الفصل الخامس

نظم قواعد البيانات الهيكلية

المقدمـــة

سبق اســتعراض برمجيــات نظم المعلــومات وتطبيقــاتها المختلــفة فى المكتبــات ومراكز المعلومات فى الفصل الثالث من هذا الكتاب مع التركيز على استرجاع المعلومات ذات الطبيعة الهيكلية المرتبطة بالفهارس والكشافات . كما سوف يستعرض الفصل الثامن أيضا أساسيات استرجاع المعلومات وتصــميم واجهات التفاعل مع المستخدمين وخاصة مــا يرتبط بالمعلومات الهيكلية والنصية على حد سواء .

وحيث أن خدمات المكتبة أو مركز المعلومات ترتكز فى الأساس على محتويات مجموعة المقتنيات ذات الوسائل / الوسائط المتعددة المتفرقة فى النظم التقليدية والمندمجة معًا إلكترونيًا فى النظم المتقدمة فى إطار نظم قواعد بــيانات النص والوسائط المتعددة ، كان من الضرورى تضمين هذا الفصل فى إطار الكتاب المقدم .

ويستعرض هذا الفصل عدة موضوعات مهمة يرتبــط أولها بنظم قواعد البيانات الهيكلية وخاصة العلاقية منها من حــيث التعريف والخصائص وتمثيل البيانات وهيكليــتها وتطبيقاتها فيما يتصل بمحتويات المكتبات ومراكز المعلومات .

وتعتبر قــاعدة البيانات مجموعــة معلومات منظمة بطــريقة يمكن فيها أن يخــتار برنامج كمبيــوتر البيانات المطــلوبة بسرعة . وبذلك يــفكر فى قاعدة البــيانات كنظام حفــظ بيانات إلكترونى . وقواعد البيانات الهيكلية مــنظمة بواسطة الحقول ، السجلات والملفات ، والحقل هو وحدة معلومات مفردة ، أمــا السجل فيمثل مجموعة حقول كاملــة ، ويشتمل الملف على مجموعة من السجلات .

وللوصول للمعلومات من قاعدة البيانات ، يحتاج المستخدم إلى نظام إدارة قاعدة بيانات DBMS الذى يمثــل فيه مجمــوعة برامج تســاعد فى إدخال البــيانات وتنظــيمها وإختــيارها واسترجاعها من قاعدة البيانات .

ويشتمل هذا الفصل على تحديد مفهوم وخصــائص نظام إدارة قاعدة البيانات الهيكلية ، تمثيل البيانات والربط بينهما على أساس مــستوى الواقع وتوصيف ما وراء البيانات والبيانات الطبيعية وطرق الــربط المختلفة المتعددة ، المشروط ، والمتبادل . . . إلــخ ، وهياكل البيانات الخاصة بالقوائم ، أسلوب الشجرة ، الهيكل الــشبكى ، والهيكل العلاقى مع مزايا وعيوب

كل أسلوب . كما استعرض هذا الفصل أيضًا أساليب الوصل المرتبطة بالمؤشرات ، الرصة، والطابور ، بالإضافة إلى تصميم قاعدة البيانات من حيث تطبيع البيانات وتحديد الكيانات وترابطها معا ، وتقرير خصائصها .

وفى الوقت الحالى ، صارت نماذج قواعد البيانات الهيكلية من شبكية وعلاقية تمثل توجهات متفاوتة حيث أنها تتضمن مشكلات كثيرة . ولكن بعض المنظمات والمؤسسات ومن بينها المكتبات ومراكز المعلومات الكبيرة استثمرت أموالاً ضخمة فى النظم المبنية على قواعد البيانات الهيكلية ، والتى يؤدى تعديلها إلى نظم جديدة إلى تحمل تكاليف باهظة وجهدًا كبيرًا ووقتًا غاليًا . لذلك قررت معظم هذه المنظمات الإبقاء على نظم قواعد البيانات الهيكلية المتواجدة لديها . إلا أن هذا القرار قد يكون مقبولاً على المدى القصير فقط ، حيث أنه قد يؤدى إلى مشكلات سوف تواجه هذه المنظمات فى المستقبل وخاصة عند إحلال النظم الحديثة فى قواعد البيانات وخاصة المبنية على التوجه الشيئى OODB محل النظم الأقدم ، مما يجعل من الصعب توفير قوى عاملة مهنية تتعامل مع النظم القديمة . وفى الوقت الحالى ، يستخدم حوالى ٩٥٪ من المكتبات ومراكز المعلومات نماذج قواعد البيانات الهيكلية ، مما قد يعنى عدم توافر استثمارات كبيرة فى هذا الصدد . ويلاحظ أنه من السهل إنشاء نظام قاعدة بيانات علاقى وإدارته ، إلا أنه فى المستقبل المنظور سوف تتحول هذه النظم إلى النظم الأحدث المبنية على التوجه الشيئى .

المفهوم والخصائص

تعرف قاعدة البيانات بأنها تجميع البيـانات التى بينها علاقة أو إرتباط ، أى أنها تجميع متحد ومنطقى من البيانات بطريقة منظمة يتم عرضها وإسترجاعها بأكثر من أسلوب ويسهل الاستفادة منها بواسطة المـتخدمين ، أى أن قاعدة البيانات هى مجموعة الملفات المرتبطة منطقيًا .

أما نظـام إدارة قاعدة البـيانات DBMS فيعرف بـأنه مجموعة مـن البرامج التى تـتيح للمستخدم إدارة وإنشاء قاعدة بيانات عن أحد النظم المعينة .

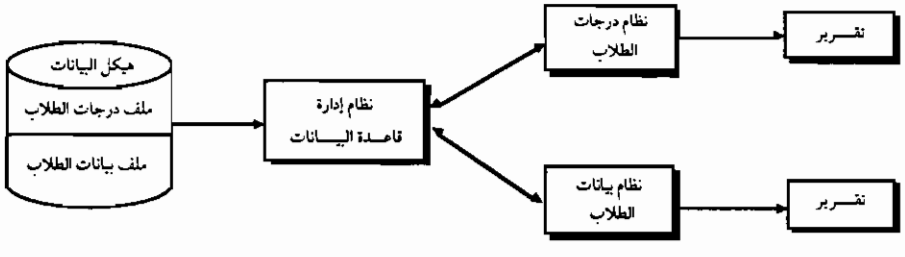
وبذلك تقسم نظم قواعد البيانات وفقا لما يلى :

١ - الارتباط بالطريقة التقليدية للبرمجة مع الملفات من حيث :

- توافر مـستخدم ينـشئ ملفًا خاصًا بـدرجات الطلاب علـى سبيل المثـال ، وحفظ الدرجات وطباعتها فى شكل تقرير خاص بها .
- تواجد مستـخدم آخر ينشئ ملفًا خاصًا عـن الطلاب أنفسهم مثلا ، يـتضمن اسم الطالب ، عـنوانه ، عمره ، فصـيلة دمه ، . . . إلخ بحـيث يحفظ ويطـبع فى شكل كشوف مثلاً .
- تكرار البيانات بين ملفى درجات الطلاب ، والطلاب ويستغرق ذلك وقتًا كبيرًا فى عملية الإنشاء والاسترجاع .

