esc.
Select a database file.
Command Line
<B:>
Enter a dBASE III PLUS command.

الشكل (١٣/١)

الانتقال الى الحالة COMMAND MODE:

اضغط على مفتاح ESC تختفى شاشة المساندة الرئيسية وتظهر النقطة المضيئة على الجانب الايسر السفلى من الشاشة واسفلها يظهر سطرين سطر التوصيف وسطر اسفله لرسائل القاعدة للمستخدم.

تطبيق على استخدام القاعدة:

١ - انشاء ملف:

أ ــ يلزم قبل استخدام القاعدة انشاء هيكل الملف أى تحديد حقول السجل المكون للملف وتوصيف هذه الحقول وتحديد حيزها من البايت ، فاذا كان الهدف انشاء ملف اسمه MONEY يوضح المطالبات المالية من العملاء وهل سدد ام لم يسدد فان خطوات انشاء الملف ستكون على النحو:

(١) تحديد الحقول:

العميل (الاسم الاول ــ اسم العائلة ــ المدينة ــ رقم التليفون ــ القيمة المالية ــ التاريخ ــ حاله التسديد).

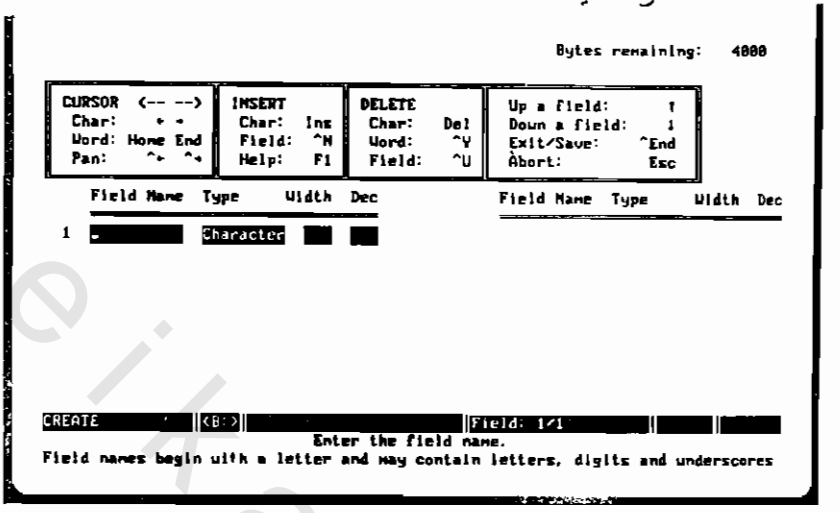
(٢) اكتب الامر انشأ ملف CREATE وادخله.

(٣) ترد القاعدة تطلب تحديد اسم الملف ويكتب اسم الملف.

```
CREATE
•
ENTER THE NAME OF THE NEW FILE: MONEY
```

(٤) بعد ادخال الامر يظهر على كما فى الشكل (١٣/٢) شاشة انشاء ملف والذى يظهر فى الجزء العلوى منها اسم الحقل ــ نوعه ــ طوله ــ الكسر العشرى.

(٥) السطر المضيء اسفل الشاشة داخل مجموعة المستطيلات هو سطر تحديد.



سكل (١٣/٢)

وتبدو المستطيلات من اليسار لليمين كما هو موضح بالجدول:

رقم المستطيل	المسمى	البيانات الموضحة
١	الأمر المستخدم.	إنشاء ملف CREATE
٢	رمز محرك القرص.	< B: >
٣	أسم الملف.	MONEY
٤	عدد الحقول.	عدد الحقول التى تم فتحها
٥	بيانات عن القاعدة.	
٦	،، ،، ،،	

أ ــ تظهر الشرطة المضيئة على سطر أول سجل عند أول حرف.

ب ــ يتم ادخال موصفات الحقول.

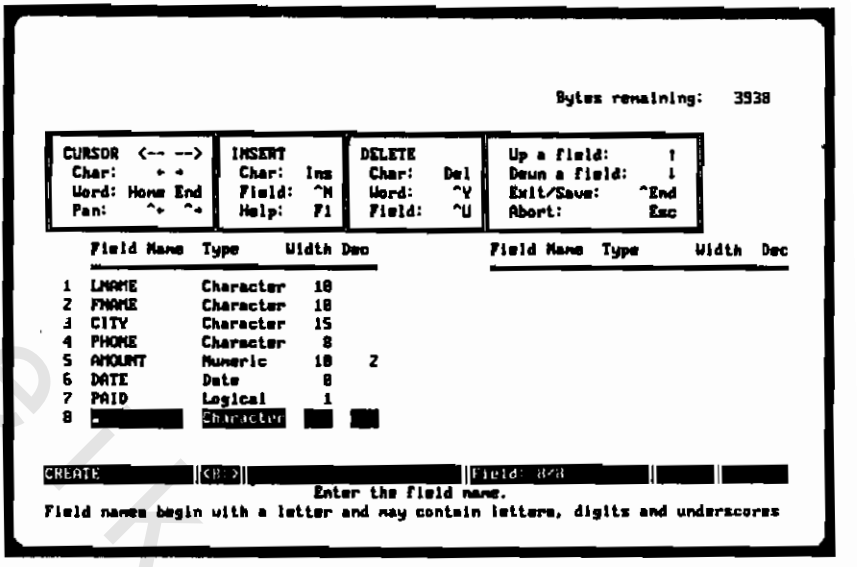
ج ــ يتم تخزين الهيكل بالضغط على مفتاحى CTRL , END فــى نفــس الوقت.

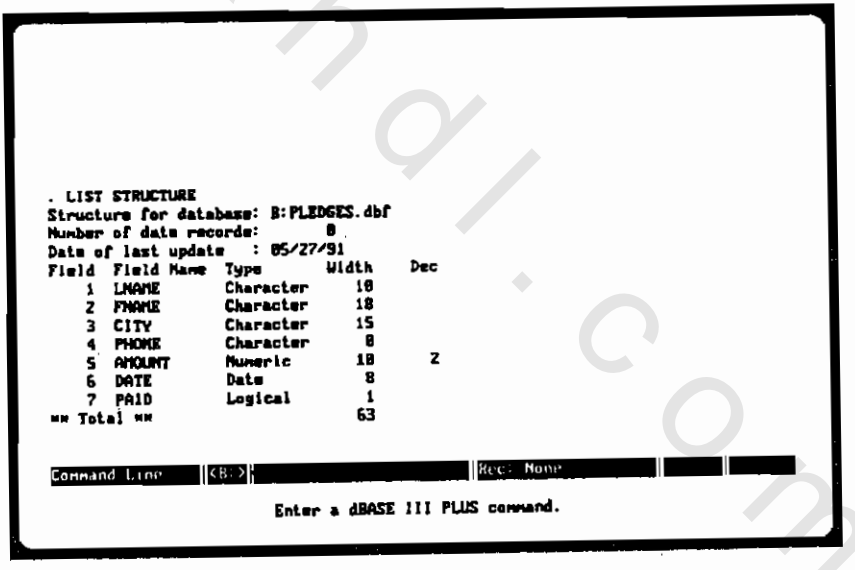
د ــ تسأل القاعدة هل سيتم ادخال البيانات الآن (N/Y) .

هــ ــ بالضغط على مفتاح الادخال مره اخرى تمسح الشاشة وتظهر النقطة المضيئة.

٢ - استعراض هيكل الملف:

اكتب الامر LIST STRUCTURE وادخاله سوف تظهر شاشة بها هيكل الملف ويمكن اجراء أى تعديل عليه ثم الضغط على مفتاحىCTRL, END ، ويوضــح الشاشة الشكل (١٣/٣).


Bytes remaining: 3938
CURSOR <-- -->
Char: ← →
Word: Home End
Pan: ^← ^→
INSERT
Char: Ins
Field: ^N
Help: F1
DELETE
Char: Del
Word: ^Y
Field: ^U
Up a field: ↑
Down a field: ↓
Exit/Save: ^End
Abort: Esc
Field Name Type Width Dec
1 LNAME Character 18
2 FNAME Character 18
3 CITY Character 15
4 PHONE Character 8
5 AMOUNT Numeric 18 2
6 DATE Date 8
7 PAID Logical 1
8
CREATE
Field: 8/8
Enter the field name.
Field names begin with a letter and may contain letters, digits and underscores


. LIST STRUCTURE
Structure for database: B:PLEDGES.dbf
Number of data records: 8
Date of last update : 05/27/91
Field Field Name Type Width Dec
1 LNAME Character 18
2 FNAME Character 18
3 CITY Character 15
4 PHONE Character 8
5 AMOUNT Numeric 18 2
6 DATE Date 8
7 PAID Logical 1
** Total ** 63
Command Line
Rec: None
Enter a dBASE III PLUS command.

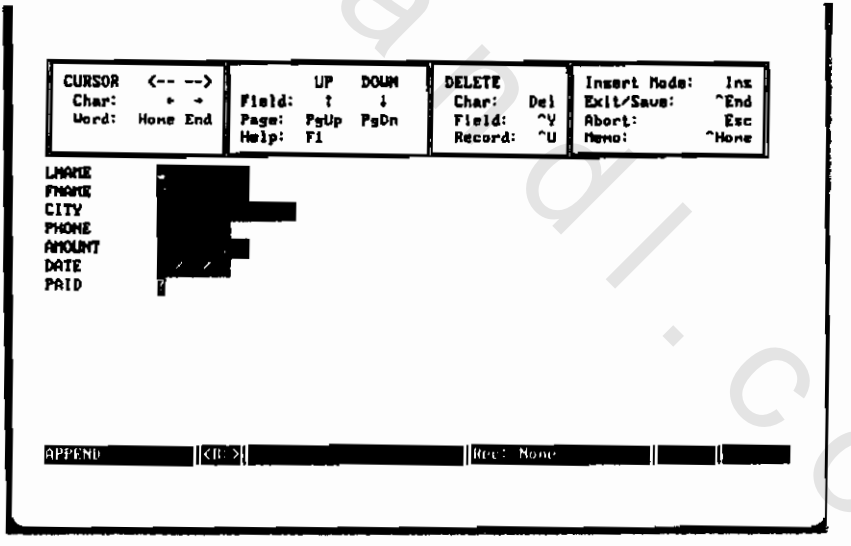
شكل (١٣/٣)

٣ - ادخال البيانات للملف:

يستخدم الامر APPEND :

- **APPEND**

وتظهر شاشة ادخال البيانات والشرطة المضيئة عند أول كلمة من أول حرف كما فى الشكل (١٣/٤) ويتم ادخال البيانات الحقول وفق ترتيبها وبعد ادخـــال كل السجلات ولضمان تسجيلها يتم الضغط على مفتاحىEND , CTRL وتمسح الشاشة وتظهر النقطة المضيئة.



الشكل (١٣/٤)

٤ - استعراض السجلات المخزنة:

يكتب الامر LIST

```
• LIST
```

تظهر جميع السجلات التى تم ادخالها دفعة ويلاحظ تحـــول بيانـــات الحقـــل المنطقى من Y / N إلى T/F , T حقيقى (نعم) ، F زائف (لا).

٥ - استعراض حقول:

يستخدم الامر LIST وبعدها اسماء الحقول المطلوبة :

```
• LIST NAME, AMOUNT, PHONE
```

ويتم ادخال الامر وتظهر الحقول المطلوبة فقط لجميع السجلات فيما يوضحه شكل (١٣/٥).

٦ - غلق الملف:

يستخدم الامر USE فتظهر النقطة المضيئة لان اعادة استخدام امر فتح الملف يعنى غلقه وفق مفهوم القاعدة.

٧ - الرجوع الى نظام التشغيل:

يستخدم الامر QUIT وينص عليه كاملا.

```
• QUIT
```

```
C :\>
```

٨ - تحديث السجلات:

أ ــ يفتح الملف بالامر [اسم الملف] USE .

