الفصل الحادي عشر
الربط بين جدولين أو أكثر

نقصد بربط الجداول هنا دمج بيانات جدولين أو أكثر ويتم دمج الجداول بناء على تساوي قيم حقلين أو أكثر موجودين في كلا الجدولين كما يمكن أن يتم بناء على اختلاف القيم الموجودة في حقلين أو أكثر من حقول الجداول .

بالانتهاء من هذا الفصل ستكتسب المعارف وتتدرب على المهارات التى تجعلك قادرا على:

- *ربط الجداول باستخدام جملة* **SELECT** *بسيطة*
- *ربط أو دمج الجداول بناء علي بيانات متساوية*
- *ربط الجداول باستخدام الحقول الغير متساوية القيمة*
- *استخدام* **JOIN** *لربط الجداول*

إن الجداول **Tables** عبارة عن وسيلة لتخزين جزء من المعلومات حيث لا يمكن تخزين كل معلومات قاعدة البيانات في جدول واحد ، فهذا أمر ليس عمليا ، ولكن ضع بعض المعلومات في جدول ، وعندما تحتاج الى معلومات أخرى ضعها في جدول آخر وهكذا ، وعندما تحتاج الى معلومة من جدول لتضعها بجوار معلومة من جدول آخر ، هنا فقط يمكن أن تختار بيانات من جدولين أو ثلاثة جداول ، تختار بعض البيانات ، وهنا الأمر يبدو منطقيا بحيث تحصل على نتيجة معقولة .

نفرض أن قائمة الكميات موجودة في مخزن **STOCK**، فليس هناك حاجة لوضع أسعار الأصناف في المخزن ولكن الطبيعي أن يكون كشف الأسعار في متناول ادارة المبيعات ، وعندما يطلب عميل بعض الكميات يقوم مسئول المبيعات باعداد كشف بيع **Order** ، وهكذا يمكن عمل ٣ جداول . جدول للمخزن به رقم الصنف ووصف الصنف والكمية ، بينما هناك قائمة الأسعار وبها رقم الصنف والسعر ، وعند كل طلبية ينفذها مسئول المبيعات فانه سيقوم

بعمل طلب بيع يحتوي على رقم الصنف ووصف الصنف والكمية والسعر ، هنا عليك أن تقرأ وصف الصنف من جدول المخزن وأن تقرأ السعر من جدول الأسعار وتضع الكمية التي سيتم بيعها وتحسب جملة السعر ، أليس هذا منطقيا ؟

ربط الجداول باستخدام جملة SELECT بسيطة :

يمكنك استخدام جملة **SELECT** لاستخراج بيانات موجودة في جدولين بدون استخدام أي شرط . وهذه الطريقة عقيمة ولا ننصح باستخدامها ولكننا نوردها هنا للتوضيح فقط ولبيان أنك يمكنك استخدام جملة **SELECT** لربط بيانات موجودة في جدولين وسنتعرف بعد قليل طرق أجدى في ربط بيانات الجداول وهي التي ستستخدمها في حياتك العملية .

نفرض أن عندنا جدولين هكذا :

SELECT * FROM EMPLOYEE

EMPNO	NAME
1	MOHAMMED
2	ALI
3	ABDULLAH
4	SAMY

SELECT * FROM SALARY

EMPNO	SALARY
1	3000
2	4000
3	5000
4	2500

وباستخدام جملة **SELECT** لدمج بيانات الجدولين سوف تحصل علي النتيجة التالية :

SELECT * FROM EMPLOYEE , SALARY

EMPNO	NAME	EMPNO	SALARY
1	MOHAMMED	1	3000
1	MOHAMMED	1	3000
1	MOHAMMED	1	3000
1	MOHAMMED	1	3000
2	ALI	2	4000
2	ALI	2	4000
2	ALI	2	4000
2	ALI	2	4000
3	ABDULLAH	3	5000
3	ABDULLAH	3	5000
3	ABDULLAH	3	5000
3	ABDULLAH	3	5000
4	SAMY	4	2500
4	SAMY	4	2500
4	SAMY	4	2500
4	SAMY	4	2500

دقق النظر في هذه النتيجة ... ماذا تلاحظ ؟

ستلاحظ أن كل صف (سجل) في جدول **SALARY** تم دمجه مع الصف الأول من جدول **EMPLOYEE** ولكن ما نوع هذا الدمج ؟ هل هو دمج بناء علي تساوي بيانات حقول معينة أو بناء علي عدم تساويهما أو باستخدام **JOIN** (وهذا ما ستعرفه بعد قليل) . هذا النوع من الدمج يسمي **CROSS JOIN** .

