الفصل الخامس
استخدام المعاملات
Operators

تعلمنا في الفصل السابق بعض الاستخدامات البسيطة لأوامر SQL Query مثل Select.....from ورأينا كيف نستخدمهم في الجملة فى هذا الفصل سوف نعرف.

- استخدام تعبير Expression داخل الاستعلام Query
- استخدام المعاملات (Operators)

استخدام تعبير أو Expression داخل الاستعلام Query

سوف تجد في هذا الكتاب أننا نبدأ باستخدام أمر ما بطريقة بسيطة ثم نتوسع في الاستخدام في الفصول التالية بحيث نبني على ما بدأنا به حتى نجد الأمور كلها بسيطة . وبأسلوب البرمجة وقواعد البيانات فهناك عدد من المسميات التي نستخدمها ولا نفكر في ترجمتها أو ما ترمي إليه في اللغة العربية لذلك سوف أتطرق إليها قليلا ولكننا سوف نتحدث معا بلغة البرامج وقواعد البيانات وباستخدام اللفظ دونما ترجمة إذا ما تأكد لي أنك عزيزي القارئ تتحدث معي بنفس اللغة .

اعلم عزيزي القارئ أن ترجمة كلمة **expression** إلى "مصطلح" أو "تعبير" غير دقيقة، فلو رجعت إلى الفصل السابق سوف تجد أنك استخدمت هذه الكلمة فهي تعني أنها اسم يحجز مساحة ويتم تحميله بقيمة ، انظر إلى هذه الجملة :

```
select class from schedule;
```

الحقل المسمى **class** يحمل قيمة من الجدول **Schedule** ولهذا فنحن نسميه **expression** وهكذا في الجملة التالية :

```
select name, age, gyear, specialty from teachers;
```

سوف تجد كل من **name, gyear, age, speciality** عبارة عن مصطلحات **expression** سوف نطلع على قيمة لها بمجرد تنفيذ الجملة السابقة .

هل تريد مزيدا من الإيضاح ، أو كما يقول الفلاسفة عن بضاعتهم أنها تعني توضيح الواضح .

استخدام المعاملات Operators :

المعاملات أو **Operators** هنا معناها العلامات التي نستخدمها داخل تعبير معين في جملةSQL لتحدد كيفية تنفيذ شرط معين للحصول علي البيانات التي ينطبق عليها هذا الشرط . وفيما يلي سنشرح هذه المعاملات في مجموعات متجانسة مثلا المعاملات الحسابية مثل علامة الجمع والطرح والمعاملات المنطقية ومعاملات المقارنة مثل "أكبر من"و"أصغر من"وغيرها .

استخدام المعاملات الحسابية **Arithmetic Operators** :

المعاملات الحسابية هي علامات الجمع والطرح والضرب والقسمة وعلامة النسبة المئوية وغير ذلك .

علامة الجمع "+"

يمكن استخدام علامة الجمع بأكثر من طريقة وسوف نستخدم عدداً من الطرق في الأمثلة التالية، ولكننا سنبدأ باستعراض محتويات قاعدة البيانات :

```
SELECT * FROM RETAILS;
```

النتيجة

Item	Price
256MB RAM	160
80GB HDD	300
15" Monitor	500
NIC 10/100	40
Speakers	30
Keyboard	25
16 Ports Switch	200
ADSL Router	400

الآن نريد إضافة مبلغ ١٠ جنيه على كل سعر من الأسعار ووضع النتيجة في عامود جديد، جرب معنا الجملة التالية :

```
SELECT ITEM, PRICE, PRICE + 10 FROM RETAILS;
```

ITEM	PRICE	PRICE+10
256MB RAM	160	170
80GB HDD	300	310
15" Monitor	500	510
NIC 10/100	40	50
Speakers	30	40
Keyboard	25	35
16 Ports Switch	200	210
ADSL Router	400	410

تعال نستخدم طريقة أخرى وهي اختيار اسم ولكن بمسمى جديد للحقل بعد الإضافة أو بعد جمع ١٠ جنيهات :

```
SELECT ITEM, PRICE, (PRICE + 10) NEWPRICE
        FROM RETAILS;
```