- USE MONEY

ب ـ لاجراء التعديل يجب اخطار القاعدة بأن المطلوب تحديث مما يستدعى استخدام EDIT (اختصار كلمة EDITOR) فاذا اتبعت EDIT برقم السجل ظهر على الشاشة السجل المطلوب بعدها تحرك الشرطة

- EDIT 3

الضوئية (الكيرسير) الى الحقل المطلوب تعديله ويتم تعديله ثم التأكد من ادخال التعديل بالضغط على المفتاحين END ، CTRL وتظهر النقطة المضيئة مما يدل على اتمام تسجيل التعديل والاستعداد لاى تعديل آخر، ويمكن باستخدام المفتاح PgUp اعادة اظهار السطر (السجل) السابق والمفتاح PgDn استحضار السجل اللاحق.

٩ – استخدام الامر BROWSE :

وله نفس خصائص الامر EDIT الا انه يعرض جميع السجلات فى الملف الواحد (١٢ سجل لكل شاشة) وباستخدام مفتاحى PgDn , PgUp يمكن عرض صفحات سابقة أو لاحقة ، وفى حالة اجراء أى تعديل يجب التأكد من تسجيل التعديل بالضغط على مفتاحى CTRL, END كما فى شكل (١٣/٦).

١٠ – اضافه سجل جديد:

يستخدم الامر APPEND

- APPEND

وتظهر شاشة ادخال البيانات ويتم ادخال البيانات وتأكيد ادخالها وتسجيلها كما اسلفنا.

١١ - حذف السجلات أو سجل:

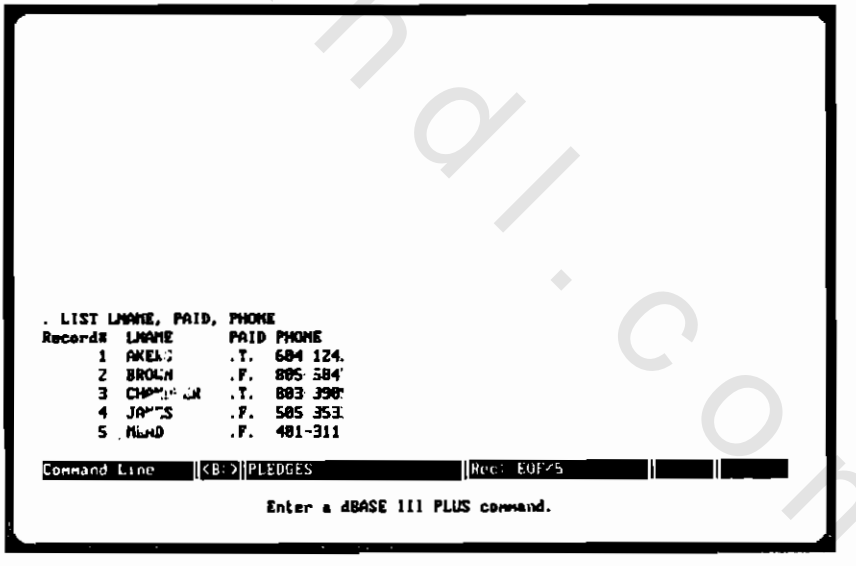
ويتم على مرحلتين:

أ ــ الإشارة الى السجل المطلوب حذفه بالامر DELETE وتوصيف حقل من السجل كالامر وبعد ادخال الامر لا يحذف فعليا من الملف.

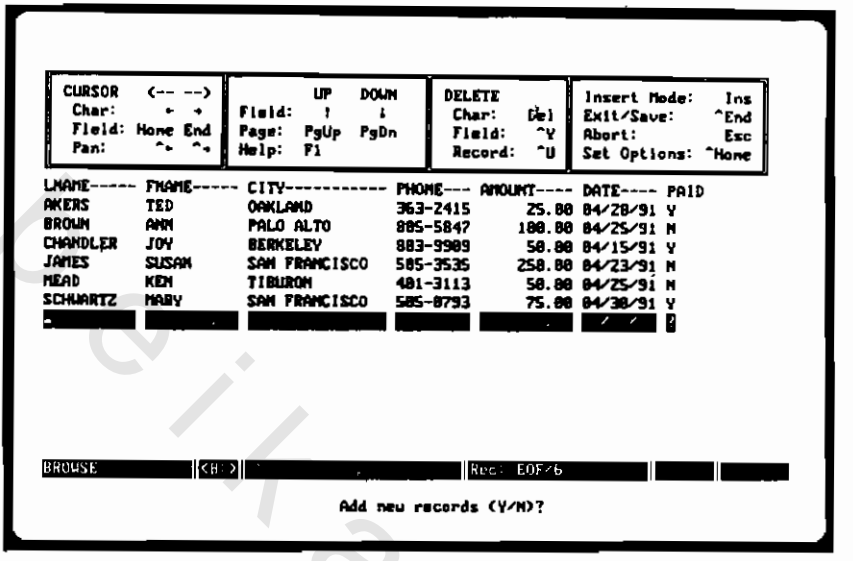
- DELETE FOR LNAME "ALI KERAA"

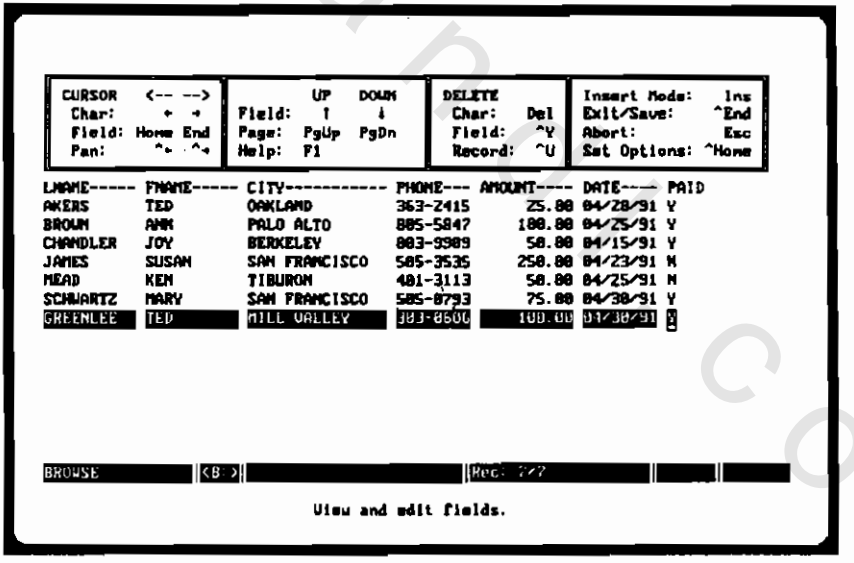
ب ــ تأكيد مسح السجل مسحا فعليا يلزم استخدام الامر PACK .

- PACK



شكل (١٣/٥)


CURSOR <-- -->
Char: ← →
Field: Home End
Pan: ^← ^→
UP DOWN
Field: ↑ ↓
Page: PgUp PgDn
Help: F1
DELETE
Char: Del
Field: ^Y
Record: ^U
Insert Mode: Ins
Exit/Save: ^End
Abort: Esc
Set Options: ^Home
LNAME----- FNAME----- CITY---------- PHONE--- AMOUNT---- DATE---- PAID
AKERS TED OAKLAND 363-2415 25.00 04/28/91 Y
BROWN ANN PALO ALTO 885-5847 100.00 04/25/91 N
CHANDLER JOY BERKELEY 883-9909 50.00 04/15/91 Y
JAMES SUSAN SAN FRANCISCO 585-3535 250.00 04/23/91 N
MEAD KEN TIBURON 481-3113 50.00 04/25/91 N
SCHWARTZ MARV SAN FRANCISCO 585-8793 75.00 04/30/91 Y
BROWSE <B:> Rec: EOF/6
Add new records (Y/N)?


CURSOR <-- -->
Char: ← →
Field: Home End
Pan: ^← ^→
UP DOWN
Field: ↑ ↓
Page: PgUp PgDn
Help: F1
DELETE
Char: Del
Field: ^Y
Record: ^U
Insert Mode: Ins
Exit/Save: ^End
Abort: Esc
Set Options: ^Home
LNAME----- FNAME----- CITY---------- PHONE--- AMOUNT---- DATE---- PAID
AKERS TED OAKLAND 363-2415 25.00 04/28/91 Y
BROWN ANN PALO ALTO 885-5847 100.00 04/25/91 Y
CHANDLER JOY BERKELEY 883-9909 50.00 04/15/91 Y
JAMES SUSAN SAN FRANCISCO 585-3535 250.00 04/23/91 N
MEAD KEN TIBURON 481-3113 50.00 04/25/91 N
SCHWARTZ MARV SAN FRANCISCO 585-8793 75.00 04/30/91 Y
GREENLEE TED MILL VALLEY 383-8600 100.00 01/30/91 N
BROWSE <B:> Rec: 7/7
View and edit fields.

شكل (١٣/٦)

وما أن ينفذ الامر يتم حذف السجل من على الاقراص ولا سبيل لاســـتعادته مره اخرى. لكن ان استشعر مستخدم القاعدة وجود خطأ وقبل الامرPACK ويريد الاحتفاظ بالسجل المحذوف بامر الاستدعاء RECALL على النحو:

- RECALL FOR LNAKE "ALI KERAA"

ملاحظات هامة:

١ - ما سبق عرضه مجرد فكرة محدودة لاستعراض امكانية فتح ملف والتعامل معه وهذا يكفى قطاع كبير من مستخدمى قواعد البيانات.

٢ - يتطلب التوصل الى استغلال القاعدة الاستغلال الامثل الاستعانة بكتب خاصة بها وهناك العديد من الكتب العربية فى شأن قاعدة البيانات + DB3 .

٣ - ضمن برامج القاعدة يوجد شرح واف لها يستدعى بالامرPLEH .

- HELP

وتظهر أول شاشة للشرح.

٤ - استخدام الامر SET يتيح امكانيات كثيرة فى القاعدة لعل ابسطها تغيير الــوان الحروف على الشاشة ولون الخلفية.

*

obeikandl.com

١٤ ـ تحليل وتصميم نظم المعلومات

كان لجوء المدنية الحديثة إلى بناء نظم معلومات مرتكزة على الحاسبات رد فعل ناجح حيال التضخم الهائل فى حجم البيانات المتاحة التى مما جعل معالجتها بالاساليب اليدوية دربا من الخيال للوصول الى نتائج ومعلومات تتصف بالسرعة والدقة فى الوقت المناسب والشكل المناسب ، إذ دون تلك المواصفات لا تستطيع الإدارة إتخاذ القرار السليم.

وبداية يجب أن نعلم أن اضخم الحاسبات فى العالم لا يمكنها مهما اتصفت بالسرعة أو الدقة أو المعولية أو الانتاجية أن تغير من طبيعة البيانات ومصداقيتها لان البيانات إن لم تكن صحيحة وصادقة فأن المعلومات الناتجة بعد معالجتها على الحاسب لن تكون صحيحة أو صادقة فما بنى على خطأ فهو خطأ ايضا لدرجة أن الأمريكيين يستخدمون تعبير GIGO أى النفايات تؤدى إلى نفايات.