وكما قلنا في البداية لا ننصح باستخدام هذا النوع من الدمج لأنه غير مفيد كما رأيت . ولكننا أوردناه هنا لتوضيح فكرة دمج البيانات . الدمج العملي والمفيد ستعرفه فيما يلي وهو إما الدمج بناء علي تساوي أو عدم تساوي بيانات حقول معينة أو الدمج الخارجي أو ما يسمى **OUTER JOIN** .

راجع النتيجة السابقة ستلاحظ أيضا أن الصفوف الناتجة عددها ١٦ صفا وهي عبارة عن ٤ صفوف من الجدول الأول X ٤ صفوف من الجدول الثاني = ١٦ .

هل لك أن تتصور إذا كان عدد سجلات (صفوف) كل جدول ٢٠٠ أن النتيجة ستكون ٢٠٠ X ٢٠٠ = ٤٠٠٠٠ صف !!!

ألم أقل لك أن هذه الطريقة غير عملية ولذلك سنشرح طرقا أخرى لدمج الجداول أكثر فائدة.

ربط أو دمج الجداول بناء علي بيانات متساوية :

المقصود بدمج أو ربط الجداول بناء علي بيانات متساوية ويسمى **EQUI-JOIN** معناه دمج بيانات سبق تخزينها في جدول مع بيانات جدول آخر بمعني آخر مقارنة قيم موجودة بحقل معين مع نظيرتها في جدول آخر .

وللحصول علي نتائج أكثر دقة يمكنك إضافة شروط إضافية للحصول علي النتائج التي ترغبها مثلا باستخدام **WHERE** .

هنا سوف نعرض محتويات الجدول الأول **stock** على حدة :

ID	Part_No	Description	Balance
1	MB01	Mother Board Giga	120
2	MB02	Mother Board Accorp	300
3	MO01	LG 15" Monitor	200
4	MO02	LG 17" Monitor	80
5	MO03	Samsong 15"	100
6	MO04	Samsong 17"	85
7	P001	1.7GHz	40
8	P002	2GHz	65
9	P003	2.4GHz	110
10	P004	2.8GHz	220
11	HD01	HDD 40GB	75
12	HD02	HDD 80GB	175
13	HD03	HDD 115GB	275

والجدول الثاني price

ID	Part_No	Price_List
1	MB01	300
2	MB02	280
3	MO01	500
4	MO02	600
5	MO03	560
6	MO04	700
7	P001	350
8	P002	400
9	P003	500
10	P004	600
11	HD01	350
12	HD02	420
13	HD03	710

والجدول الثالث orders وهو يمثل طلبية معينة :

ID	Part_No	QTY
1	MB01	20
2	MO01	10
3	MO04	5
4	P002	5
5	P004	10
6	HD01	4
7	HD03	8

عندما نستخدم أكثر من جدول table في جملة query واحدة يجب أن نقوم بتحديد الحقل الذي نتعامل معه لكي يعرف نظام قواعد البيانات أن هذا الحقل من جدول معين فنقول مثلا stock.part_no هنا وضعنا اسم الجدول وبعده نقطة ثم نكتب اسم الحقل هذا عندما يكون اسم الحقل مكررا في الجداول التي نستخدمها ، أما إذا كان اسم الحقل غير مكرر فيمكن إغفال اسم الجدول وان كان وضعه في حالة استخدام أكثر من جدول (زيادة تأكيد).
فمثلا هذه الجملة تحتوي على تعريف للجدول قبل اسم كل حقل :

MySQL

```
select stock.Part_No,
stock.description, price.prices, orders.qty
from stock, prices, orders ;
where stock.Part_no = orders.Part_no
and orders.Part_no = Prices.part_no;
```

Part_No	Description	Price	qty
MB01	Mother Board Giga	300	20
MO01	LG 15" Monitor	500	10
MO04	Samsong 17"	700	5
P002	2GHz	400	5
P004	2.8GHz	600	10
HD01	HDD 40GB	350	4
HD03	HDD 115GB	710	8

في هذا المثال قمنا بعرض رقم الصنف part_no وأخذنا الوصف Description من جدول stock ثم أخذنا السعر من جدول prices ثم أخذنا الكمية من جدول orders وحصلنا على النتيجة السابقة .ولكي نحصل علي صف واحد لكل بند ونضمن عدم تكرار الصف استخدمنا الشرط WHERE .