٢ - إكتفاء المدخل الحديث لـنظم قواعد البيانات بإنشاء مخزن بيانـات واحد لجميع الملفات المراد إنشائها وتعمل بشكل متصل القيم من خلال نظم إدارة قاعدة البيانات DBMS ، كما فى الشكل التالى :

شكل (٥ - ١) : نظام قاعدة البيانات



تمثيل البيانات والربط بينها

توجد ثلاث مستويات لتمثيل البيانات أو تجريدها ، هى :

١ - مستوى الواقع Reality :

تتضمن أى مؤسسة سواء كانت مدرسة أو شركة أو مكتبة . . . إلخ . أشياءً وعناصر متنوعة تعمل معا على تحقيق أهداف المؤسسة المعنية ، وترتبط بالبيئة المحيطة بها ، وتسمى هذه الأشياء «كيانات Entitities» . ويمكن ملاحظة وجود فروق بين صنوف أو أنواع الكيانات المختلفة ، كما تمثل مجموعة صنف الكيان التى لها نفس الخصائص كالمستعيرين ، الطلاب ، الوثائق . . . إلخ ، بينما يكون الكيان أحد عناصر هذه المجموعة مثل الطالب ، الكتاب ، . . . إلخ .

ويطلق على الخصائص المرتبطة بكل كيان سمات Attributes ، أو حقول Fields .

٢ - مستوى توصيف ما وراء البيانات Metadata :

يرتبط هذا المستوى بالتعامل مع مواصفات البيانات للوصول إلى النموذج المنطقى Logical Model لكيانات المؤسسة وتحديد الروابط Associates المتواجدة بينها . ويستخدم هذا المستوى بواسطة مدير أو إدارى قاعدة البيانات Database Administrator (DBA) الذى يحدد أسماء الحقول وأنواعها وأطوالها ، وينشئ قاموس بيانات Data Dictionary لها كما يقوم بوصف العلاقات والروابط بين الحقول الموجودة فى كل صنف أو نوع من السجلات .

٣ - مستوى البيانات الطبيعية Physical Data :

تمثل البيانات الطبيعية أو الفعلية الكيانات التى تنتمى إلى نوع أو صنف كيانات معين وتبين إحداثياته Entity Occurances ، فعلى سبيل المثال إذا كان الكتاب أو الطالب مثلاً أحد أصناف الكيان فإن عنوان الكتاب أو اسم الطالب يمثل حدوث الكيان الخاص الذى يعبر عنه فى سجل الكتاب أو سجل الطالب بما يحتويه أى منهما من حقول البيانات العقلية التى يتم تمثيلها فى الكمبيوتر ، على الرغم من تواجد سجل واحد لأى منهما يتم تعريفه فى

مستوى توصيف ما وراء البيانات السابق الإشارة لـه ، وبذلك تخزن البيانات الطبيعية أو الفعلية فى قاعدة البيانات ، بينما لا يخزن التوصيف فيها .

وفيما يتعلق بالربط بين البيانات فإن ذلك يمثل تحديد القيم الخاصة بوحدات البيانات المترابطة معًا والمعتمدة بعضها على بعض بصورة ما ، أى أن كل كتاب يكون له رقم مميز ، وعنوان ومؤلف وناشر وتاريخ . . . إلخ ، كما أن لكل طالب رقم واسم وعنوان . . . إلخ. وتوجد ثلاثة أنواع من الربط ، هى :

(١) **الربط الأحادى** : One Association بمعنى أن كل قيمة لوحدة بيانات (أ) يقابلها قيمة واحدة فقط لوحدة بيانات (ب) مثل :

(أ) وحدة بيانات ← (ب) وحدة بيانات

(رقم كتاب) أو (رقم طالب) ← (عنوان كتاب) أو (عنوان طالب)

أى أن لكل كتاب عنوان محدد ، ولكل طالب اسم معين ، يقابله قيمة واحدة لوحدة بيانات محددة .

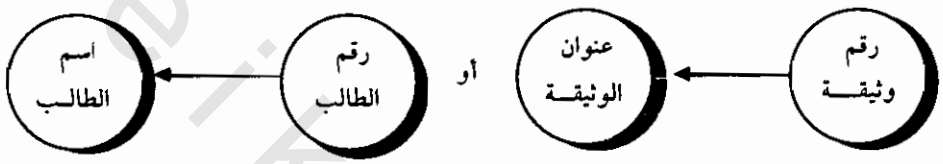
(٢) **الربط المتعدد** : Many Association بمعنى أن كل قيمة لوحدة بيانات (أ) مثلا يقابلها قيمة واحدة أو عدة قيم لوحدة بيانات (ب) . مثال ذلك ، أن كل وثيقة قد يكون لها موضوع الذى يشترك مع وثائق أخرى ، كما أن لكل طالب تمرين أو لم يؤدى التدريب .

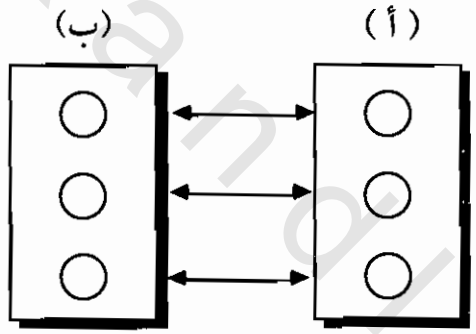
(٣) **الربط المشروط** : Conditional Association يعنى أن كل قيمة لوصف وحدة بيانات (أ) يقابلها قيمة واحدة لوحدة بيانات (ب) بناء على شرط معين . على سبيل المثال، فى قاعدة بيانات خاصة قد يخصص لوثيقة معينة موقع محدد على الرف مثلا قد لا يخصص لأى وثيقة أخرى . وبنفس الطريقة فى قاعدة بيانات الطلاب يخصص لكل طالب مكان أو كرسى خاص به لا يشاركه فيه طالب آخر .

(٤) **الربط الناضج** : Mutual Association يتواجد هذا النوع من الربط فى اتجاهين ، أى من وحدة بيانات (أ) إلى وحدة بيانات (ب) وبالعكس .