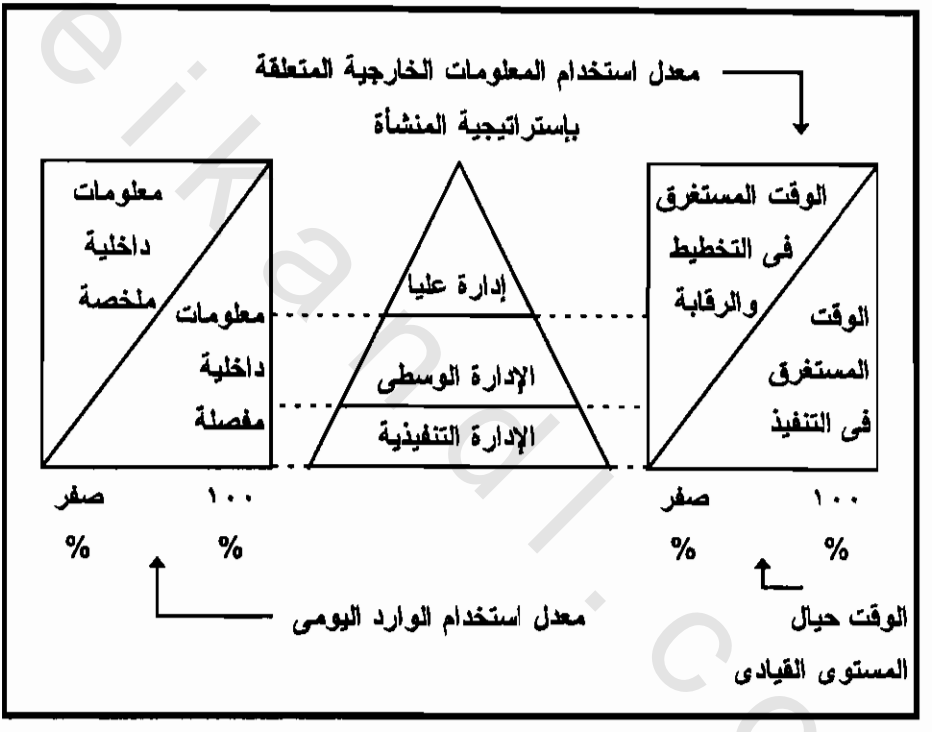
GARBAGE IN ..... GARBAGE OUT -

ان البداية الحقيقة وحجر الزاوية فى معالجة البيانات وإنشاء نظم المعلومات هى البيانات وفيما عدا ذلك فهى مجرد ادوات ... قد تكون ادوات الكترونية أو غير ذلك ، فالاهمية القصوى للبيانات الدقيقة والمؤكدة التى بعد معالجتها تقدم للإدارة أمرين فى غاية الأهمية هما:

الأول : تقارير وفق المستوى الادارى سيان كانت تقارير دورية أو سنوية أو عند الطلب .

الثانى : مساندة دعم القرار من خلال نظام فرعى ضمن نظام المعلومات الشـــامل يساند الادارة العليا فى اتخاذ قراراتها على المستوى الاستراتيجى والتنفيذى.

والبيانات التى تعالجها نظم المعلومات يمكن تصنيفها الى نوعين من البيانات ، بيانات داخلية وبيانات خارجية وكلاهما حيوى فى النظام كما هو مبيـــن بـــالشكل (١٤/١) والذى يوضح ايضا الوقت المستغرق فى الرقابة والتخطيط ونظيره فى التنفيذ والذى يعكس مدى حجم استخدام المعلومات الخارجية المتعلقة باستراتيجية المؤسسة.



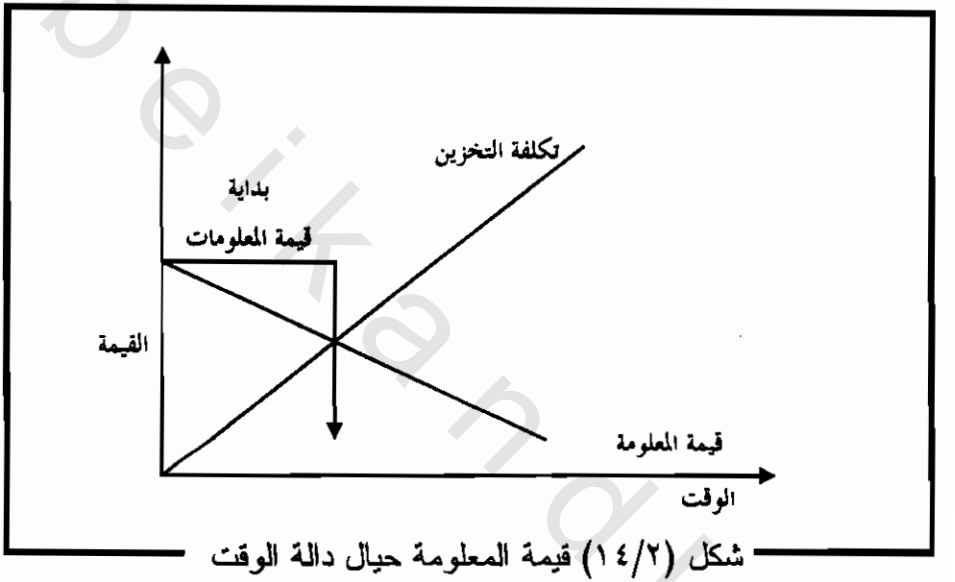
الشكل (١٤/١)

مميزات نظم المعلومات المرتكنة على الحاسب:

بفرض أن دقه البيانات امر مؤكد لا نقاش حوله فما هى الاهداف المحققة من بناء النظام على الحاسب؟ يجيب على هذا التساؤل عده حقائق نتناولها على النحو التالى:

السرعة:

لقد وجد بالدليل القاطع أن قيمة المعلومة تصل الى اقصى قيمة لها أن وصلت الى مستخدمها فى الوقت المناسب ، فان قسنا القيمة بالسعر فاننا نلحظ أن قيمة المعلومة تتدنى مع مرور الوقت حتى تصبح بعد فترة طالت أو قصرت عديمة الفائدة فيما يوضحه الشكل (١٤/٢).



شكل (١٤/٢) قيمة المعلومة حيال دالة الوقت

وبالتالى فان معالجة البيانات يدويا يستغرق وقت طويل نسبيا مما لا يترك مدى زمنى كاف لوصول المعلومة الى مستخدمها فى الوقت المناسب ، عكس ذلك تتفوق الحاسبات فى هذا المضمار اذ تعمل وحداتها الالكترونية على انجاز عده ملايين أو بلايين من العمليات فى الثانية الواحدة فيما يعجز عنه الانسان تماما.

الدقة:

يستطيع الحاسب معالجة كم ضخم من البيانات طبقا لمجموعة من الاجراءات الصعبة والعمليات المعقدة والمتكررة والتى يستحيل على الانسان اجراؤها حيث يتعرض الفرد للاجهاد والاعياء.

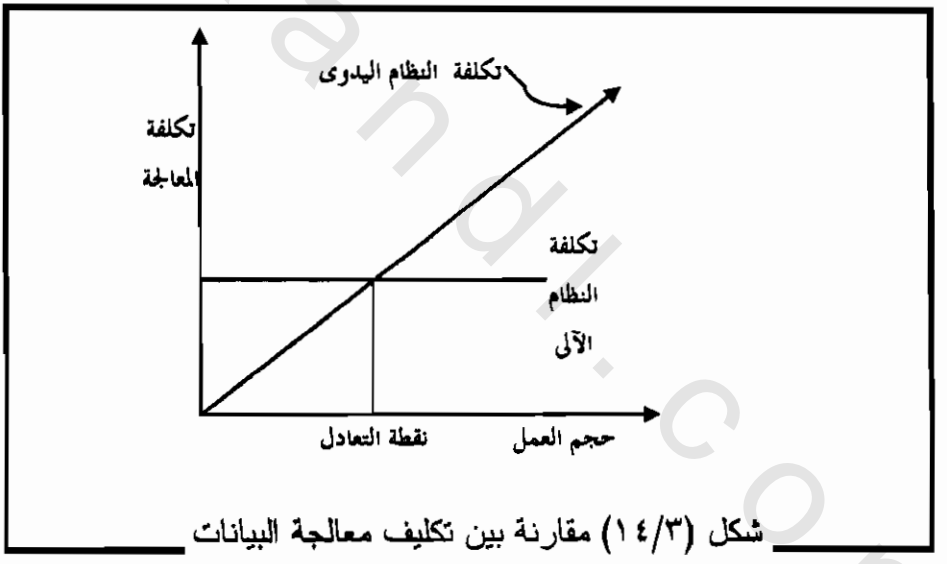
تكاملية المعلومات:

اتاحت قدرات قواعد البيانات الضخمة اجراء تكاملية بين مختلف انواع الملفات للمنظمة الواحدة ولإنها تدار جميعا بواسطة مدير قواعد البيانات الموحد DBMS فإن الحاسب يمكنه تقديم معلومات متكاملة عن الموضوع الواحد من مختلف الزوايا واجراء الترابط المنطقى بينها وتقديم تقارير تتصف بالشمولية والاحاطة الى جانب اعدادها فيزيائيا بالشكل المناسب.

الاقتصاد:

ويمكن أن نتناول هذا الجانب من خلال منظورين:

الاول : تكلفة المعالجة ويوضح الشكل التالى أن التشغيل الالكترونى للبيانات يقل كثيرا عن نظيره اليدوى كلما زاد حجم البيانات فيما يوضحه الشكل (١٤/٣).

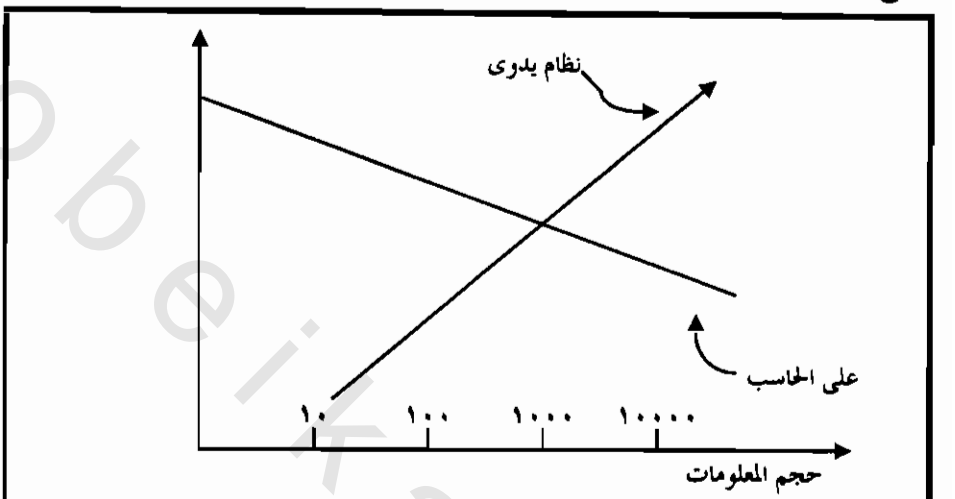


شكل (١٤/٣) مقارنة بين تكليف معالجة البيانات

الثانى: من حيث الاحتفاظ بالبيانات.

على ضوء المتاح حاليا من التكنولوجيا فى مجال تخزين البيانات والمعلومات فأن تكلفة تخزينها على وسائط التخزين الخاصة بالحاسبات الالكترونية

تقل كثيرا عن تكلفة تخزينها فى النظم اليدوية فيما يوضحه (١٤/٤) اضافة الى تلافى مشاكل فقد أو تلف أو تدمير أو تزوير البيانات.



شكل (١٤/٤)

تكلفة تخزين البايت

وقبل أن تتطرق الى تحليل وتصميم نظم المعلومات فانه من الافضل والاوفق عرض مبسط لمفهوم النظم لان الحياة ذاتها عبارة عن نظام ضخم يضم عددا هائلا من النظم الاقل حجما وهذه بدورها تشمل نظما فرعية وهكذا حتى نصل الى الميكروب الذى يشكل بدوره نظاما متكاملاً.