الشرط المحدد في هذا المثال هو تساوي رقم الصنف Part_No في كل من جدول stock وجدول order بالإضافة إلي تساوي رقم الصنف في جدول orders مع رقم الصنف في جدول prices .

هنا حذفنا اسم الجدول من الحقول التي لا يتكرر اسم الحقل فيها ، ونحصل على نفس نتيجة الجملة السابقة :

MySQL

```
select stock.Part_No,
description, price, qty
from stock, prices, orders
where stock.Part_no = orders.Part_no
and orders.Part_no = Price.part_no;
```

ومعناها طباعة قائمة تحتوي على رقم الصنف والوصف والسعر والكمية باستخدام ثلاثة جداول سوف تحصل علي نفس النتيجة السابقة :

بينما الجملة التالية لن تعمل وسوف تعطينا رسالة error وذلك لأن اسم الحقل part_no متكرر في الجداول المستخدمة وهي stock, price, orders :

Microsoft sql:

```
select Part_No,
description, price, qty
from stock, prices, orders
where stock.Part_no = orders.Part_no
and orders.Part_no = Price.part_no;
```

وسنحصل على رسالة تفيد أن النظام لا يستطيع التعامل مع حقل part_no لانه لا يعرف أي جدول يستخدم معه حقل part_no والرسالة كالتالي :

```
Server: Msg 209, Level 16, State 1, Line 1
Ambiguous column name 'Part_No'.
```

انظر المثال التالي :

MySQL

```
select stock.Part_No,
stock.description, prices.price, orders.qty,
prices.price * orders.qty total
from stock, price, orders
where stock.Part_no = orders.Part_no
and orders.Part_no = Prices.part_no;
```

في هذا المثال قمنا باضافة عملية ضرب إلى المثال السابق حيث قمنا بضرب الكمية في السعر وحصلنا على حقل المجموع total . لاحظ أن اسم الجدول متبوعا بنقطة سابقة علي اسم الحقل ليدل على أن هذا الحقل موجود في هذا الجدول وأن ناتج عملية الضرب تم وضعه في حقل مؤقت وهو total وحصلنا على النتيجة التالية :

Part_No	Description	Price	qty	total
	Mother Board			
MB01	Giga	300	20	6000
MO01	LG 15" Monitor	500	10	5000
MO04	Samsong 17"	700	5	3500
P002	2GHz	400	5	2000
P004	2.8GHz	600	10	6000
HD01	HDD 40GB	350	4	1400
HD03	HDD 115GB	710	8	5680

ولأن أسماء الجداول كبيرة فيمكن الاشارة اليها بحرف من الحروف فمثلا جدول stock نشير اليه بحرف s وجدول price نشير اليه بحرف p وجدول orders نشير اليه بحرف o هذه الاشارة تكون بعد فقرة from فنقول مثلا from stock s, orders o, price p هنا يفهم query أن أي استخدام لحرف معين مذكور في فقرة from هو اشارة للجدول نفسه كما في المثال التالي :

MySQL

```
select s.Part_No,
description, price, qty,
p.price * o.qty total
from stock s, price p, orders o
where s.Part_no = o.Part_no
and o.Part_no = P.part_no;
```

Part_No	Description	Price	qty	total
MB01	Mother Board Giga	300	20	6000
MO01	LG 15" Monitor	500	10	5000
MO04	Samsong 17"	700	5	3500
P002	2GHz	400	5	2000
P004	2.8GHz	600	10	6000
HD01	HDD 40GB	350	4	1400
HD03	HDD 115GB	710	8	5680

ربط الجداول باستخدام الحقول الغير متساوية القيمة

وفي هذه الطريقة يتم التعامل مع الجداول على أساس عدم التساوي أو بشرط عدم التساوي ، وفي المثال التالي تجد أن شرط عرض محتويات الجدولين هي عدم تساوي حقل مع مثيله في الجدول الأول (رقم الصنف في الجدول الأول أقل من رقم الصنف في الجدول الثاني) ، انظر المثال التالي :