- **الربط من واحد إلى واحد** : One - to - one ، حيث أن كل قيمة لوحدة بيانات (أ) يقابلها قيمة واحدة لوحدة بيانات (ب) ، وكل قيمة لوحدة بيانات (ب) يقابلها قيمة لوحدة بيانات (أ) مرتبطة بها .

مثال ذلك رقم وثيقة يقابله عنوان وثيقة معينة ، وكذلك عنوان الوثيقة يقابله رقم هذه الوثيقة ، وفى حالة قاعدة بيانات الطلاب فإن رقم طالب يقابله اسم الطالب ، واسم الطالب يقابله رقمه المحدد .



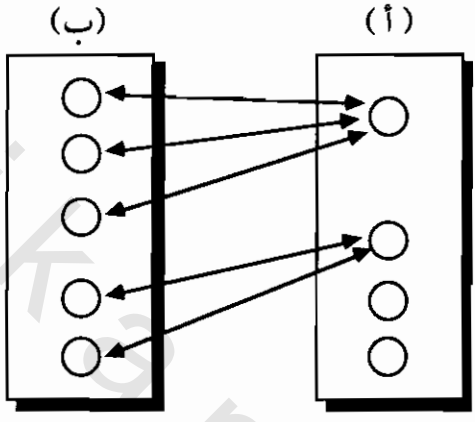


- **الربط من واحد إلى كثير** : One - to - many

كل قيمة وحدة بيانات (أ) يقابلها قيمة أو أكثر أو لا يقابلها قيمة لوحدة بيانات (ب) ، أى وجود قيمة واحدة لوحدة بيانات ، مثال ذلك ، فى حالة العلاقة بين رقم الطالب ورقم جلوسه فى الامتحان النهائى ، حيث يؤدى كل طالب امتحان نهائى واحد أو عدة امتحانات نهائية ، أو لا يؤديه ، فى حين يختص كل رقم جلوس بطالب محدد واحد .

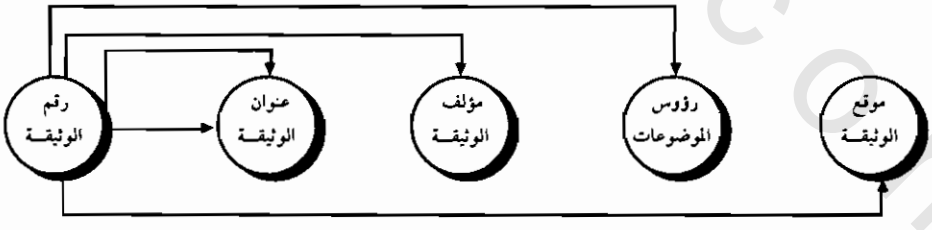
- **الربط الناضج من كثير إلى كثير : Many - to - many**

كل قيمة لوحدة بيانات (أ) يقابلها قيمة أو لا يقابلها قيمة لوحدة بيانات (ب) ، كما أن كل قيمة لوحدة البيانات (ب) يقابلها قيمة أو أكثر أو لا يقابلها لوحدة البيانات (أ) .



- **الربط بين الحقول :**

يتكون السجل من عدة حقول ، لذلك يصبح من الطبيعى أن يكون هناك ربط بين الحقول داخل كل سجل أو الحقل المفتاح Key field الذى يستخدم ويرتبط به حقول السجل التى توضح الإحداثيات Occurances المختلفة ، مثال ذلك :



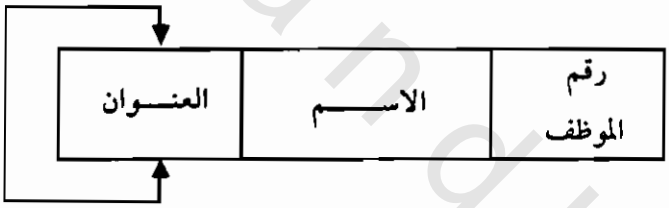
- **الربط بين السجلات :**

يمكن الربط الناضج بين السجلات عن طريق حقل المفتاح Key Field فى كل

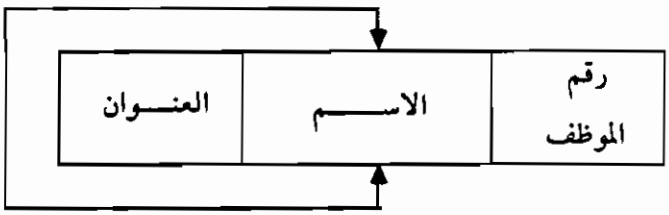
سجل ، ففى حالة قاعدة بيانات الطلاب ، نجد أن سجل كل طالب يقابله مقرر دراسى أو عدة مقررات أى أن العلاقة تكون واحد لكثير . كما أنه فى حالة نظام معلومات وثائقى ، يوجد لكل وثيقة رأس موضوع أو عددة رؤوس موضوعات أى أن العلاقة تكون واحد لكثير .

- **الربط الذاتى :**

فى بعض الأحيان ، توصف السجلات نوعية الكيانات مثل أعضاء هيئة التدريس أو الموظفين بالكلية أو المدرسة أو المكتبة فيحدد جنس الموظف أو المستعير فى المكتبة ، وقد يكون بعض موظفى المكتبة أو المدرسة أو الكلية متزوجين من بعض، أى يكون هناك ربط ذاتى Recussive Association داخل الكيان الذى يقوم السجل بتوصيفه مثل علاقة واحد لواحد فى حالة الزواج مثلاً :



نوع آخر من الربط الذاتى قد يكون واحد لكثير مثل الموظف الذى يتبعه عدد من المرؤوسين .



هياكل البيانات

تقسم البيانات إلى نوعين : نوع بيانات أولى أو أصلى Atomic مثل الأرقام الصحيحة التى يمكن إجراء العمليات الحسابية عليها ؛ النوع الثانى يمثل نوع البيانات الهيكلى Structured Data Type وهى البيانات التى يمكن تحليلها إلى مجموعة من الوحدات بعضها يرتبط بالصفر والبعض الآخر ذات طبيعة هيكلية وتربطهما معًا علاقات . ونستعرض فى العرض التالى أنواع هياكل البيانات ، كما يلى :

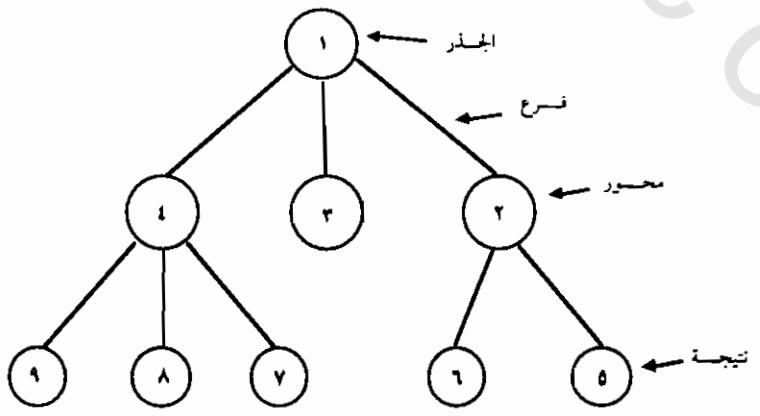
١ - القوائم المرتبطة Linked Lists :