النظام :

" النظام عبارة عن مجموعة من الاجراءات المترابطة والموارد المتكاملة التى تحقق هدف النظام " ، فوزارة التعليم العالى تشكل نظاما هدفه التعليم والتثقيف والبحث العلمى وهو نظام ضخم يضم عددا من النظم الفرعية كالكليات والمعاهد ومراكز البحوث . وتنقسم النظم الى نوعين اساسيين :

أ - **نظم مفتوحة :**

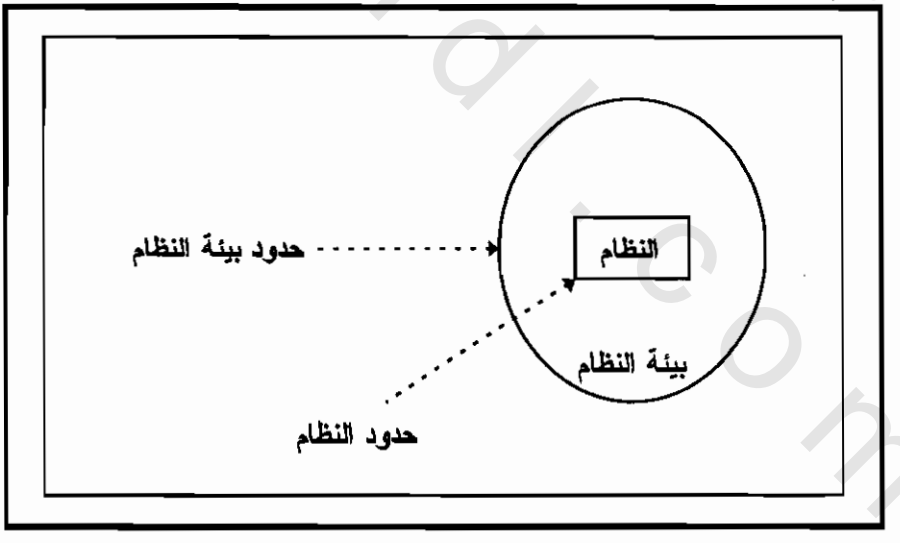
وهى النظم المتفاعلة مع البيئة المحيطة بها تؤثر فيها وتتأثر بها ، ومن امثلـــه النظم المفتوحة الانسان ، الكلية ، المؤسسة ، المكتبة ، نظم المعلومات.

ب - **نظم مغلقة :**

وهى النظم التى لا تتعامل مع البيئة المحيطة بها ، ولا تؤثر فيها أو تتأثر بـــها مثل الكرسى نظام مغلق ، الضفدعة فى فترة البيات الشتوى نظام مغلق.

إطار وحدود النظام :

وهو الشكل أو المحيط الذى يحدد هذا النظام ، فالحدود الخارجية للانســـان هى حدود نظامه والسور الخارجى لمصنع هو حدود نظامه ، وبالتالى فان لكل نظام حدود ويستحيل وجود نظام دون حدود حيث يستحيل توصيفة. وحدود النظام تفصـــل وتحدد النظام عن النظام اكبر محيط به يسمى بيئة النظام كما هـــو مبيـــن بـــالشكل (١٤/٥).



الشكل (١٤/٥) حدود النظام وبيئة النظام

وتنقسم بيئة النظام إلى ثلاث بيئات تؤثر فى بعضها البعض :

أ - **البيئة الداخلية :**

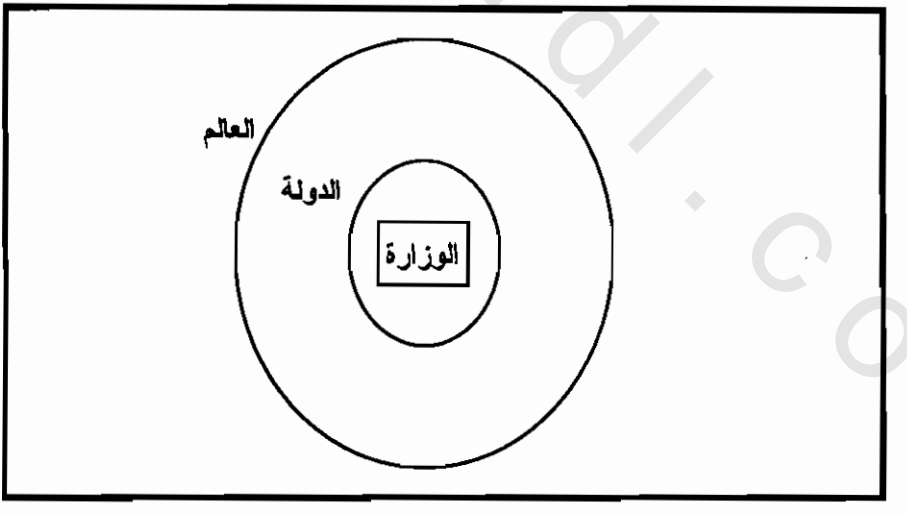
وهى البيئة اللصيقة بالنظام ، فأسرة الفرد ، ومقر سكنه ، ومقر عمله هى البيئة الداخلية له ، والمترددين على المكتبة والعاملين بها والجهة التابعة لها هى بيئتها الداخلية ، والعمال والموردون والمستوردون هم البيئة الداخلية لمصنع أو شركة .

ب - **البيئة الخارجية :**

وهى البيئة الاكبر التى تحيط بالبيئة الداخلية فالوزارة نظام بيئته الخارجية هى الدولة .

ج - **البيئة البعيدة :**

وهى النظام الشامل الذى يحيط بالبيئة الخارجية ، فالدولة جزء من بيئة اوسع هى العالم ... فيما يوضحه شكل (١٤/٦) مفهوم هذه البيئات.



شكل (١٤/٦) بيئات النظام

openkandil.com

هدف النظام :

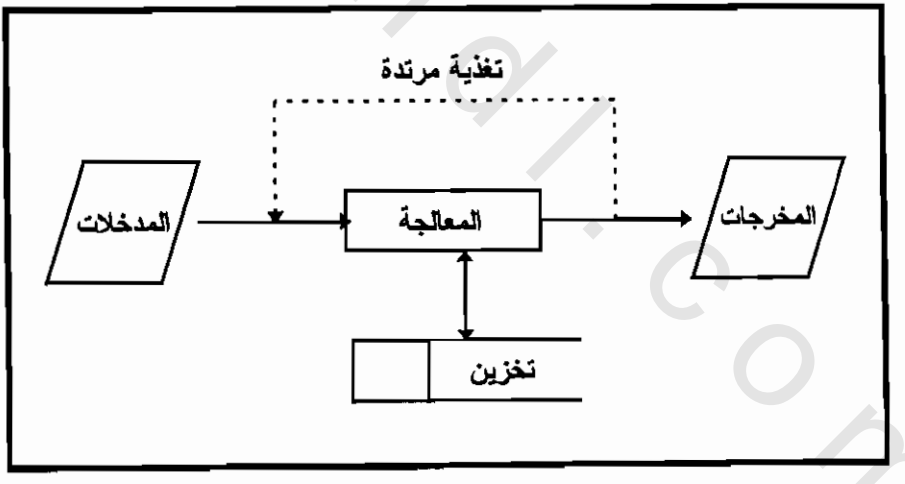
لكل نظام هدف يسعى الى تحقيقه ، فالانسان نظام هدفه عبادة الله واعمار الارض بالعمل لا بالقول الاجوف ، وللشركة هدف هو تعظيم ارباحها ، ولنظام المعلومات هدف يتمثل فى اعداد البيانات على هيئة معلومات لمساندة متخذ القرار ، وأى نظام بلا هدف ينفى سبب وجوده ولا يعتد به.

عناصر النظام :

لان النظام المفتوح يتصف بالتفاعلية مع البيئة ويتلقى منها ويقدم إليها لذا فأى نظام لابد أن يضم العناصر التالية الموضحة بالشكل (١٤/٧):

أ - **المدخلات والمخرجات :**

طالما كان النظام مفتوحا فان تفاعله مع البيئة يتمثل فى الاخذ والعطاء أو المدخلات والمخرجات.



شكل (١٤/٧)

عناصر النظام

ب - **المعالجة :**

تعتبر المعالجة قلب النظام النابض الذى يتولى تحويل المدخلات إلى مخرجات.

ج - **التغذية المرتدة :**

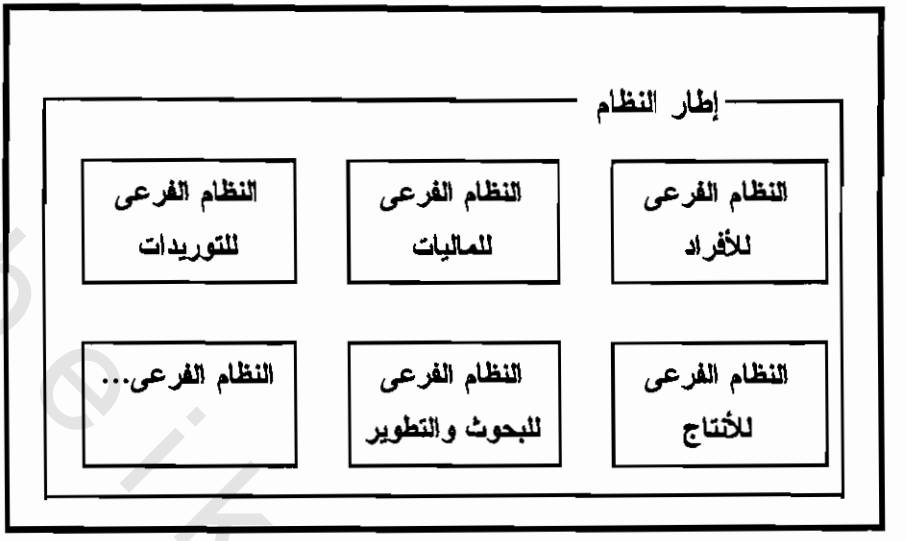
وهى اشارات الضبط والتحكم التى يصدرها النظام لضبط المدخلات لتوافق المخرجات ، ففى نظام مثل تقطير البترول قد تنخفض درجة الحرارة فجأة فيصدر النظام اشارات لتقليل تدفق المدخلات من البترول الخام ، والانسان يلقى فى جوفه باقراص الاسبرين متى احس بآلام الصداع .

د - **التخزين:**

وتمثل قدرة النظام على الاحتفاظ بما يفيض عن حاجته من مدخلات تحسبا لاستخدمها فى وقت لاحق . فالانسان قد يدخر قدرا من المال يفيض عن حاجته ، ونظام المعلومات قد يحتفظ ببعض المدخلات لحين الحاجة اليها رغم عدم استخدامها فى المخرجات المباشرة للنظام .

النظم المعقدة :

فى الغالب والأعم والأشمل تعتبر كل النظم المعاصرة نظما معقدة ، ولا نقصد بالتعقيد هنا الصعوبة بمضمونها الشائع ، إنما نقصد أنها نظم كبيرة تتكون من مجموعة من النظم الفرعية المترابطة بعلاقات تبادلية ولكل نظام فرعى مدخلات ومعالجات ومخرجات وغالبا تكون مدخلات احد النظم الفرعية هى مخرجات نظام فرعى آخر ، وهذه المنظومة المتشابكة هى التى تشكل النظام المعقد أو النظام الاكبر ، فيما يوضحه شكل (١٤/٨) عن نظام احدى المؤسسات أو الشركات ومكوناته من النظم الفرعية.



شكل (١٤/٨)

منظومة النظم الفرعية فى النظم المعقدة

نظم المعلومات :

توصف نظم المعلومات بأنها تجمع من مختلف الموارد (بشر – معدات – اساليب – اجراءات – سياسات ...) لتأدية الوظائف التالية :

١ – جمع البيانات وتدققها .

٢ – تسجيل البيانات .

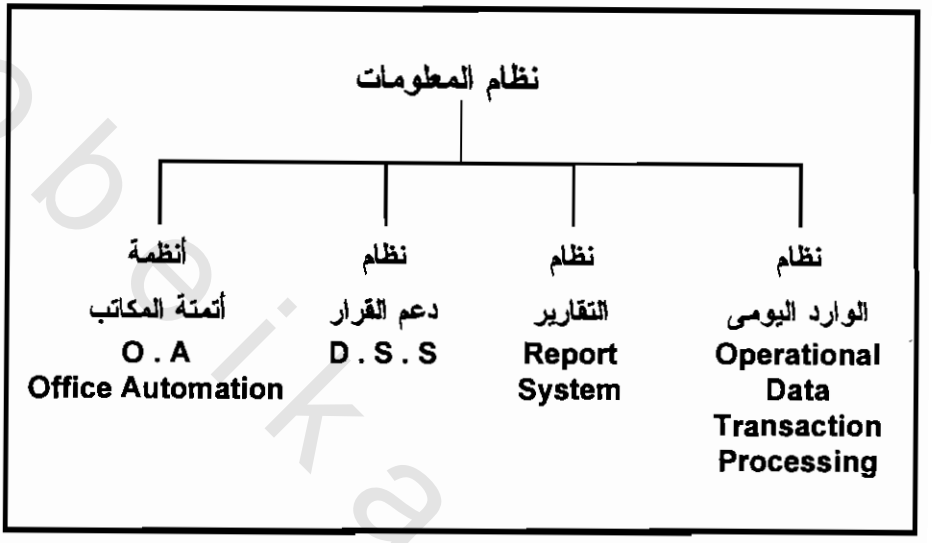
٣ – معالجة البيانات وتحويلها إلى معلومات.