MySQL

```
select s.description, s.part_no,
o.part_no, o.qty
from stock s, orders o
where s.part_no < o.part_no;
```

استخدمنا علامة أصغر من بدلا من علامة يساوي في الأمثلة السابقة وبالتالي حصلنا على النتيجة التالية . لاحظ أن أرقام الأصناف الغير متساوية أكثر من تلك المتساوية :

Description	Part_no	Part_no	QTY
HDD 40GB	HD01	MB01	20
HDD 80GB	HD02	MB01	20
HDD 115GB	HD03	MB01	10
Mother Board Giga	MB01	MO01	10
Mother Board Accorp	MB02	MO01	10
HDD 40GB	HD01	MO01	10
HDD 80GB	HD02	MO01	10
HDD 115GB	HD03	MO01	5
Mother Board Giga	MB01	MO04	5
Mother Board Accorp	MB02	MO04	5
LG 15" Monitor	MO01	MO04	5
LG 17" Monitor	MO02	MO04	5
Samsong 15"	MO03	MO04	5
HDD 40GB	HD01	MO04	5
HDD 80GB	HD02	MO04	5
HDD 115GB	HD03	MO04	5
Mother Board Giga	MB01	P002	5
Mother Board Accorp	MB02	P002	5

LG 15" Monitor	MO01	P002	5
LG 17" Monitor	MO02	P002	5
Samsong 15"	MO03	P002	5
Samsong 17"	MO04	P002	5
1.7GHz	P001	P002	5
HDD 40GB	HD01	P002	5
HDD 80GB	HD02	P002	5
HDD 115GB	HD03	P002	10
Mother Board Giga	MB01	P004	10
Mother Board Accorp	MB02	P004	10
LG 15" Monitor	MO01	P004	10
LG 17" Monitor	MO02	P004	10
Samsong 15"	MO03	P004	10
Samsong 17"	MO04	P004	10
1.7GHz	P001	P004	10
2GHz	P002	P004	10
2.4GHz	P003	P004	10
HDD 40GB	HD01	P004	10
HDD 80GB	HD02	P004	10
HDD 115GB	HD03	P004	8
HDD 40GB	HD01	HD03	8
HDD 80GB	HD02	HD03	8

في هذه الطريقة تتم مقارنة رقم الصنف الأول Part_No مع أرقام الصنف في الجدول الثاني وكلما تحقق الشرط وهو عدم التساوي يتم عرض الصف وهكذا لنجد أن رقم الصنف يتكرر عدة مرات

استخدام JOIN لربط الجداول :

تستخدم JOIN لدمج بيانات جدولين وذلك طبقا لمحتويات الحقول المحددة لإظهارها وكيفية استخدام JOIN داخل جملة SELECT .

هناك ثلاث حالات لاستخدام JOIN لربط الجداول هي INNER JOIN و LEFT OUTER JOIN و RIGHT OUTER JOIN .ويوضح الجدول التالي السجلات التي نحصل عليها في كل حالة من هذه الحالات .

نوع الربط	السجلات الناتجة من الجدول الأيسر	السجلات الناتجة من الجدول الأيمن
INNER JOIN	السجلات المتعلقة بالجدول الأيمن فقط	السجلات المتعلقة بالجدول الأيسر فقط
LEFT OUTER JOI N	جميع السجلات	السجلات المتعلقة بالجدول الأيسر فقط
RIGHT OUTER JOIN	السجلات المتعلقة بالجدول الأيمن فقط	جميع السجلات

ويجب الانتباه إلي أن الجدول الأيسر هو الموجود علي يسار كلمة JOIN والجدول الأيمن هو الجدول الموجود علي يمين JOIN . الجدول السابق يعني أنك إذا استخدمت الأمر JOIN مع النوع INNER وتكتب داخل الأمر هكذا :

INNER JOIN:

للربط بين جدولين بينهما علاقة ، فإن الناتج سيكون في هذه الحالة عبارة عن سجلات مستخلصة من الجدول الأول ولها علاقة بالجدول الثاني بالإضافة إلي سجلات مستخلصة من الجدول الثاني ولها علاقة بالجدول الأول . أما إذا استخدمنا النوع LEFT OUTER JOIN فإن الناتج سيكون جميع السجلات الموجودة داخل الجدول الأول بالإضافة إلي السجلات الموجودة داخل الجدول الثاني التي لها علاقة بالجدول الأول . وسيحدث العكس في حالة استخدام النوع الثالث RIGHT OUTER JOIN وذلك طبقا للجدول الموضح سابقا .