تمثل القائمة المرتبطة مجموعة من وحدات البيانات المرئية التى ترتبط معًا من خلال عملية الترتيب التى تتم باستخدام روابط أو صلات Links أو مؤشرات Pointers موجودة داخل وحدات البيانات . وتتميز القائمة المرتبطة بوجود مؤشر لأول وحدة بيانات يسمى رأس القائمة Head ، وكذلك مؤشر فى آخر وحدة بيانات فى القائمة يطلق عليه ذيل القائمة Tail .

٢ - أسلوب الشجرة Tree :

يعتبر أسلوب الترتيب الشجرى من أساليب هياكل البيانات التى تستخدم بطريقة هرمية، تبدأ من الجذر الذى يتفرع إلى فروع لكل منها محور يتفرع أيضًا إلى فروع وهكذا إلى أن ينتهى بالثمار أو النتائج أى الأفعال المطلوبة ، كما فى الشكل التالى :

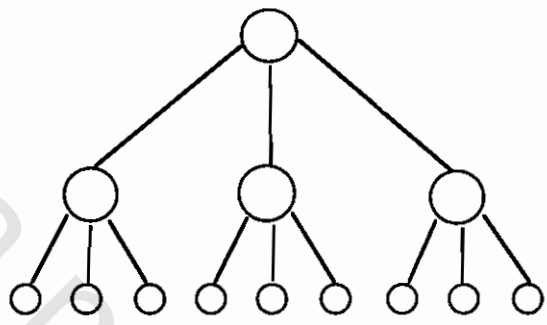
شكل (٥ - ٢) : الهيكل الشجرى



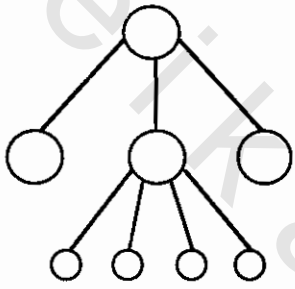
ويفسر الشكل السابق أن النظام المعين يتفرع تدريجيًا من الجذر الأصلى إلى مجموعة محاور تمثل نقاط الربط التى تسبقها .

وقد يكون الأسلوب الشجرى إما متزنا حيث تتساوى فيه تفريعات المحاور وما يرتبط بها من أوراق أو نتائج نهائية لكل تفرع ، أو أسلوب غير متزن حيث قد يتفرع محور واحد أو عدة محاور وليست كلها إلى النتائج النهائية ، كما فى الشكلين التاليين :

شكل (٥ - ٣) : الشجرة المتزنة



شكل (٥ - ٤) : الشجرة غير المتزنة



ويستخدم الأسلوب الشجرى فى علاقات البيانات المنطقية كما يرتبط بتكرار الحقل فى سجل معين . فمثلا فى حالة نظام قاعدة بيانات الطلاب يرتبط حقل رقم الطالب بالمقررات الدراسية المختلفة التى يقيد فيها هذا الطالب ، كما أنه فى قاعدة بيانات نصية يرتبط فيها رقم النص بالنصوص المختلفة المتصلة به .

وكل سجل سوف يكون له أطوالا إما ثابتة Fixed أو متغيرة Variable طبقًا لتصميم قاعدة البيانات الشجرية .

وتوجد علاقات بيانات مختلفة فى إطار الأسلوب الشجرى ، حيث أن سجل قاعدة البيانات كما فى حالة أعضاء هيئة التدريس بإحدى الكليات قد يحتوى على رقم واسم عضو هيئة التدريس والمسمى الوظيفى ، ومؤهلاته العلمية وهما حقلان ذات طبيعة تكرارية والمواد التى يقوم بتدريسها فى إطار كل مرحلة من مراحل الدراسة أى مرحلة البكالوريوس ، ومرحلة الدراسات العليا ، وبياناته الشخصية التى تمثل عمره وحالته الاجتماعية ، وعنوانه . . . إلخ . ويسهل تمثيل هذه العلاقات كلها من خلال الأسلوب الشجرى .

ويعتبر هذا الهيكل الشجرى من أقدم هياكل البيانات الذى يرجع إلى الستينيات من القرن العشرين . وشكل هذا الهيكل يشبه شكل الهيكل المستخدم فى تصميم البرنامج . ويوجد مكونان أساسيان له : المكون الأول يرتبط بالصناديق Boxes المستخدمة فى عرض أنواع السجل ، والمكون الثانى يرتبط بالخطوط Lines التى تصل بين صندوقين .

ويخدم الأسلوب الشجرى الغرض نفسه لخريطة العلاقة بين الكيانات Entity Relationship Diagram (ERD) حيث يحدد الهيكل المنطقى الشامل لقاعدة البيانات . وبذلك يستخدم الأسلوب الشجرى لتمثيل قاعدة بيانات هرمية التى تمثل علاقات واحد لواحد أو واحد لكثير . وفيما يلى مزايا وعيوب هذا الشكل :

المزايا :

- تحفظ كل البيانات فى قاعدة بيانات مشتركة ، تصبح المشاركة فى البيانات عملية ، كما تقدم خاصية الأمن التى تفرز بواسطة نظام إدارة قاعدة البيانات DBMS .
- تقليل جهد البرمجة وصيانة البرنامج ، حيث ينتج نظام إدارة قاعدة البيانات بيئة يمكن أن يحتفظ فيها باستقلالية البيانات .
- يساند الشكل الهرمى سلامة البيانات وتكاملها ، حيث أن الجزء الأصغر يشير دائمًا وبطريقة آلية إلى الجزء الأعلى منه مباشرة .
- يميل الهيكل الهرمى أن يكون كفء جدًا عندما تشتمل قاعدة البيانات كميات كبيرة من البيانات فى علاقة واحد لكثير ، وعندما يتطلب المستخدمون عدد كبير من التصرفات فإنهم يستخدمون البيانات التى تعتبر علاقاتها ثابتة عبر الوقت .