٤ – استرجاع البيانات واتاحتها للمستخدمين .

٥ – تأمين البيانات والمعلومات .

ويمكن النظر لنظام المعلومات على انه القطاع الداخلى من المؤسسة أو المنظمة المسئول مسئولية مباشرة عن اجراء التوافق والربط على المستوى

الاستراتيجى بين المؤسسة والمتعاملين معها عبر مجموعة من النظم الفرعية الموضحة بالشكل (١٤/٩):



شكل (١٤/٩)

عناصر نظام المعلومات

أ - نظام الوارد اليومى :

وهو نظام فرعى يتولى جميع الاجراءات - السالف الاشارة اليها فى الفقره السابقة - حيال البيانات المتحركة من والى المؤسسة أو الشركة يوميا.

ب - نظام التقارير :

وهو نظام فرعى آخر ضمن اطار النظام الشامل يستخلص من معالجة بيانات الوارد اليومى المادة اللازمة لاعداد مستويات مختلفة من التقارير تناسب المستويات الثلاث فى المؤسسة سيان كانت التقارير:

(١) دورية .
(٢) أو عند الطلب .
(٣) أو مفاجئة .

ج – نظام دعم القرار :

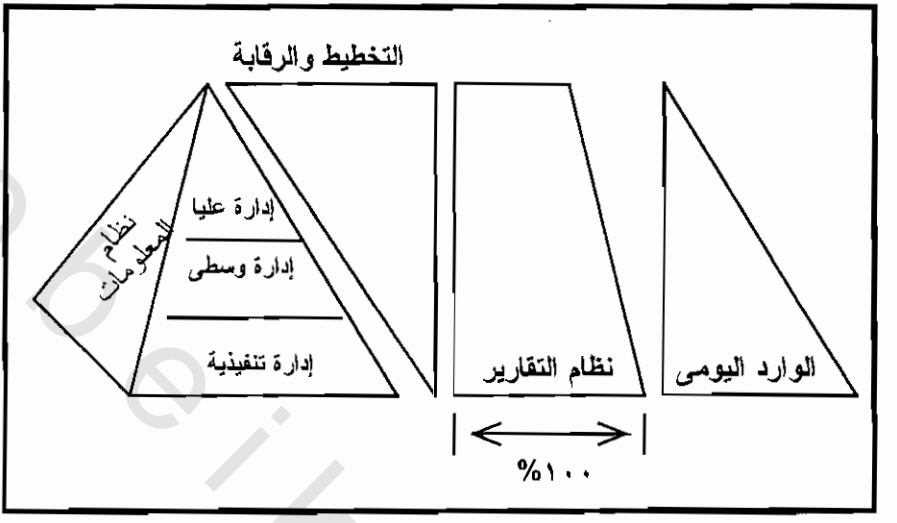
وهى نظم تتيح التعرف على بدائل مختلفة لحل احدى المشاكل التى يواجهها العمل بناء على المعلومات المؤكدة المتوافرة فى نظام المعلومات ، واود الاشارة انها نظم لدعم القرار وليست لاتخاذ القرار كما قد يشاع عنها فاتخاذ القرار مسئولية القيادات البشرية وليس مسئولية الحاسب أو نظام المعلومات مهما تعاظمت قدراته.

د – النظم الخبيرة :

تمثل هذه النظم ذروة التقدم والنجاح فى نظم المعلومات بفضل جمعها بين التراكمات المعلوماتية المتاحة فى نظام المعلومات وبين خبرة الخبراء كل فى مجاله وتخصصه ، وقد يرى البعض أن النظم الخبيرة تندرج تحت بنود الذكاء الصناعى وليست مكونا فى نظم المعلومات ، وأعتقد أنه رغم إرتكانها على أساليب الذكاء الصناعى إلا أن محل أهتمامها هو إتخاذ القرار وبالتالى أعتقد أن موقعها هنا صائباً.

* * *

ويوضح الشكل (١٠/١٤) علاقة النظم الفرعية لنظام المعلومات بالمستويات الادارية الثلاث :



شكل (١٠/١٤)

المستويات الإدارية وإنتاج نظام معلومات

المكونات الآلية لنظم المعلومات المرتكنة على الحاسب :

{ CBIS} = COMPUTER BASED INFORMATION SYSTEM

اوضحنا فى بداية الباب الأول الفوائد والمزايا المحققة من ارتكان نظم المعلومات على الحاسبات ، واهمية البيانت شريطة أن تكون :

أ – منطقية وواقعية ومتناسقة .

ب – لها صلة بالموضوع .

ج – دقيقة .

د – بالقدر المناسب والحجم المناسب .

هـ – وفى التوقيت المناسب .

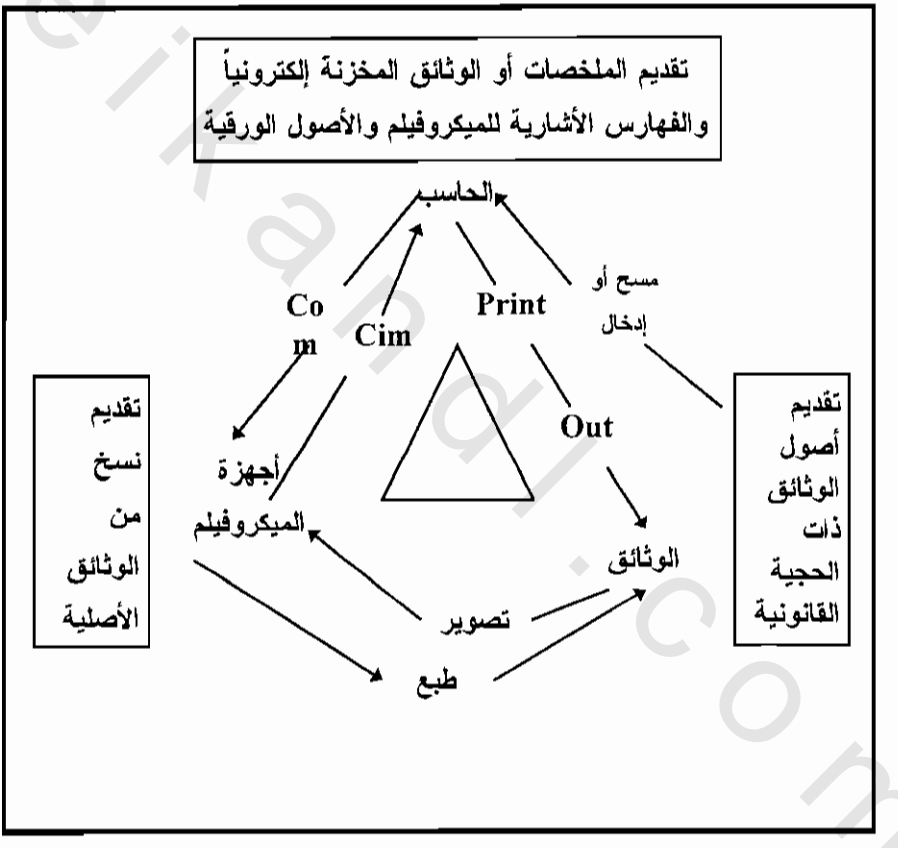
والبيانات الواردة الى النظام قد تحمل على وثائق لها حجية قانونية مما يستدعى حفظها حفظا دائما مع اتاحة استخدام بياناتها أو صور منها مما يستدعى تصميم الهيكل الآلى الداعم لنظم المعلومات من المكونات الآلية التالية:

١ – الحاسبات .

٢ – معدات تسجيل الوثائق (الميكروفيلم – اجهزة المسح الضوئى –CD - ROM).

٣ – معدات حفظ الوثائق الورقية .

ويوضح الشكل (١٤/١١) الاسلوب التنفيذى لحركة البيانات بين هذه الوسائل.



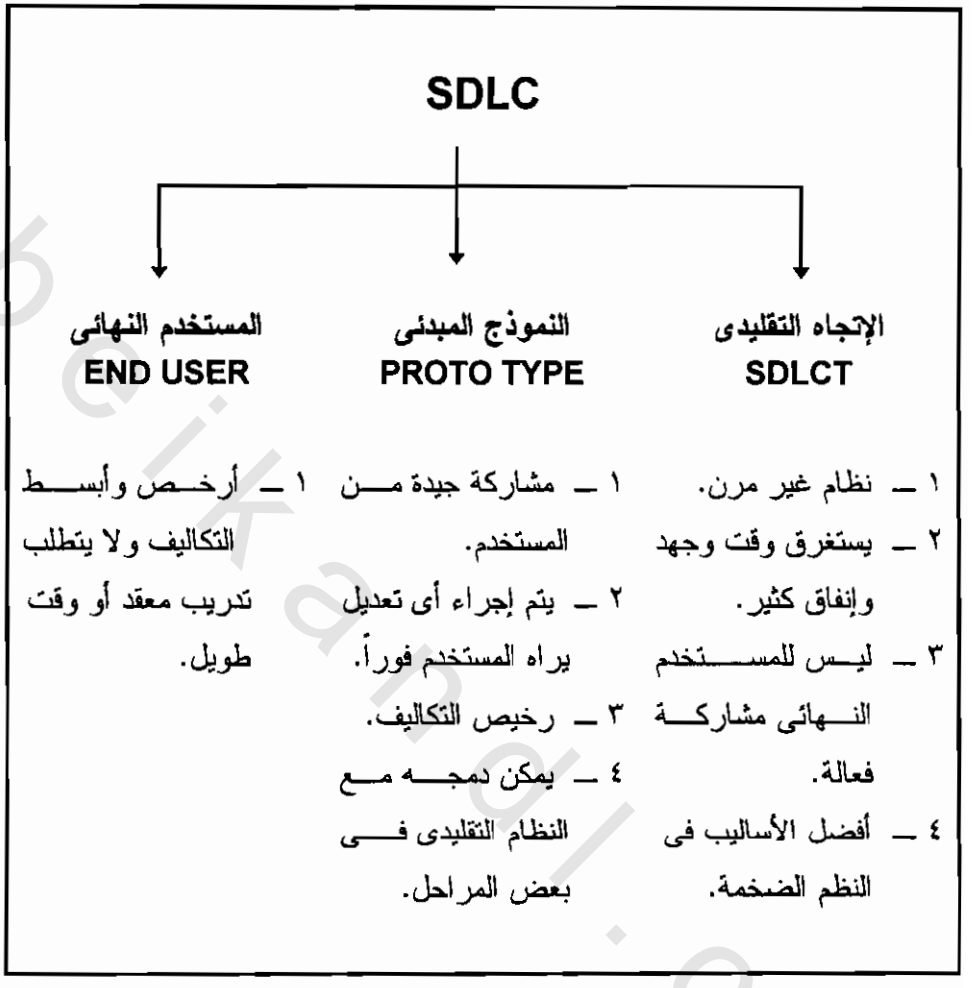
الشكل(١٤/١١)