ITEM	PRICE	NEWPRICE
256MB RAM	160	170
80GB HDD	300	310
15" Monitor	500	510
NIC 10/100	40	50
Speakers	30	40
Keyboard	25	35
16 Ports Switch	200	210
ADSL Router	400	410

لقد قمنا بإعطاء اسم جديد للحقل داخل جملة QUERY وهذا ليس قاصرا على التعليمات والعلامات ولكن يمكن استخدامه بصفة عامة بحيث نضع الاسم الجديد بجوار الاسم القديم، لاحظ هذا المثال :

```
SELECT ITEM PRODUCT, PRICE, PRICE + 10 NEWPRICE
                        FROM RETAILS;
```

وضعنا كلمةPRODUCT بجوار الحقل ITEM وكلمة NEWPRICE بجوار حقل PRICE+10 فكانت النتيجة طباعة الكلمة الجديدة بدلا من اسم الحقل الموجود داخل قاعدة البيانات ، ولكن دون تغيير فعلي في اسم الحقل ، التغيير حدث أثناء الطباعة فقط.

PRODUCT	PRICE	NEWPRICE
256MB RAM	160	170
80GB HDD	300	310
15" Monitor	500	510
NIC 10/100	40	50
Speakers	30	40
Keyboard	25	35
16 Ports Switch	200	210
ADSL Router	400	410

- لاحظ أن Item تحولت إلى Product علما بأنه لم يحدث تغيير حقيقي داخل قاعدة البيانات ، ولكن يبقى الاسم في قاعدة البيانات كما هو Item ولكن الاسم الجديد هو مجرد اختصار Alias . ولا عجب إذا لم تقتنع بفائدة تذكر لهذه الاختصارات Aliases ، ولكنك سوف

ترحب كثيرا بذلك عندما تبدأ في إنشاء تقارير reports .

- بعض أنظمة قواعد البيانات تجيز استخدام أسماء مختصرة Alias ولكن باستخدام علامة = مثال:

```
SELETC ITEM = PRODUCT, PRICE, PRICE + 10 = NEWPRICE
                                    FROM RETAILS;
```

علامة الطرح (-) Minus

تستخدم علامة الطرح (–) لطرح بيانات حقل من حقل أو لطرح قيمة من حقـــل موجود . سوف نأخذ بعض الأمثلة عن عمليات الطرح المختلفة سوف نستعرض الجدول الأساسي أولا ثم نجري عملية الطرح :

```
SELECT ITEM,
       2 PRICE,
       3 SOLD,
       4 FROM SALES;
```

ITEM	PRICE	SOLD
256MB RAM	160	170
80GB HDD	300	310
15" Monitor	500	510
NIC 10/100	40	50
Speakers	30	40
Keyboard	25	35
16 Ports Switch	200	210
ADSL Router	400	410

الآن نريد أن نستخرج الفرق بين سعر البيع والسعر الأصلي :

```
SELECT ITEM, PRICE, sold,
       2 (SOLD – PRICE) PROFIT,
       3 FROM SALES;
```

ITEM	PRICE	SOLD	PROFIT
256MB RAM	160	170	10
80GB HDD	300	310	10
15" Monitor	500	510	10
NIC 10/100	40	50	10

Speakers	30	40	10
Keyboard	25	35	10
16 Ports Switch	200	210	10
ADSL Router	400	410	10

الحقل المعرف على أنه حروف Char لا يمكن استخدامه في العمليات الحسابية مثل الجمع والطرح وإلا فإننا سوف نحصل على خطأ عند تنفيذ الأمر .

علامة القسمة (/) Divide

نستخدم هذه العلامة في عمليات القسمة، وتستخدم للحصول علي خارج قسمة رقم على رقم آخر ويمكن أن نضع النتيجة أو ما يسمى خارج القسمة في حقل جديد . من هذا نفهم أننا يمكن أن نقسم حقل على حقل أو نقسم حقل على رقم محدد .