العيوب :

- يساعد الشكل الهرمى مصممى قاعدة البيانات عن طريق التخلص من مشكلات اعتمادية البيانات ، ولكن لا يزال المصمم يحتاج إلى معرفة عن الهيكل الطبيعى لتخزين البيانات . وعلى ذلك ، إذا رغب المصمم فى تغيير هيكل قاعدة البيانات ، فإن كل تطبيقات قاعدة البيانات المستخدمة للوصول إلى قاعدة البيانات يجب أن تتغير أيضًا مما يجعل تنفيذ قاعدة البيانات صعبا جدًا .

- تتطلب طبيعة قاعدة البيانات الهرمية علاقة واحد لكثير ، ولكن توجد حالات كثيرة التى تتطلب علاقة كثير لكثير .
- عند إنشاء شيء ما مثل خط ترتيب على فاتورة من جزئين آخرين مثل جزء الطلب ، أو أن السجل الذى يمثل Child المرتبط بأكثر من علاقة . قد يؤدى ذلك إلى صعوبات كبيرة ، حيث أن هذه العلاقة تعتبر صعبة الأداء مع الهيكل الهرمى .
- يمكن أن تكون إدارة قاعدة البيانات الهرمية صعبة جدًا ، حيث أن هذا الهيكل غير مرن ويمكن أن يعود إلى مهام إدارة نظام معقدة .
- عند كتابة التطبيقات لقاعدة البيانات فإن المبرمج يحتاج إلى معرفة هيكل قاعدة البيانات نفسها . وقد أضاف ذلك عدة مشكلات حيث يحتاج التطبيق أن يتغير عند تغير هيكل قاعدة البيانات . وعلى ذلك تعتبر قواعد استقلالية البيانات محدودة بواسطة الاعتمادية الهيكلية .
- يتوافر لنظام إدارة المعلومات نفسه برامج قليلة لإدارة قاعدة البيانات ، ولكنها يجب أن تشغل بصفة منفصلة عن نظام إدارة قاعدة البيانات DBMS ، مما يؤدى إلى جعل النظام غير مرن .
- لا توجد مجموعة مقننة بين كل البرمجيات التى تستخدم الهيكل الهرمى . ويعتبر نظام إدارة المعلومات من أكثر نظم إدارة قاعدة البيانات المشتركة المستخدمة مما لا يشكل مشكلة كبرى ، ولكن توجد بعض نظم إدارة قواعد بيانات أخرى مبنية على الهيكل الهرمى . ويمكن أن توجد إختلافات فى المفهوم والمصطلحات المستخدمة مع كل نظام إدارة المعلومات . وقد ثبت أن إمكانية النقل بين هذه الحزم صعب .
- لكى تنفذ قاعدة بيانات مستخدمة هيكلية رقمية ، يحتاج إلى معرفة مفصلة ومتعمقة بالبرمجة . ويجعل ذلك صعبًا على كثير من المستخدمين المبتدئين . ويطلق على هذا النظام بأنه النظام المنشأ بواسطة المبرمجين لهم أنفسهم .

٣ - الهيكل الشبكى Network Strueture :

يتكون هيكل البيانات الشبكى من مجموعة نقاط وصل أو ربط التى يطلق عليها محاور Nodes وفروع Branches كما فى الهيكل الشجرى السابق الإشارة إليه ، إلا أنها تختلف فى هذا المحور من أن الورقة Leaf أو النتيجة الواحدة التى تمثل الابن الواحد قد يكون لها أكثر من أصل أو أب ، أى أن العلاقة قد تكون واحد لكثير أو كثير لكثير . فعلى سبيل المثال ، عند تسجيل عدد ما من الطلاب للقيد فى المواد الدراسية المختلفة ، فإن العلاقة بين كل طالب والمواد الدراسية تعتبر علاقة واحد لكثير ، والعلاقة بين المادة الدراسية الواحدة وبين الطلاب المقيدين فيها تمثل علاقة واحد لكثير أيضًا ، فالطالب رقم (أ) يمكن أن يسجل فى المواد الدراسية أرقام ١ ، ٢ ، ٣ . . . إلخ ؛ والمادة (١) يسجل فيها الطلاب ، أ ، ب ، جـ . . . إلخ . كما توجد علاقة كثيرة لكثير مع الطلاب بعضهم ببعض .

ويشبه الهيكل الشبكى الهيكلية المستخدمة فى الهيكل الهرمى السابق ، والإختلاف الرئيسى بين الهيكلين يتمثل فى أن الهيكل الشجرى يستخدم وصل بين الأصل والفرع أو الابن والأب حيث يحتاج الابن إلى أب متواجد بالفعل ، أما فى هيكل قاعدة البيانات الشبكية فإن الابن يمكن أن يكون له أكثر من أب . وخلافًا عن الهيكل الهرمى فإن الهيكل الشبكى يستخدم سجلات وعلاقات بين البيانات التى تمثل بواسطة وصلات .

وتستبعد هياكل البيانات الشبكية مشكلات التكرار ولكن قد يؤدى التغيير فى هيكل قاعدة البيانات إلى إعادة الهيكلية لهيكلية قاعدة البيانات . ويمكن إضافة مجموعات جديدة بسهولة بإنشاء وحدات بيانات جديدة وربطها مع البيانات المتواجدة .

وفيما يلى المزايا والعيوب المرتبطة بالهيكل الشبكى :

المزايا :

- تعزيز وتقوية سلامة البيانات وأمنها ، حيث يجب على المستخدم أن يعرف السجل والحقول المتضمنة .
- يعتبر نوع الوصول إلى البيانات ومرونتها أحسن جدًا مما يتوافر فى هيكل قاعدة البيانات الهرمى فى نظم الملف . ويمكن لأحد التطبيقات الوصول إلى السجل

الأساسى وكـل ما يرتبط به من سـجلات وحقول فى إطار المجمـوعة . وعلى ذلك ، إذا كان للـسجل الفرعى المـعين أكثر من تطبـيق ، فمن الممكـن الانتقال مباشرة من سجل تطبيق لآخر .

- تكـون الـعـلاقات مـن كثـير لكثير أسهـل فى التنفيـذ من هيكل قاعدة الـبيانات الهرمية .
- تتحقق اسـتقلالية البيانات بـدرجة كافية فيما يـتصل بفصل البرامج مـن تفاصيل التخزين الطبيعية المعقدة . وعلى ذلـك ، فإن التغييرات فى خصائص البيانات لا تتطلب تغييرات فى برامج التطبيق .

العيوب :

- لكى تصمم وتسـتخدم قاعدة بيانات بتوظيف الهيكل الشـبكى ، من المهم أن يلم المستـخدم بمعرفة مـفصلة عن كـيفية هيكـلة قاعدة البـيانات ، أو لن يقـدر على الوصول إلى مزايا النظام بكفاءة .
- لا ينتج هيكل البيانات الشبكى استقلالية هيكلية ولكنه يقدم استقلالية بيانات .
- يحتاج مبرمجو التطـبيق إلى معرفة مفصلة عن هياكل قاعدة البـيانات الداخلية قبل تمكنهم من استخدامها .
- كما فى حالة هيكـل البيانات الهرمى ، فإنه لكى يكتـسب الوصول إلى سجلات نظام معين يجب الإبحار خلال كل السجلات واحد بعد الآخر .