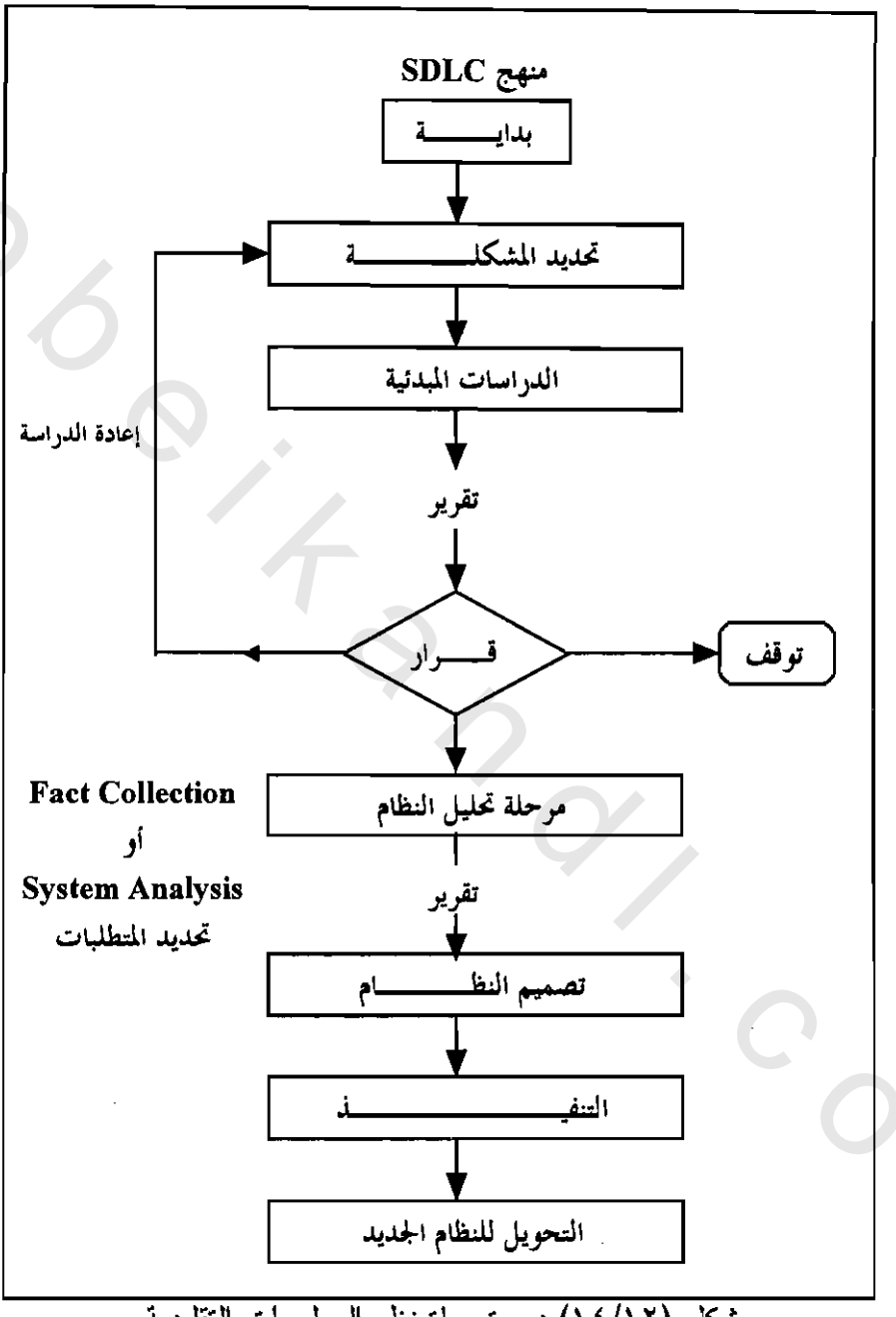
العلاقة بين عناصر الدعم الآلى فى نظم ***CBIS***

دورة حياة نظم المعلومات:

نظم المعلومات شأنها شأن مختلف النظم ويسرى عليها ما يسرى على حياة الانسان ، تبدأ صغيرة ثم يشتد عودها ثم يصيبها الوهن وتصبح عبئا على مستخدميها مما يستدعى تطويرها باستمرار ، أو إعادة انشائها. وتسمى المراحل التى يمر بـــها نظام المعلومات دورة حياة للنظـــام SYSTEM DEVELOBMENT LIFE CYCLE ويطلق عليها اختصاراً {SDLC} . وهى دورة أو خطوات ليست جامدة أو ملزمة بل يمكن تطويرها وتطويعها وفق ظروف النظام المقترح انشاؤه ، أو تطويره ، ومـــدى حجم وعدد المتعاملين معه ، ففى نظم المعلومات الضخمة من الأفضل اتبـــاع دورة طويلة ذات خطوات متعددة تسمى الدورة التقليدية لحياة النظام ويرمز لها SDLC الا أن هذا البطىء النسبى والعمل الممتد الشاق والتكلفة المرتفعة لانشاء النظام تعطـــى قدرا جيدا من الضمان لبناء نظام معلومات جيد ومحقق لاهدافه . ويمكـــن اســتخدام دورة حياة نظام لا تشمل جميع المراحل ، تعتمد على اســـلوب النمـــوذج المبدئـــى PROTO TYPE الذى يشارك فيه المستخدم النهائى مشاركة فعالة أو تطبيق المبـــدأ القائل ان اردت ان تعلم الانسان العوم القه فى الماء ويسمى هذا الأســـلوب أســلوب المستخدم النهائى END USER ، وفى هذا الاتجاه يتم تدريب المستخدم على إحـــدى الحاسبات الشخصية وإعطاءه الفرصة لإنشاء ما يود من نظم معلومات .

ويوضح الشكل (١٤/١٢) مفهوم وخطوات دورة حيـــاة النظـــام التقليديـــة ويوجز الإطار التالى الفروق الجوهرية بأختصار شديد بين الاتجاهات الثلاث .


SDLC
الإتجاه التقليدى
SDLCT
١ ــ نظام غير مرن.
٢ ــ يستغرق وقت وجهد وإنفاق كثير.
٣ ــ ليس للمستخدم النهائى مشاركة فعالة.
٤ ــ أفضل الأساليب فى النظم الضخمة.
النموذج المبدئى
PROTO TYPE
١ ــ مشاركة جيدة من المستخدم.
٢ ــ يتم إجراء أى تعديل يراه المستخدم فوراً.
٣ ــ رخيص التكاليف.
٤ ــ يمكن دمجه مع النظام التقليدى فى بعض المراحل.
المستخدم النهائى
END USER
١ ــ أرخص وأبسط التكاليف ولا يتطلب تدريب معقد أو وقت طويل.



شكل (١٢/١٤) دورة حياة نظم المعلومات التقليدية

عندما تظهر مشكلة فى النظام القائم أو الرغبة فى إنشاء نظـــام معلومـــات جديد CBIS نتيجة وجود خلل فى النظام القائم يتم أخطـــار أدارة النظـــم بالمؤسســـة بمظاهر المشكلة وفق التقرير بالشكل (١٤/١٣) وبدروها تـــدرس المشكلـــة وتحـــدد مجموعة عمل مصغرة لأجراء دراسات مبدئية عن المشكلة على أن تقـــدم تصـــورا للحل كما تقدم تصورا للتكلفة الإجمالية التقريبية من خلال دراسات جـــدوى مبدئيـــة وعلى ضوء هذه المعلومات تتخذ إدارة النظم قرارها.

وتبدأ على الفور أول مرحلة فى إنشاء نظام المعلومات وهى مرحلة تحليـــل نظام المعلومات.

أولا : فكرة موجزة عن تحليل نظم المعلومات :

تسمى هذه المرحلة مرحلة جمع حقائق النظام الحالى ، ويتولاها محللو النظم ولانها جمع حقائق فأن محلل النظام يلتزم التزاما تاما بما يجمعه من بيانات وليس له حق إضافة أو تعديل أو إعطاء اقتراحات أنما يكون رجلا متعادلا .

وتتم هذه المرحلة فى الخطوات التالية :

١ - **مرحلة دراسات الجدوى المبدئية :**

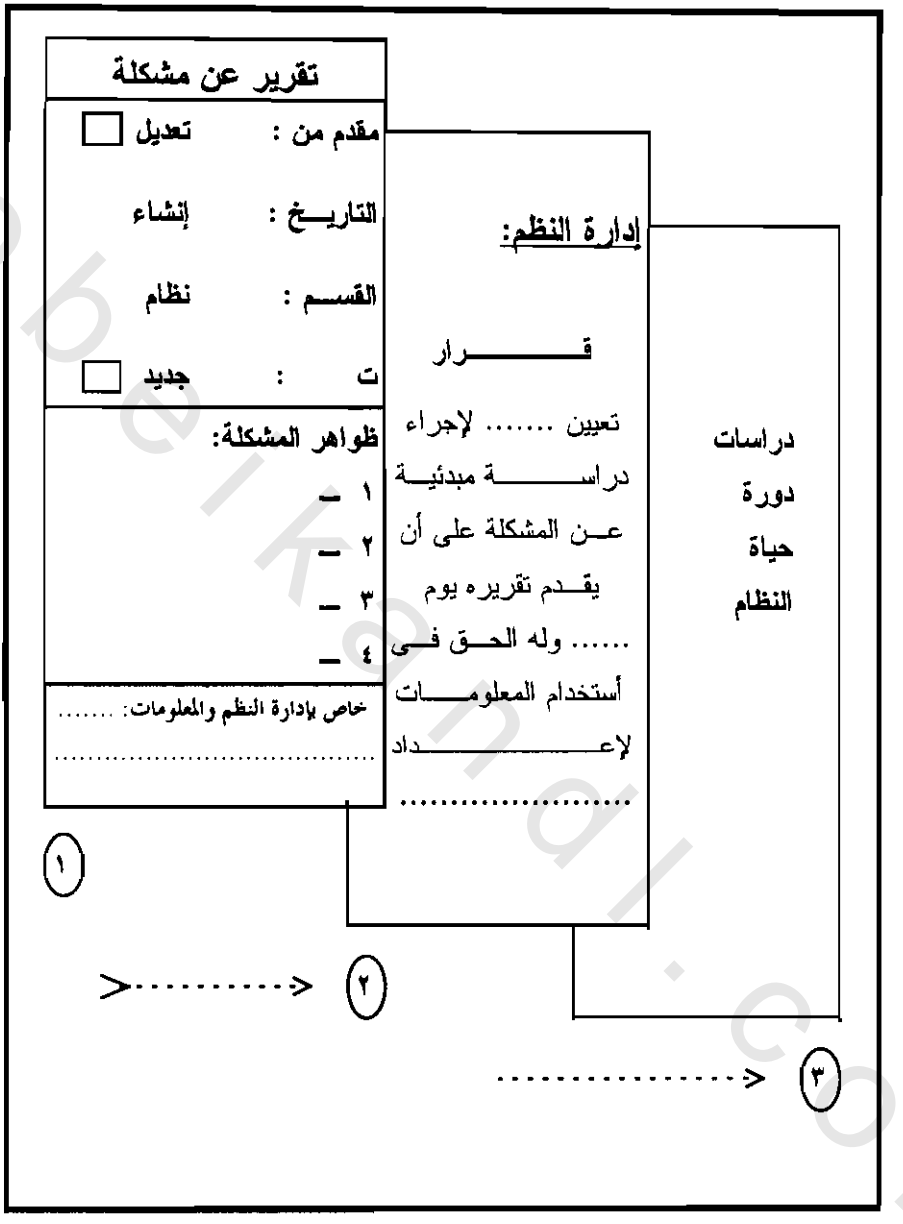
وتهدف الى التأكد من مدى جدوى النظام المقترح هل يحقق عوائد منظوره أو غير منظوره أكبر من تكلفته الفعلية ؟ وتنقسم دراسات الجدوى الى عدة أقسام:

أ - **دراسة الجدوى الأقتصادية والمالية :**

وتهدف الى دراسة العائد الاقتصادى من المشروع وأسلوب تمويله.

ب - **دراسة الجدوى التنظيمية :**

وهدفها الرد على تساؤل هل إدخال نظام المعلومات سوف يغير الـــهيكل التنظيمى للمؤسسة ومن هم المتضررون من إدخال النظام الجديد .