سوف نستخدم نفس الجدول Price Table من أجل التدريب فقط :

```
SELECT * FROM PRICES;
Item                 Price
-------------------- ----------
256MB RAM            160
80GB HDD             300
15" Monitor          500
NIC 10/100           40
Speakers             30
Keyboard             25
16 Ports Switch      200
ADSL Router          400
```

سنقسم السعر على رقم ٢ في المثال التالي :

```
SELECT ITEM, PRICE, (PRICE / 2) SALESPRICE
       2 FROM PRICES;
```

النتيجة

```
ITEM                 PRICE      SALESPRICE
-------------------- ---------- ----------
256MB RAM            160         80.0
80GB HDD             300        150.0
15" Monitor          500        250.0
NIC 10/100           40          20.0
Speakers             30          15.0
```

Keyboard	25	12.5
16 Ports Switch	200	100.0
ADSL Router	400	200.0

علامة الضرب (*) Multiply

مثل القسمة ، فان عملية الضرب تتم بين حقل ورقم صحيح أو بين حقلين ، وطبعا بشرط أن تكون الحقول معرفة على أنها رقمية number .

سوف نستخدم نفس الجدول Prices من أجل التدريب فقط :

سنضرب السعر في رقم ٢ في المثال التالي :

```
SELECT ITEM, PRICE, (PRICE * 2) SALESPRICE
                                2 FROM PRICES;
```

ITEM	PRICE	SALESPRICE
256MB RAM	160	320
80GB HDD	300	600
15" Monitor	500	1000
NIC 10/100	40	80
Speakers	30	60
Keyboard	25	50
16 Ports Switch	200	400
ADSL Router	400	800

لاحظ وجود عامود للكمية ، سنضرب السعر في الكمية ونحصل على عامود النتيجة باسم Total وذلك باستخدام الجملة التالية :

```
SELECT ITEM, PRICE, QTY, (PRICE * QTY) TOTAL
 FROM PRICES;
```

Item	Price	QTY	Total
256MB RAM	160	5	800
80GB HDD	300	6	1800
15" Monitor	500	7	3500
NIC 10/100	40	8	320
Speakers	30	9	270
Keyboard	25	4	100
16 Ports Switch	200	5	1000
ADSL Router	400	6	2400

علامة باقي القسمة (%) Modulo

أحيانا كثيرة نريد معرفة باقي تقسيم رقم معين على رقم آخر ، سوف نستخدم في هذه الحالة علامة (%) Modulo وتختصر هكذا mod ، وكعادتنا في الأمثلة السابقة سوف نعرض الجدول أولا ثم نقوم بكتابة التعليمات أو جمل Query ثم نرى النتيجة :

```
SELECT * DIVREM;
Amount      QTY
---------   ------
100         3
200         7
300         7
400         3
500         9
```

الآن نقوم بعملية القسمة مع الاحتفاظ بباقي القسمة :

```
SELECT AMOUNT, QTY, MOD(AMOUNT,QTY) REMAINS
                                FROM DIVREM;
Amount      QTY    remains
---------   ------ ------
100         3      1
200         7      4
300         7      6
400         3      1
500         9      5
```

ملاحظات هامة

- بالنسبة للعمليات الحسابية ، وكما هو معروف في مجال البرامج فتتم العمليات الحسابية وفقا للترتيب التالي : "عمليات الضرب ثم القسمة ثم الجمع ثم الطرح" وهذا الترتيب هام جدا بفرض أن هذه العمليات اجتمعت في معادلة واحدة أو بعض منها، فمثلا ٢+٣*٥ فان النتيجة سوف تكون قطعا ١٧ وليس ٢٥ لماذا؟ ارجع للترتيب وسوف تعرف لماذا ؟

- إذا أردنا أن نحافظ على الترتيب نستخدم الأقواس مثلا (٢+٣)*٥ = ٢٥، مثال آخر لنتعرف معا ماذا تكون النتيجة ٥*٨+١٢/٦ ترى ما هي النتيجة ؟ شخصيا أعتقد أن النتيجة سوف تكون ٤٢ !
- الملاحظة التالية هي أننا نستخدم النجمة * في الضرب والشرطة المائلة / في القسمة.

معاملات المقارنة Comparison Operators

هنا يمكن أن نتحدث طويلا (ولا حرج) فالإنسان من الميلاد والى المحطة الأخيرة من الرحلة يعيش في مقارنات دائمة ومستمرة .فمثلا نقول هذا السطح ناعم ولكن الآخر أكثر نعومة وقد نقول غير ناعم كما يمكن أن نقول خشن أو غير خشن كما أننا نستطيع القول أن هذا السطح أصغر من سطح الجيران مثلا ! وهذا الولد أطول أو أقصر كما يمكن أن نقول أنه ليس أطول أو ليس أقصر أو يساويه أو لا يساويه أو الجزء الأول مساوي للجـــزء الأول ولكنا لا نعلم هل باقي الشكل مساو أو غير مساو ويمكن أن نقول أن الـــشعب الـــصيني يشبه الشعب الكوري ولكن في السر حتى لا نتعرض لمسائلة من جانـــب شـــعبان تعـــديا المليارات من البشر ويتحدثان بلغة الذرة .

وهكذا عزيزي القارئ يمكننا أن نتحدث كثيرا في موضوع المقارنات بغير أن ينتهي الحديث أو يمكن أن ينتهي بنهاية نسبية .أما عن معاملات المقارنة التي تعنينا هنا فنقصد بها مقارنـــة تعبيرين أو أكثر والحصول علي نتيجة المقارنة وهي إما صحيحة **True** أو خاطئة **False**.

علامة (=) Equal Sign

من أكثر علامات المقارنة استخداما مع الشرط **where** ولنلاحظ هذا المثال :

```
select * from prices;
```

Item	Price	QTY	Total
256MB RAM	160	5	800
MB	300		
80GB HDD	300	6	1800
15" Monitor	500	7	3500

```
select * from prices where price = 300;
        Item      Price   QTY   Total
          MB        300
   80GB HDD        300      6    1800

select * from prices where item = 'mb';
        Item      Price   QTY   Total
          MB        300
```

عندما نقارن حقل برقم نضع الرقم كما هو أما إذا أردنا مقارنة حقل بحروف نضعها بين فاصلتين لماذا ؟ لكي نميز بين الحروف وبين اسم الحقل ، ففي المثال السابق إذا أهملنا الفاصلتين ووضعنا الحروف كما هي mb فسوف يقوم البرنامج بالبحث عن حقل اسمه mb لكي يضعه في المقارنة بينما الفاصلتين تخبر البرنامج أن ما بين الفاصلتين 'mb' قيمة يتم استخدامها في المقارنة .

ولتوضيح استخدام العلامات "أكبر من >" أو "أكبر من أو يساوي =< " سوف نستخدم الأمثلة التالية :

للحصول علي الأسعار التي تزيد عن ٣٥٠ فقط استخدم علامة المقارنة '>' كما يلي"

```
select * from prices
where price > 350;
    Item      Price      QTY      Total
 Monitor        520        3       1560
```

في هذا المثال تم اختيار السجلات في حالة واحدة وهي عندما يكون السعر أكبر من ٣٥٠

```
select * from prices
   where price >= 350;
    Item      Price      QTY      Total
      MB        350        6       2100
     HDD        350        3       1050
 Monitor        520        3       1560
```

في هذا المثال تم اختيار السجلات في حالة إذا كان السعر price أكبر من أو يساوي ٣٥٠ .

```
select * from prices;
```

Item	Price	QTY	Total
MB	350	6	2100
RAM	160	7	1120
KB	25	5	125
HDD	350	3	1050
Monitor	520	3	1560

علامة "أصغر من >" أو "أصغر من أو يساوي =>"

هذه العلامة عكس السابقة ، فالتعليمات التي سنكتبها سوف تقوم علـى تنفيـذ هـذه العلامات . للحصول على الأسعار التي تساوي ٣٥٠ من جدول Prices :

```
select * from prices where price < 350;
```

Item	Price	QTY	Total
RAM	160	7	1120
KB	25	5	125

وللحصول على الأسعار التي تساوي أو تقل عن ٣٥٠ استخدم الجملة التالية :

```
select * from prices where price <= 350;
```

Item	Price	QTY	Total
MB	350	6	2100
RAM	160	7	1120
KB	25	5	125
HDD	350	3	1050

أرجو عزيزي القارئ أن تدقق معي في الأمثلة ولا تمر عليها ، بل طابق ما نقول بتعليمات Query لكي تتأكد أن ما نقول يطابق الواقع

علامة غير مساو (=! أو < >)

إحدى هاتين العلامتين تعني يتم اختيار الصف إذا كانت قيمة الحقل غير مـساوية لقيمة أخرى ، فمثلا الحقل Price لا يساوي ٣٥٠ ، لاحظ المثال التالي :

سوف نجرب العلامتين الآن :

```
select * from prices where price < > 350;
```

Item	Price	QTY	Total
RAM	160	7	1120
KB	25	5	125
Monitor	520	3	1560

تم اختيار السجلات التي لا يساوي السعر فيها ٣٥٠

```
select * from prices where price != 350;
```

Item	Price	QTY	Total
RAM	160	7	1120
KB	25	5	125
Monitor	520	3	1560

تم اختيار السجلات التي لا يساوي السعر فيها ٣٥٠

البحث عن حقل بلا قيمة Null

تحدثنا من قبل عن الحقل الذي لا توجد به قيمة ، وهو طبعا مختلف عن الحقل الذي قيمته 0 أو الذي قيمته مسافة blank فالصفر والمسافة هي قيم ، ولكننا نتحدث عن الحقل الذي لم يتم إدخال أي قيمة فيه ويسمى Null value ولنأخذ مثالا :

```
select * from prices;
```

Item	Price	QTY	Total
256MB RAM	160	5	800
MB	300		
80GB HDD	300	6	1800
15" Monitor	500	7	3500

```
select * from prices where qty is null;
```

Item	Price	QTY	Total
MB	300		

معاملات الحروف Character Operators :

نوضح فيما يلي أهم المعاملات التي تستخدم لمعالجة الحروف مثل المعامل LIKE للبحث داخل قاعدة البيانات والمعامل (_) underscore للتعويض عن غياب حروف معينة .

عند استخدامك للحروف أو الكلمات المخزنة داخل قاعدة البيانات ، تأكد من طريقة كتابتها في التعليمات Query التي تكتبها ، فبعض أنظمة قواعد البيانات مثل ORACLE تفرق بين الحروف الكبيرة والحروف الصغيرة وتعطي للحرف a قيمة تختلف عن الحرف A وهذا ما يطلق عليه Case Sensitive بينما قواعد البيانات الأخرى مثل mysql تتعامل مع الحروف الصغيرة والكبيرة بقيمة واحدة أي أنها ليست Case Sensitive

علامة *LIKE*

تستخدم هذه العلامة للبحث داخل قاعدة البيانات عن كلمة وغالبا نستخدم مع الكلمة التي نبحث عنها علامة النسبة المئوية "%" كما في المثال التالي.

```
SELECT * FROM PRICES
WHERE ITEM LIKE '%RAM%' ;
```

Item	Prices	QTY	Total
SDRAM	180	5	900
RAM	160	7	1120
DDRAM	190	8	1520

لاحظ أنه بعد تنفيذ الجملة تم استعراض الصفوف التي تحتوي على كلمة RAM في أي مكان من الحقل وذلك لاستخدام علامة النسبة المئوية قبل وبعد الكلمة وهي تستخدم للتعويض عن غياب بعض الحروف واستخدامها . في هذا المثال قبل وبعد كلمة RAM معناه مهما سبقها من حروف أو تلاها من حروف .

فماذا لو أردت البحث عن كلمة RAM في بداية الحقل:

```
SELECT * FROM PRICES
WHERE ITEM LIKE 'R%' ;
```

Item	Prices	QTY	Total
RAM	160	7	1120

المعاملات المنطقية :

تستخدم SQL ثلاث معاملات منطقية، المعامل And والمعاملor والمعامل not وعـــادة يوضع معامل المقارنة بين الشروط التي نرغب في مقارنتها عند استخدام WHERE

المعامل AND:

يستخدم المعامل AND لعقد مقارنة بين قيمتين أو أكثر ويشترط أن تكون نتيجـــة المقارنة في كل الحالات صحيحة TRUE . إذا تحقق الشرط مع حالة من اثنين أو حـــالتين من ثلاثة فإن النتيجة تكون خطأ FALSE ولن ينفذ الأمر .تأمل معي المثال التالي :

مطلوب عرض السجلات إذا تحقق فيها ثلاث شروط الأول أن يكون salary يساوي أو أكبر من ٢٥٠٠ والثاني أن يكون الحقل marital أكبر من ١ والثالث أن يكون الحقـــل marital أصغر من ٤ أنظر المثال ولاحظ الفائدة من استخدام and.

```
select * from personal
where salary >= 2500  أكبر من أو يساوي
and marital > 1   وأكبر من
and marital < 4;   وأصغر من
```

fnam	lastname	gender	birth	Salary	marital
medo	jakob	m	1977-10-04	2500	2
mona	ahmed	f	1973-10-06	4500	2
ali	hasan	m	1980-11-11	4500	3
saad	saeed	m	1985-11-23	5000	3

المعامل OR:

يستخدم المعامل OR ليطلب من SQL تنفيذ الأمر إذا تحقـــق أحـــد الـــشرطين المذكورين قبله وبعده . إذا تحقق أحد الشرطين المحددين في الأمر تكون النتيجة صـــحيحة TRUE وينفذ الأمر أما إذا لم يتحقق أي منهما فتكون النتيجة FALSE ولـــن يتحقـــق الأمر . أنظر المثال التالي:

في هذا المثال مطلوب إظهار بيانات المدرسين إذا كانت أعمارهم تزيد عن ٣٥ أو تساوي ٢٧ عاما .

```
SELECT*
FROM teachers
WHERE age>35
OR Age=27;
```

Name	Age	gyear	Specialty
zaid	27	1997	English
Waleed	27	1998	Arabic

هنا تحقق شرط واحد فقط وهو المدرسين الذين تساوي أعمارهم ٢٧ ولـــذلك اعتـــبرت النتيجة صحيحة وتم تنفيذ الأمر .

معاملات Set :

نعرف أن قواعد البيانات العلائقية تتكون من مجموعات مرتبطـــة بـــبعض Sets بعلاقات معينة ، وفي هذه الحالة نحتاج للتعامل مع مجموعات معينة في وقت واحد ، كما لو أردنا البحث عن بيانات في جدولين في آن واحد ، ولنجرب هذا المثال :

نستعرض أولا بيانات الجدولين كما يلي :

```
select * from friends;
```

fname	lastname	gender
medo	jakoub	m
hoba	yakoub	f
nona	ahmed	f
ali	hasan	m
saad	saeed	m
gamal	gameel	m

```
select * from relatives;
```

fname	lastname	gender
hoba	Yakoub	f
nona	ahmed	f
Maged	Adel	m
Adel	Adam	m
Mohamed	Ahmed	m
Mahmoud	Reda	m

المعامل UNION ALL

يقوم بعرض محتوى الجدولين واحدا بعد الآخر ،لعرض محتويات الجدولين الموجودين في المثال السابق استخدم UNION ALL هكذا :

```
select * from friends
UNION All
Select * from relatives;
```

fname	lastname	gender
medo	jakoub	m
hoba	yakoub	f
nona	ahmed	f
ali	hasan	m
saad	saeed	m
gamal	gameel	m
hoba	Yakoub	f
nona	ahmed	f
Maged	Adel	m
Adel	Adam	m
Mohamed	Ahmed	m
Mahmoud	Reda	m

المعامل UNION

هذا المعامل يستخدم لعرض حقول جدولين ولكن مع حذف السجلات المكررة في كليهما ، أنظر المثال التالي ولاحظ الفرق بين الجدولين :

```
select * from friends
        UNION
        select * from relatives;
```

fname	lastname	gender
medo	jakoub	m
hoba	yakoub	f
nona	ahmed	f
ali	hasan	m
saad	saeed	m
gamal	gameel	m
Maged	Adel	m
Adel	Adam	m
Mohamed	Ahmed	m
Mahmoud	Reda	m

المعامل MINUS

يعرض السجلات الموجودة في أول جدول وليست موجودة في الجـــدول الثـــاني . للاستعلام في الجدولين عن الأسماء الموجودة في جدول friends وغير موجودة في الجدول الآخر relative انظر المثال التالي :

```
select * from friends
        MINUS
        select * from relatives;
 fname      lastname   gender
 nona       ahmed      f
 ali        hasan      m
 saad       saeed      m
 gamal      gameel     m
```

أما المثال التالي فهو يعرض الأسماء الموجودة في جدول relative وغير موجودة في الجدول friends:

```
select * from relative
        MINUS
        select * from friends;
 fname      lastname   gender
 Maged      Adel       m
 Adel       Adam       m
 Mohamed    Ahmed      m
 Mahmoud    Reda       m
```

المعامل Intersect

يعرض السجلات الموجودة في المجموعتين "الجدولين" فقط . لعرض الأسماء الموجودة في كلا الجدولين السابقين فقط :

```
select * from friends
        INTERSECT
         select * from relatives;
 fname      lastname   gender
 hoba       yakoub     F
 nona       ahmed      F
```

معاملات أخري

المعامل in

يستخدم هذا المعامل للاختيار ضمن قيم معينة فمثلا إذا أردنا اختيار السجلات التي يساوي حقل SALARY فيها ٢٥٠٠ أو يساوي ٥٠٠٠ استخدم المثال التالي :

```
use employee;
select * from personal;
```

Fname	lastname	gender	birth	salary	marital
medo	jakoub	m	4-Oct-77	2500	2
hoba	yakoub	f	2-Oct-80	3200	1
nona	ahmed	f	10-Jun-73	4500	2
ali	hasan	m	11-Nov-80	4500	3
saad	saeed	m	23-Nov-85	5000	3
gamal	gameel	m	5-Jun-67	6000	4

```
select * from personal
   where salary in(2500, 5000);
```

Fname	lastname	gender	birth	salary	marital
medo	jakoub	m	4-Oct-77	2500	2
saad	saeed	m	23-Nov-85	5000	3

وهي تعادل الجملة التالية

```
where salary = 2500 OR salary =5000
```

المعامل Between

عندما نختار إظهار الحقول التي تحتوي علي مرتبات بين ٢٥٠٠ و ٥٠٠٠ فهذا يعني أننا نكتب الجملة باستخدام علامة AND هكذا :

```
select * from personal
    where salary >= 2500
    and salary <= 5000;
```

للحصول على نفس النتيجة باستخدام المعامل BETWEEN استخدم الجملة هكذا :

```
select * from personal
   where salary between 2500 and 5000;
```

Fname	lastname	gender	birth	salary	marital
medo	jakoub	m	4-Oct-77	2500	2
hoba	yakoub	f	2-Oct-80	3200	1
nona	ahmed	f	10-Jun-73	4500	2
ali	hasan	m	11-Nov-80	4500	3
saad	saeed	m	23-Nov-85	5000	3

لاحظ أن استخدام العلامة Between تشمل القيمة الأولى ((تعني أكبر من أو يساوي)) والقيمة الأخيرة أيضا ((تعني أصغر من أو يساوي)) كما في المثال السابق فان الحقول التي تتضمن ٢٥٠٠ و ٥٠٠٠ تم اختيارها أيضا. أما العلامة in فهي تعني (يساوي القيم المذكورة فقط) .