في المثال التالي ستظهر فقط من جدول parts السجلات التي لها علاقة مع جدول orders:

```
MYSQL > SELECT P.PARTNUM , P.PRICE,
        > O.NAME , O.PARTNUM
        > FROM PART P
        > INNER JOIN ORDERS O ON O.PARTNO =16;
```

PARTNUM	PRICE	NAME	PARTNUM
16	454	HARD DISK	16
24	350	HARD DISK	16
54	1200	HARD DISK	16
33	530	HARD DISK	16
42	24	HARD DISK	16

5 rows in set

احتدم النقاش بيني وبين ابنتي حول جدوى استخدام Join Outer بصرف النظر عن left أو right فقد قلت لها أن هذه الخاصية لا فائدة منها ويمكن تنفيذها بطرق أخرى ، قالت أبدا ، وسأضرب لك مثالا ، قلت تفضلي . قالت هب أن لديك جدول المخزن التالي :

```
select * from stock ;
```

Part_no	Description	Balance
MB01	Mother Board Giga Mother Board	120
MB02	Accorp	300
MO02	LG 17" Monitor	80
MO03	Samsong 15"	100
MO04	Samsong 17"	85
P001	1.7GHz	40
P003	2.4GHz	110
HD01	HDD 40GB	75
HD02	HDD 80GB	175
HD03	HDD 115GB	275

ثم جدول الأسعار التالي :

```
select * from prices ;
```

Part_no	Price
MB01	300
MB02	280
MO03	560
MO04	700
P002	400
P003	500
P004	600
HD02	420

وتريد طباعة الأصناف كلها وبجانب كل صنف السعر ، لو استخدمت أي طريقة أخرى فانك ستطبع فقط الأصناف من جدول المخزن التي لها سعر في جدول الأسعار ولكنك لن تستطيع طباعة كل الأصناف سواء التي لها سعر أو التي ليس لها سعر بدون **Join Outer** ثم قالت حاول أن تفعل ذلك بدون هذه الطريقة ، قلت سأحاول ، ولكن حتى الآن لم أجد غير هذه الطريقة ، وطبعا لم أتحدث معها في الموضوع مرة أخرى حتى لا تكسب التحدي .

```
select * from stock s
left outer join
prices p on p.part_no = s.part_no;
```

Part_no	Description	Balance	Part_no	Price
MB01	Mother Board Giga	120	MB01	300
MB02	Mother Board Accorp	300	MB02	280
MO02	LG 17" Monitor	80	NULL	NULL
MO03	Samsong 15"	100	MO03	560
MO04	Samsong 17"	85	MO04	700
P001	1.7GHz	40	NULL	NULL
P003	2.4GHz	110	P003	500
HD01	HDD 40GB	75	NULL	NULL
HD02	HDD 80GB	175	HD02	420
HD03	HDD 115GB	275	NULL	NULL

ونتيجة لتنفيذ هذه **Query** فاننا حصلنا بالفعل على جميع الأصناف بالمخزن وأمام كل صنف السعر ان وجد وان لم يوجد تتم طباعة **NULL** أمام كل صنف ليس له سعر من جدول الأسعار . فهل ابنتي محقة في تحديها ؟ أرجوك عزيزي القارئ لا تخبرها !

تختلف طريقة دمج الجداول **Join** من نظام إلى آخر ، والاختلافات بين الأنظمة غير كبيرة في طريقة كتابة الشكل العام **syntax** ، وعندما تكتب أحد الجمل **query** تأكد من التعليمات .

والسؤال هنا هل يتم دمج حقيقي للجداول كنتيجة لجملة **query** ؟

في الحقيقة لا يتم أي دمج للجداول وانشاء جداول جديدة نتيجة لاستخدام **query** ، ولكن الهدف أن تكون حقول أكثر من جدول أمامك تختار منها ما تريد لتحصل على النيجة المرجوة دون الحاجة الى انشاء مزيد من الجداول أو تعديلها . أما ما يحدث فان **query** تقوم بانشاء جداول تخيلية **Virtual Tables** تنتهي بانتهاء تنفيذ **query** .