الشكل (١٣/١٤)

تقرير عن مشكلة فى نظام المعلومات

ج – **دراسة الجدوى الفنية :**

وتركز على الامكانيات الفنية المتاحة لتحقيق متطلبات المشروع .

د – **دراسة الجدوى التشغيلية :**

ما هى الفائدة على التشغيل الحالى من إدخال نظام المعلومات.

هـ – **دراسة الجدوى القانونية :**

وترد على التساؤل هل هناك أى مخالفات قانونية من إنشاء النظام .

وتستخدم نتائج هذه الدراسات فى أعداد تقرير يعرض على الادارة العليا لأتخاذ القرار فى أستكمال المرحلة أو إنهائها أو أعادة الدراسة . ودراسة الجدوى ليست نهائية بل عند تصميم النظام يجب إجراء دراسة جدوى شاملة وتفصيلية للمشروع قبل بدء الإنشاء.

٢ – **مرحلة جمع الحقائق :**

وتستخدم فيها الاساليب التالية :

أ – **الملاحظة :**

للعمل القائم من مدخلات ومعالجات ومخرجات وأفراد ونماذج وتخزين.

ب – **المقابلة :**

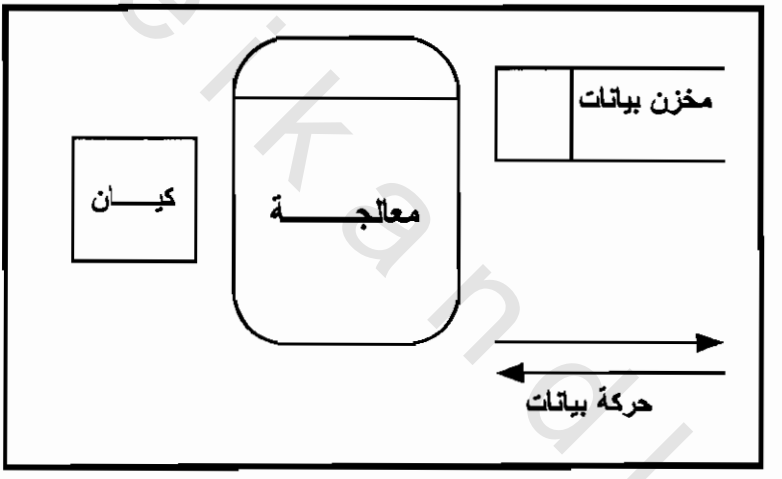
مع مسئولى نظام المعلومات على مختلف المستويات الادارية لاستطلاع آرائهم .

ج – **الاستقصاء :**

وتستخدم أستمارات الاستقصاء بديلا عن المقابلات فى حالة الانتشار الجغرافى لمواقع العمل بحيث يصعب أجراء المقابلات مع العاملين.

د – نمذجة النظام بأستخدام :

(١) تتم نمذجة النظام بأستخدام وسيلة خرائط تدفق البيانات DATA FLOW DIAGRAM ويجرى رسمها على مستويات تجريد مختلفة تبدأ من المستوى الشامل ثم تتدرج إلى التفاصيل رويدا وفق مستويات أكثر من التفصيلات مستخدمة فى ذلك أربعة رموز فقط على النحو المبين فى الشكل (١٤/١٤).



شكل (١٤/١٤)

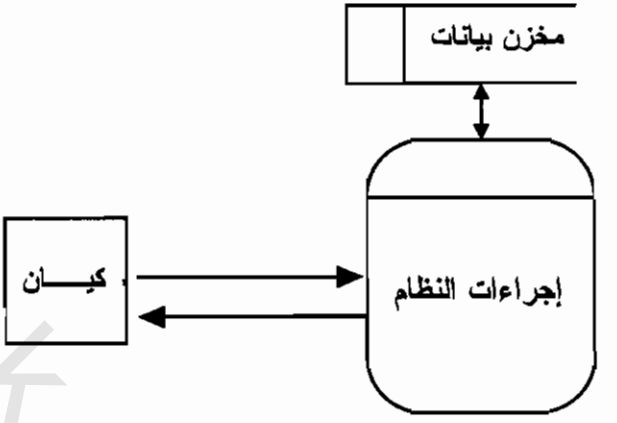
الرموز الموحدة لخرائط التدفق

(٢) قاموس بيانات النظام DATA DICTIONARY ويشمل توصيف كامل لجميع البيانات بالنظام سيان كانت بيانات ساكنة أو متحركة كما يشمل توصيف المعالجات ومخازن البيانات وسياسات النظام وقرارات وخيارات النظام .

(٣) دراسة وثائق المؤسسة وتحديد العلاقات بين الكيانات.

مثال:

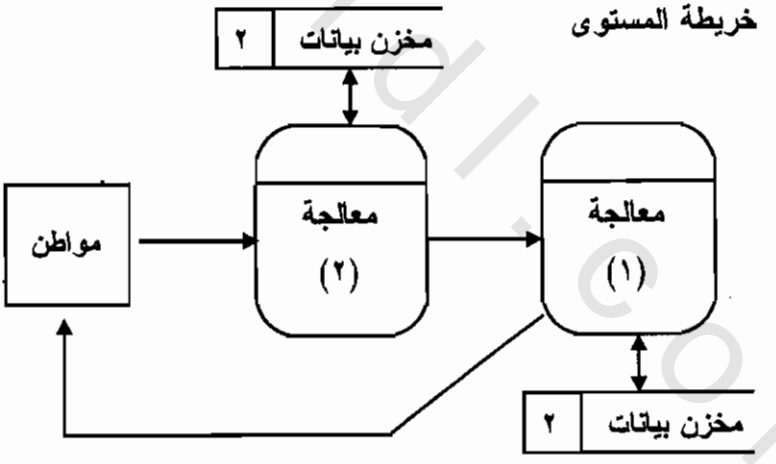
خريطة المفهوم الشامل DFD / 0 :



الشكل (١٤/١٥)

خريطة المفهوم الشامل

خريطة المستوى



الشكل (١٤/١٦)

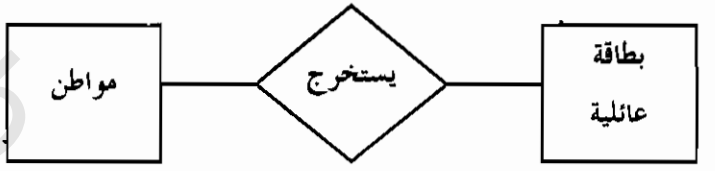
خريطة المستوى الأول

وهكذا تفصيلا أكثر وأكثر.

obeikandl.com

٣ - مرحلة اعداد تصور منطقى للنظام المقترح :

ويحدد خلال هذه المرحلة منطق النظام المقترح ووسائل ضبط البيانات وتصميم قاعدة بيانات أو ملفات النظام باستخدام اسلوب (الكيان-علاقة) E . R.



ثانيا : فكرة موجزه عن تصميم النظام :

١ - بعد اتمام تحليل النظام وعرض الحل المنطقى يتم اعداد تقرير للعرض على الادارة العليا لاخذ موافقة على استكمال تصميم النظام أو اتخاذ أى قرار أخر بشأن الموضوع .

٢ - تتولى مجموعة التصميم قراءة التقرير السابق واستشارة محلل النظم فيما تراه من غموض.

إن الهدف من التصميم هو استنباط المواصفات اللازمة التى تحقق متطلبات المستخدم والتى ظهرت خلال مرحله التحليل ، ومثلما كان التحليل يبدأ من اعلى لاسفل TOP - DOWN فأن التصميم الهيكلى هو افضل اساليب التصميم وفيه يجرى تقسيم النظام الى هياكل MODULES متكاملة مما يحقق سرعة انجاز التصميم ويضمن الدقة والوضوح وامكانية صيانة كل جزء دون التشابك مع الاجزاء الاخرى.

المراحل الاساسية فى التصميم :

يمكن تلخيص المراحل الاساسية فى التصميم على النحو التالى :

١ - تحديد الهدف النهائى للنظام .

٢ - تحديد حدود وقيود النظام .

٣ - تحديد نموذج منطقى اذا فشل محلل النظام فى الوصول الى الهدف .

بدائل التصميم :

الخيار الاول : الابقاء على النظام الحالى كما هو **DO NOTHING** .

الخيار الثانى : اجراء تطوير محدود فى النظام .

الخيار الثالث : تصميم نظام جديد .

والخيار الثالث يعتبر اعقد الحلول واكثرها تكلفة وهنا يجب دراسة ، هل سيتم انشاء نظام جديد تماما من البداية للنهاية ، أم هناك امكانية شراء نظام مماثل مع تحديد مزايا وعيوب وتكلفة كل بديل وعلى نتائج هذه المقارنة سوف يتخذ قرار.

خصائص التصميم الجيد :

يجب ان يتصف تصميم النظام بالخصائص التالية :

أ - **البساطة :**

كلما أمكن الى ذلك سبيلا باختبار اقصر الطرق وأبسطها للوصول الى تحقيق الهدف من النظام .

ب - **النمطية MODULES :**

مما يساعد على مرونة وبساطة النظام وامكانية الربط بينهما وكذلك الاستعانة بالانماط الجاهزة المتاحة .

ج - **إمكانية التوسع :**

وذلك لاستيعاب المتغيرات والمتطلبات الجديدة بديلا عن اعادة التصميم:

(١) امكانية الضبط والمراجعة .

(٢) سهوله التعامل .

(٣) التجانس .

(٤) الكفاءة .

(٥) الاقتصاد وقلة التكلفة كلما امكن .

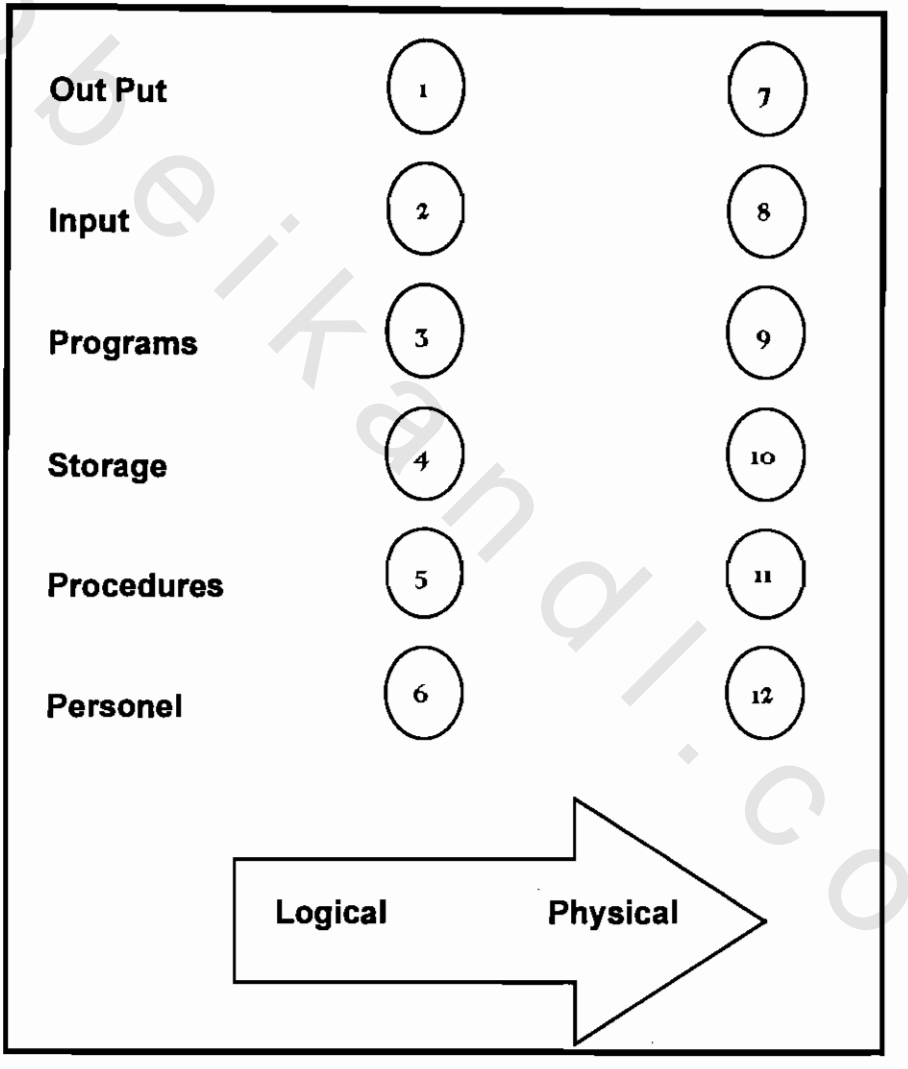
(٦) تأمين المعلومات بحيث لا يسمح بالدخول غير القانونى عليها .

obeikandi.com

(٧) التكاملية بين البيانات المطروحة عبر النظام بحيث تكون محدثة ومتكاملة.