٤ - الهيكل العلائقى Relational Structure :

تعتبر قاعدة البيانات الشبكية صعبة فـى الصيانة وعلى وجه الخصوص عندما تنمو أكثر تعقيدا . على سبيل المثال ، الحاجة لتشغيـل تساؤلات عشوائية مطلوبة من المبرمجين الخبراء حتى فى حالة إنتاج تقارير سهلة وبسـيطة جدًا . وينتج من ذلك مشكلات عديدة ، إذا أنجز أى تغيير لهـيكل قاعدة البيانات . وعلى ذلـك ، يصبح الهيكل العلاقى أو الـعلائقى أكثر ملاءمة لمخاطبة هذا النوع من المشكلات .

وتشتمل قاعدة البيانات العلاقية على مجموعة جداول يطلق عليها علاقات ، حيث يخصص لكل منها اسم أو عنوان فريد ، وتصبح الحقول المفتاح الرئيسى فى كل جدول وترتبط بعضها ببعض ويحدد ذلك السبب فى تسميتها قاعدة بيانات علاقية . وتستخدم هذه الجداول للإحتفاظ بالمعلومات . أى أن نظم قواعد البيانات العلاقية هى التى تتلقى البيانات من المستخدم فى هيئة جداول مثل :

شكل (٥ - ٥) هيكل قاعدة البيانات العلاقية

Doc #	DocAut	DocTitle	DocPub
Doc. 0030	.....	.....	.....
Doc. 0331	.....	.....	.....
Doc. 1334	.....	.....	.....

وتشتمل قاعدة البيانات الهرمية على لغة التساؤل الهيكلية SQL ، التى تنقسم إلى لغة تعريف البيانات Data Definition language ولغة تداول البيانات Data Manipulation Language (DML) .

(١) لغة التساؤل الهيكلية SQL :

تعتبر لغة البناء والاستعلام التى عن طريقها يمكن إنشاء قواعد البيانات فى شكل جداول وإجراء العمليات على هذه الجداول . ويتم ذلك عن طريق أوامر أو تعليمات هذه اللغة التى تتمثل فى :

- Create أمر إنشاء جدول جديد .
- Alter أمر يغير تركيب الجدول ، مثل إضافة حقل جديد .
- Drop أمر إسقاط جدول من قاعدة البيانات .
- Insert أمر إدخال بيانات جديدة للجدول .
- Select أمر إختيار بيانات معينة وعرضها أو استغلالها فى عمل معين .

- Delete أمر حذف أو إلغاء بعض البيانات غير المفيدة .
- .... إلخ .

(٢) لغة تعريف او وصف البيانات Data Definition Language (DDL) :

تمثل هذه اللغة تحديد جدول الأساس Base Table الذى يتكون من أعمدة ، كل عمود هو حقل لهذا الجدول ، وتكون هذه الحقول صفوف من البيانات .

ويكون لجدول الأساس مفتاح رئيسى Primary Key ، كما يتسم جدول الأساس بخصائص مثل :

- ترتيب الصفوف Row ordering يمكن أن يتغير فى أى وقت دون التأثير على البيانات أو المعلومات المطلوبة .
- ترتيب الأعمدة Column ordering يتم حسب اللغة المعينة من اليسار إلى اليمين كما فى حالة اللغة الإنجليزية ، أو من اليمين إلى اليسار كما فى اللغة العربية .

وفى هذا الجدول يمكن استخدام لغة التساؤل الهيكلية SQL كما يلى :

Create table base : table name

(Column - definition [column definition]...)

[Primary - Key definition], [foreign - key definition].

ويعتبر ما بين القوسين [ ] اختيارى للمستخدم ، ويعتبر المفتاح الأصلى primary key أحد حقول الجدول الذى ترتب فيه البيانات داخل الجدول عن طريق هذا الحقل ، كما ترتب صفوف الجدول تبعًا لهذا الحقل حيث يكون كما يتضح من اسمه بمثابة مفتاح الحصول على أى معلومات أو استعلام من الجدول ، لذلك يجب أن يكون أحد حقول الجدول . ويشتمل هذا الحقل فى كل صف على قيمة واحدة فقط له لا يمكن تكرارها ،

مثال ذلك : إنشاء جدول عن الوثائق تتم أوامره كما يلى :

Create Table Doc

(Doc # = 5 Characters) Doc Title = 40 Charcicter,

Doc Type = 3 Character, Doc Date = 10 Charcter date

يتضح فى المثال السابق ، أنه تم إنشاء جدول أساسى للوثائق هو Doc ويحتوى هذا الجدول على أربعة حقول هى : رقم الوثيقة Doc.# ، عنوان الوثيقة Doc. Title ، نوع الوثيقة Doc Type ؛ وتاريخ الوثيقة Doc. Date .

ويعتبر المفتاح الرئيسى لهذا الجدول حقل رقم الوثيقة Doc.# ، لذلك يجب أن يأخذ هذا الحقل قيمة المفتاح الأجنبى Foreign Key الذى يمثل أحد حقول الجدول ، ولكنه فى نفس الوقت مفتاح أصلى Primary Key فى جدول آخر ، وعند الرجوع إلى الجدول الآخر الذى يكون فيه مفتاح أصلى والتأكد من صحة القيمة التى نريد إدخالها نجد أنها موجودة فى هذا الجدول بالفعل .

وتشتمل أنواع البيانات Data Types على : بيانات رقمية Numeric إما أن تكون Integer ، Smallint ، عشرية Decimal ، و Float ؛ بيانات الحروف String Data التى تتضمن الحروف Character ، الرسوم Graphic ، وبيانات التاريخ/ الوقت Date Time Data التى تتضمن التاريخ ؛ والوقت .

(٣) لغة تداول البيانات Data Manipulation Langauage (DML) :

تستخدم هذه اللغة مجموعة من الأوامر فيها مثل :

- Select أمر إختيار .
- Update أمر تحديث .
- Delete أمر حذف .
- Insert أمر إضافة .
- ... إلخ .

وتشتمل على التساؤلات Query ، الاسترجاع Retrive ، ربط Join ، كما تتضمن عدد من الوظائف المبنية فيها مثل :

- Count أمر العد .
- Sum أمر الجمع .
- Average أمر المتوسط .

- Max أمر الأكبر .
- Min أمر الأصغر .

وفيما يلى المزايا والعيوب المرتبطة بالهيكل العلاقى :

المزايا :

- يحقق هيكل قاعدة البيانات العلاقى استقلالية كل من البيانات والهيكلية .
- هذه الهيكلية أسهل فى تصميم قاعدة البيانات وإدارة مكوناتها ، حيث يمكن التغاضى عن خصائص تخزين البيانات الطبيعى الفعلى والتركيز على الرؤية المنطقية لقاعدة البيانات.
- تشتمل هذه الهيكلية على لغة تساؤل قوية ومرنة جدًا يطلق عليها لغة التساؤل الهيكل SQL .
- تميل قاعدة البيانات العلاقية إلى طلب برمجة أقل من أى هيكلية قاعدة بيانات آخرى.
- تبني لغة التساؤل الهيكلية SQL على العمل الذى أنجزته شركة IBM منذ عام ١٩٧٤ والتى أصبحت تمثل منتجا مهما جدًا منذ ذلك التاريخ . وقد طبق معهد المعايير القومى الأمريكى ANSI معيارًا فى عام ١٩٨٦ ساند هذه اللغة وجعلها لغة معيارية .
- تسمح واجهة التفاعل للمستخدم بالتفاعل مع البيانات ، حيث يمكن تصميم واجهة التفاعل من القوائم Menus ، التساؤل ، العمليات Operations ، ومولدات التقارير Report generators ، . . . إلخ .
- يؤدى محرك لغة التساؤل الهيكلية SQL Engine أعمال قاعدة البيانات الصعبة على الرغم من أن ذلك يكون مختفيا إلى حد كبير من المستخدم .
- يعتبر نظام إدارة قاعدة البيانات العلاقى RDBMS الجيد أكثر تعقيدا من نظام إدارة قاعدة البيانات DBMS العادى الموجود فى هيكلى قاعدة البيانات الهرمية وقاعدة البيانات الشبكية . وتبعًا لذلك ، يجعل نظام إدارة قاعدة البيانات العلاقى RDBMS الجيد فى الإمكان إخفاء تعقيدات النظم الطبيعية من مصمم قاعدة البيانات ومستخدمها النهائى .

العيوب :

- تعتبر قواعد البيانات العلاقية سهلة جداً فى الاستخدام ، وتخفى مهام معقدة كثيرة من المستخدمين . ولكن يؤدى ذلك بتكلفة معينة . ولتشغيل نظم قواعد البيانات العلاقية يحتاج إلى قدر كبير من قوة المعالجة ، ويعنى ذلك أن هذه الهيكلية قد تكون بطيئة إلى حد ما .
- على الرغم من أن قواعد البيانات العلاقية تعتبر سهلة جداً لأى شخص لكى ينفذها حيث لا يقوم بتصميم قاعدة بياناته بطريقة ملائمة ، إلا أن ذلك قد يكون مقبولا لقواعد البيانات الصغيرة فقط ، ولكن بنمو قاعدة البيانات تبدأ فى الظهور المشكلات النابعة من عدم التقيد بالتصميم .

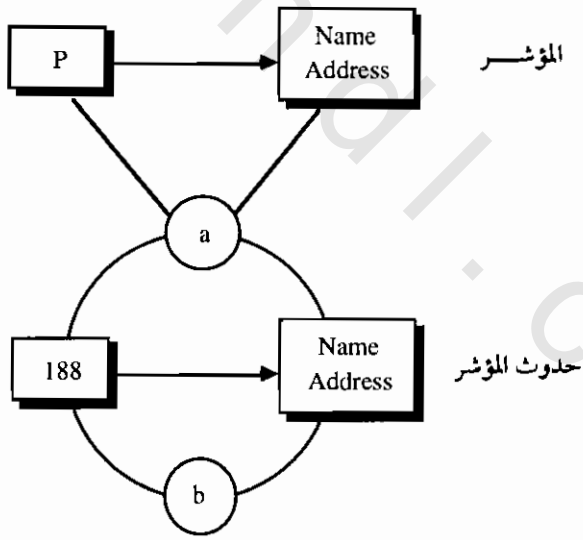
أساليب الوصل فى قواعد البيانات

يوجد ثلاث أساليب رئيسية للوصل فى قواعد البيانات ، هى :

١ - المؤشرات Pointers :

المؤشر هو نوع البيانات التى تتمثل فيه قيمته فى عناوين أخرى من البيانات ، ومن أمثلة ذلك ما يلى :

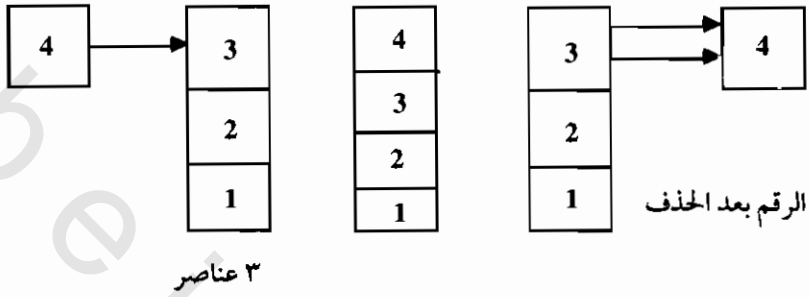
- **المثال الأول** : إذا كانت البيانات المراد البحث عنها تقع فى الصف الخامس من قائمة معينة . فإن قيمة المؤشر تكون خمسة ، ويأخذ المؤشر قيمة تتراوح من صفر إلى واحد إلى كثير Many ويرتبط ذلك بعدد الحروف أو البيتات Bytes فى الذاكرة .
- **المثال الثانى** : يوضح ربط محور معين Node والمؤشر ، وبذلك نجد أن قيمة المؤشر هى عنوان الذاكرة Memory Address لأول بايت Byte فى البيانات الموجودة فى هذه النقطة وقد يكون ذلك رقم ١٨٨ مثلاً . كما فى الشكل التالى :



٢ - الرصة Stack :

تعرف الرصة بأنها قائمة بيانات Data List التى يتم تحديد صفاتها الرئيسية عن طريق القواعد التى تحكم عمليات الإدخال أو الحشر Insertion أو الحذف Deletion لعناصرها .

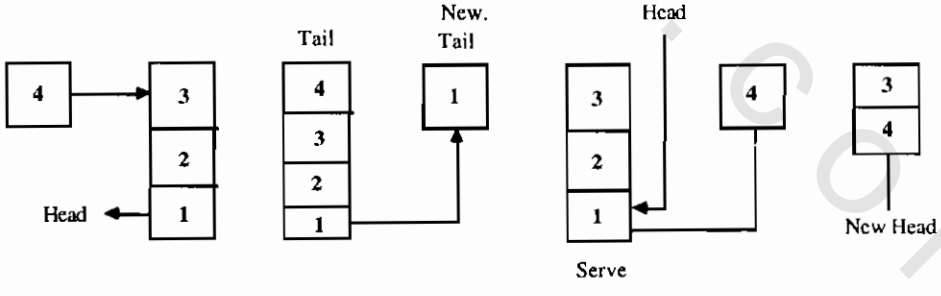
ومن القواعد المستخدمة فى ذلك « قاعدة الداخل أخيراً يخرج أولاً Last - in - First - out (LIFO) » ، كما فى المثال التالى :



٣ - الطابور Queue :

يعتبر هذا الأسلوب من أعلى مستويات تمثيل البيانات ، وله أهمية كبيرة فى العمليات الحسابية ، ويتميز بقواعد التحكم فى البيانات مثل قاعدة « الذى يدخل أولاً يخرج أخيراً FIFO» وتسمى عملية إضافة عنصر جديد إلى الطابور «دخول الطابور ENQUEUE» وعملية الحذف أو التحريك بالحذف DQUEUE ».

ويمكن أن يكون الطابور جزءاً من مصفوفة معينة كما فى المثال التالى :



تصميم قاعدة البيانات

١ - تطبيع البيانات Normalization :

تعتبر مرحلة تطبيع البيانات من أهم مراحل التصميم المنطقى وخاصة لقاعدة البيانات العلاقية . وفى هذا الإطار يحتوى كل سجل على مجموعة من الحقول التى تمثل كيانات Entities النظام المراد تصميمه ممثلا فى قاعدة بيانات .

ومن نماذج التطبيع Normalization Forms ما يلى :

(١) **نموذج التطبيع الأولى** : يتم فى هذا النموذج التخلص من المجموعات التكرارية Repeating groups الذى يشتمل على عدة مشكلات ترتبط بما يلى :

- الإضافة Insertion ، أى إضافة تاريخ جديد تتم فيه عملية من العمليات ، التى لا تتم إلا بعد تحديد رقم الكيان الأصلى .
- الحذف Deletion ، عند حذف بيانات فإنها تختفى ويصعب الرجوع إليها فيما بعد .
- التحديث Updating عند تعديل أحد الكيانات المتكررة فى أكثر من سجل .

(٢) **نموذج التطبيق الثانى** : يستخدم هذا النموذج للتخلص من الحقول المعتمدة جزئيًا على أحد عناصر الحقل المركب بتقسيمها إلى علاقتين أو جدولين .

٢ - تحديد الكيانات Specifying Entities :

من المفيد فى تصميم قاعدة بيانات البدء من مستوى تصميم عالٍ جدًا ، من حيث :

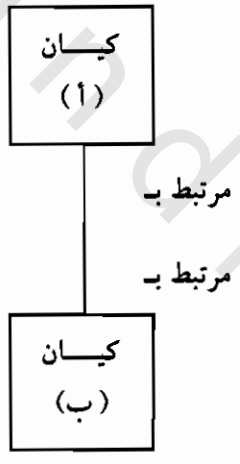
- تحديد الكيانات أو الموضوعات التى تتضمنها قاعدة البيانات بدلاً من الاستنتاجات المطلوبة .
- التفكير فى الموضوعات أى الكيانات ، وبالتالى البيانات بطريقة منفصلة عن الاعتبارات العملية مثل من سوف يدخل البيانات .
- التفكير فى الموضوع بصفة مستقلة عن أى برمجيات قاعدة بيانات معينة أو حتى الاستخدام الآلى لها .

- تجنب الارتباك بين إطار البيانات وتفاصيل التنفيذ .

ويعبر عن الكيانات فى أسلوب يطلق عليه علاقات الكيان والخاصية - Entity attribute - relationship . والكيانات هى الأشياء التى تشتمل على اهتمام معين لمصمم قاعدة البيانات . ويمكن التفكير فى الكيانات كموضوعات تغطيها قاعدة البيانات ، إلا أن الكيان يمثل لفظا أكثر دقة . ويوجد تصميم للأشياء التى يجب أن تمثل تفسيراً محدداً جداً ، كما قد يكون الكيان شيئًا محدداً ملموسًا ، أو شيئًا مجرداً .

٣ - ترابط الكيانات :

سبق استعراض موضوع تمثيل البيانات والربط بينها فى هذا الفصل ، وتمثل العلاقة ترابط جوهرى بين كيانين ويحدد ذلك بواسطة خط ـــــ يصل كيانين معًا . ولكل علاقة نهايتين حيث يوجد اسم لكل منهما ، كما فى المثال التالى :



وتوجد عدة درجات من العلاقات تتمثل فى :

أ - علاقة واحد لواحد 1 : 1

ب - علاقة واحد لكثير 1 : M

جـ - علاقة كثير لكثير M : M

٤ - تقرير خصائص الكيانات :

يرتبط ذلك بإضافة التفاصيل التى يشتمل عليها الكيان المعين . وقد يقرر ذلك بكتابته على خريطة حتى يمكن ترابطها وتمثيلها مع الكيان .

وتمثل الخصائص التفاصيل المختلفة التى يتسم بها الكيان ، وتمثل وصف من أوصافه ، كما أنها تعبر عن تضمينات الأشياء التى نريدها عن الكيان المحدد . وبذلك تمثل الخصائص Attributes الخاصة بالبيانات التى يجب أن تحفظ عن الكيان وتمثل الصفوف Rows فى الجداول .

المراجـــع

(1) Bhavani Thuraisingham Consulting, ed. Handbook of Data Management. CRS Press, 1998.

(2) Coronel, C. and Rob, P. Database Systems : Design, Implementation, and Managmeent. 2nd ed., Massachusetts : Boy & Fraser, 1995.

(3) Date, C. J. An Introduction to Database Systems. 6th ed. Massachusetts : Addison – Wesley, 1994.

(4) "Design Your Own Database", [http : // Seastorm ncl.ac.uk/itti/ design.html]

(5) Foley, Marry Jo. "Hot New Database Technologies" Datamation, Vol. 42, No. 15 (September 1996), pp. 44-50.

(6) Korth, H. F. and Silberschotz, A. Database Systems Concepts, 2nd ed. New York : MgGraw-Hill, 1991.

(7) "Object-Relational Dabase Managers" **Distributing Computer Monitor**, Vol. 10, No. 2, (Feb. 1995).

(8) Ozkaraham, F. Database Management : Concepts, Design and Practice. Englewood Cliffs : NJ : Prentice-Hall, 1990.

(9) Stodder, David. "The Database Dozen" **Database Programming & Design**, Vol. 9, No. 13 (December 15, 1996), pp. 10-23.