خطوات التصميم :

يجرى التصميم وفق الخطوات الشكل (١٤/١٧).

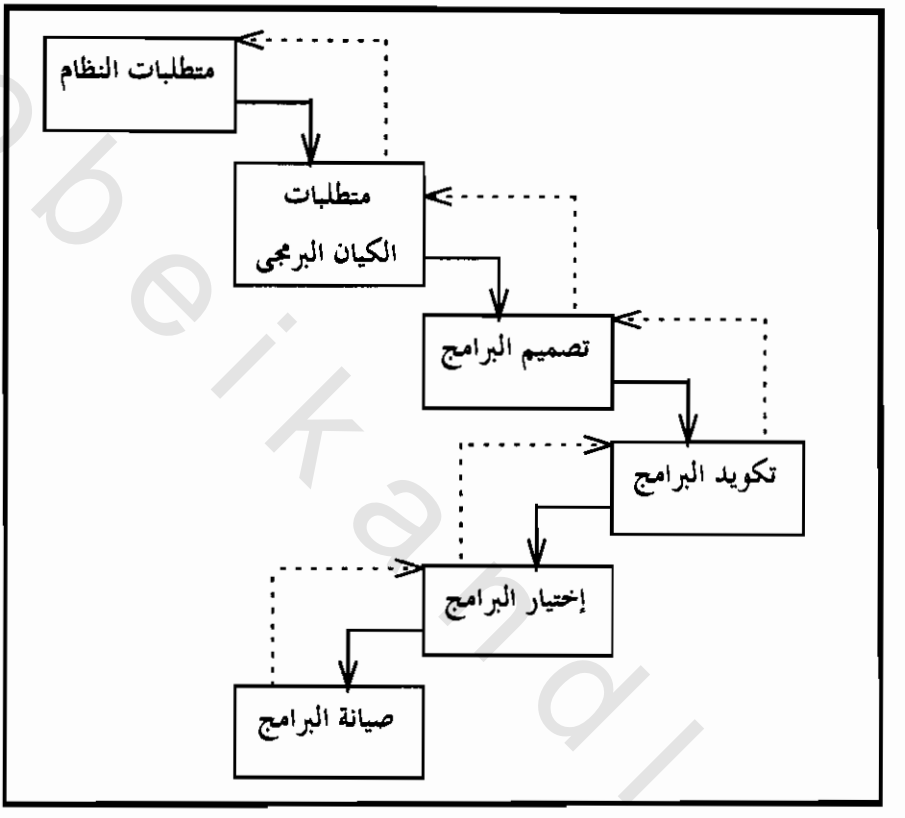


الشكل (١٤/١٧) خطوات تصميم النظام

obeikandl.com

دورة كتابة البرامج :

كما يوضحها الشكل (١٤/١٨) :



الشكل (١٤/١٨)

مساعدات كتابة البرامج :

تشمل مساعدات كتابة البرامج مايلى:

١ - خرائط التدفق .

٢ - تصميم الخرائط الهيكلية .

٣ - HIPO CHARTS .

obeikandl.com

٤ - شجرة القرارات .

٥ - GLS 4 .

تصميم مخازن البيانات :

قواعد البيانات :

ويجب أن تأخذ فى الإعتبار النقاط التالية:

أ - حجم البيانات .

ب - زمن الاستجابة .

ج - عدد مستخدمى النظام .

د - طبيعة النظام .

هـ - المدخلات .

و - المخرجات .

الاجراءات :

وتنقسم إلى

إجراءات التحكم والسيطرة وهى عبارة عن مجموعة ضوابط تبنى داخل نظام الكيان البرمجى لإقرار Security Privacy أى تحقيق تأمين المعلومات وخصوصيتها.

تحديد واجبات كل فرد فى النظام وأداء العمل وتحديد التقارير.

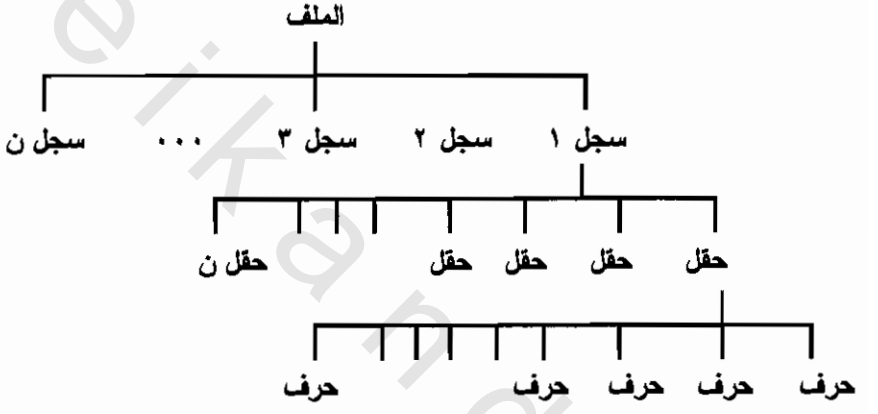
Integrity التكاملية وتشمل:

Security
تأمين النظام

الدقة
رقم الإختبار
حقل ليس صفر

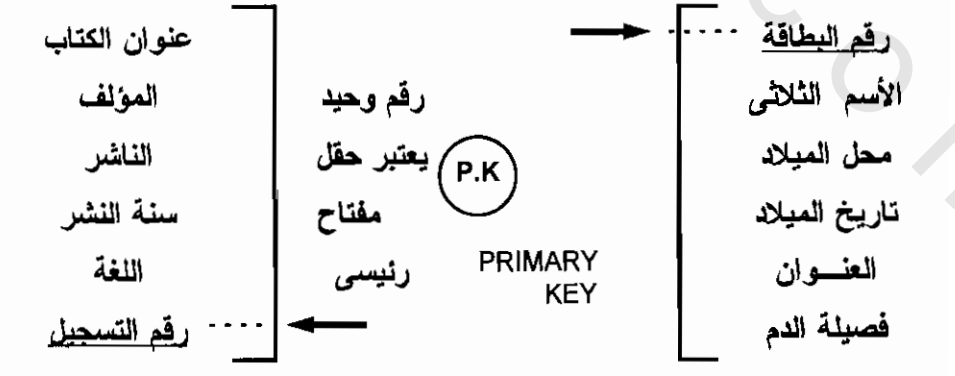
الخصوصية
وضع الضوابط
لإسترجاع بيانات الأفراد

obeikandl.com



أنواع حقول السجل:

إذا تدارسنا بيانات البطاقة الشخصية أو بيانات تسجيل كتاب بأحد المكتبـــات لوجدنا أن موصفات صاحب البطاقة أو موصفات الكتاب كما هى موضحـــة علـــى النحو:



وسوف نلاحظ ان هناك نوعين من الحقول حقل وحيد له قيمة لاتتكرر على مستوى السجل وعلى مستوى الملف كله مثل رقم البطاقة / الرقم القومى / رقم تسجيل الكتاب ، وكلاهما رقم فريد Unique ولذلك اذا سألت عن بيانات صاحب البطاقة رقم (كذا) أو ما هو عنوان واسم مؤلف الكتاب الذى رقم تسجيله (كذا) فما إيسر الوصول إلى البيانات المطلوبة لإن هذا النوع من الحقول يشير مباشرة الى باقى حقول السجل ويحدد الكيان المعبر عنه تحديداً دقيقا.

مثل هذه الحقول الخاصة تسمى حقل مفتاح رئيسى Primary Key ولاينفى وجود هذا النوع من الحقول وجود حقول اخرى تصلح للاشارة للسجل ويطلق عليها المفاتيح الثانوية Secondary Key ، كما يمكن الدمج بين اكثر من حقل غير مفتاحى لتكوين حقل مفتاحى Unique ، أما باقى الحقول فهى حقول يحتمل تكرارها فى أى عدد من السجلات فى الملف الواحد ، ففى ملف البطاقات الشخصية قد يتواجد عدد "ن" من السجلات لها نفس أسماء أصحاب البطاقات ، وربما نفس محل الميلاد ويحتمل نفس تاريخ الميلاد ولاتنفى احتمال وجود ولو عدد محدود جدا من الاشخاص لهم نفس العنوان وهكذا .

الرقم القومى	الأسم	محل الميلاد	تاريخ الميلاد	العنوان	فصيلة الدم

والحقل الفريد أو المفتاح الرئيسى هو احد الادوات المنطقية فى إسترجاع البيانات من الملفات كما سنأتى إلى ذلك تفصيلا.

أنواع الملفات:

تقسم الملفات إلى عدة أنواع أبرزها:

تساعدهم على مناقشة مغزى المعادلات أو جعل النتائج التى يتوصلون اليها اقرب مثالاً للانسان العادى ، مثل مقارنة مرور التيار الكهربى فى سلك بتدفق الماء فى انبوب ، أو يقولون ان تركيب الذرة يشبه النظام الشمسى.

معنى ذلك أن المحاكاة SIMULATION والنمذجة MODELLING على الحاسب تبغى فى المقام الاول المعادلة الرياضية الحاكمة للنظام ، اما التفسير فهو شىء ثانوى قد يكون وقد لايكون ، لذلك تختلف معالجة كلا من رجال الاقتصاد أو الفيزياء أو مهندسى الانشاءات لمعلوماته على الحاسب عن معالجة علماء الذكاء الصناعى اختلافا كبيرا للغاية ، فاول والثانى والثالث تشكل لديه المعادلات عالما مجردا يمكنه اكتشافه ويحول المعادلات الى برامج ويراقب كيف يؤدى النموذج عمله مع مختلف الافتراضات والمعطيات ، فى حين يبدأ عالم الذكاء الصناعى عمله بمعزل عن آلته ويدرس كيف يخزن المخ (الذاكرة البشرية) المعلومات ، كيف يعالج هذه المعلومات ــ كيف يسترجعها... بعدها يترجم هذا التصور الى برامج يختبرها على الحاسب الالكترونى ويشذبها ساعيا دائما لان يجعل الحاسب الالكترونى اكثر التصاقا بفكرته عن الطريقة التى يعمل بها ذهن الانسان.

والواقع ان المحاكاة تنظر للحاسب على أنه وسيلة قد يستخدمها الباحث اذا كانت معادلاته بالغة التعقيد وقد لا يستخدمها ، اما الاخر فيبدأ به لانه ذاته هو نموذجه وهدفه وغايته لهذا يعتمد علماء الذكاء الصناعى على الاستعارة من العلوم الطبية والبيولوجية وعلم النفس والحواس اكثر مما يعتمدون على النموذج الرياضى او العلمى.

والواقع ان ابحاث الذكاء الصناعى تركز على محاولة فهم كيف يعالج مخ الانسان الامور المختلفة ولم تركز بالطبع على كيفيه عمل واداء الخلايا العصبية التى يبلغ عددها عدة بلايين أو كيف تطلق نبضاتها الالكتروكيميائية مما يجعل الانسان تلقائيا يحدد اشياء لاقبل للحاسبات بها مثل: