



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات 2

رقم الجلسة (2)

عنوان الجلسة

ربط الجداول والاستعلامات الفرعية





جدول المحتويات

Contents

رقم الصفحة	العنوان
3	ربط الجداول
5	الربط البسيط أو الديكارتي
6	الربط بالتساوي Equi Join
7	الربط اللا متساوي Non-Equi Join
8	الربط الخارجي Outer Join
10	الاستعلامات الفرعية Sub Queries
11	الاستعلامات الفرعية مع EXISTS

الغاية من الجلسة:

تعريف الطلاب باستخدام ربط الجداول و تقنيات الربط المختلفة من أجل توليد تقارير قابلة للاستخدام بالإضافة إلى استخدام الاستعلامات الفرعية و شروط استخدامها

ربط الجداول Joining Tables

فكرة الربط بين الجداول:

هي إنشاء استعلامات من أكثر من جدول ، حيث يمكن لاستعلام SQL التعامل مع عدد كبير من الجداول, شرط أن تتشارك بحقول معينة ,حيث يتم دمج صفوف من جداول متعددة عن طريق تحديد معايير المطابقة ، عادة ما تستند إلى المفتاح الأساسي - المفاتيح الغريبة

مثال بفرض لدينا جدول المديرية أو الأقسام في الشركة و الذي يحوي رقم القسم و اسمه و عنوانهالخ و جدول الموظفين و الذي يحوي بيانات الموظفين الشخصية و رقم القسم الذي ينتمي إليه الموظف و نريد استعلام يعيد أسماء الموظفين و أسماء الأقسام التي ينتمون إليها كما هو مبين بالشكل

جدول الموظفين			جدول الأقسام		
EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
100	King	90	10	Administration	1700
101	Kochhar	90	20	Marketing	1800
202	Fay	20	50	Shipping	1500
205	Higgins	110	60	IT	1400
206	Gietz	110	80	Sales	2500
			90	Executive	1700
			110	Accounting	1700
			190	Contracting	1700

EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
102	90	Executive
205	110	Accounting
206	110	Accounting

أنواع الربط :

- الربط البسيط أو الجداء الديكارتي *Cartesian product*
- الربط بالتساوي *Equi join*
- الربط بعدم المساواة *Non-Equi join*
- الربط الداخلي
- الربط الخارجي *Outer join*
- الربط لجدول مع نفسه *self-join*

للتنفيذ سيكون لدينا قاعدة بيانات صغيرة تضم ثلاث جداول الأقسام و الموظفين و أولاد الموظفين وفق التالي :

Department	
DeptID	DeptName
Actg	Accounting
Admn	Administration
Fin	Finance
Mktg	Marketing

Employee				
EmpID	FirstName	LastName	Gender	DeptID
1	عمر	السهلي	Male	Actg
2	طارق	عواد	Male	Mktg
3	سامية	سمان	Female	Mktg
4	حنان	آغا	Female	NULL

Children				
ChildID	ChildName	Gender	Birthdate	EmpID
1	سحر	Female	11/02/1990	1
2	طوني	Male	19/02/1990	2
3	رهام	Female	30/08/1992	1
4	لميس	Female	19/09/1980	3
5	وسيم	Male	04/03/1999	4
6	حسن	Male	02/07/2000	1
7	محمد	Male	29/05/1991	3
8	عبير	Female	09/01/2001	2

الربط البسيط أو الجداء الديكارتي Cartesian Product

تتم عملية الربط بمقابلة كل سجل من الجدول الأول مع جميع سجلات الجدول الثاني. وهذا ما يدعى بالجداء الديكارتي أو *Cartesian product*. وفي هذه الحالة يكون عدد السجلات الناتجة عن الاستعلام هو عدد سجلات الجدول الأول \times عدد سجلات الجدول الثاني.

مثال: نتيجة الربط البسيط بين الجدولين: Employee و Department هي:

```
select * from Employee ,Department
```

Cartesian Product

EmpID	FirstName	LastName	gender	DeptID	DeptID	DeptName
1	عمر	السهلي	male	actg	actg	Accounting
2	طارق	العواد	male	mktg	actg	Accounting
3	سامية	سمان	female	mktg	actg	Accounting
4	حنان	آغا	female	NULL	actg	Accounting
1	عمر	السهلي	male	actg	adm	Administration
2	طارق	العواد	male	mktg	adm	Administration
3	سامية	سمان	female	mktg	adm	Administration
4	حنان	آغا	female	NULL	adm	Administration
1	عمر	السهلي	male	actg	fin	Finance
2	طارق	العواد	male	mktg	fin	Finance
3	سامية	سمان	female	mktg	fin	Finance
4	حنان	آغا	female	NULL	fin	Finance
1	عمر	السهلي	male	actg	Mktg	Marketing
2	طارق	العواد	male	mktg	Mktg	Marketing

يمكن التعبير عن صيغة الربط بالصيغة التالية:

```
SELECT Table1.Column1, Table2.Column2 FROM Table1, Table2;
```

يسمى استعلام الربط البسيط أيضاً بالربط المتصالب أو Cross join. ويمكن التعبير عن نفس صيغة الربط السابقة، بالصيغة:

```
SELECT Table1.Column1, Table2.Column2  
FROM Table1 CROSS JOIN Table2;
```

الربط بالتساوي Equi Join

يُعرّف الربط بالتساوي على أنه الربط البسيط بين سجلات جدول أول، وسجلات جدول ثان اعتماداً على مساواة بين قيمة حقل في سجل من الجدول الأول (عادة المفتاح الخارجي) وقيمة حقل في سجل من الجدول الثاني (عادة المفتاح الرئيسي). يُعبّر عن الربط بالتساوي بالصيغة:

```
SELECT Table1.Column1, Table1.Column2, Table2.Column3
FROM Table1, Table2
WHERE Table1.Column1 = Table2.Column2;
```

عموماً، لا تستخدم عملية الربط بالضرورة نفس الحقول التي يجب أن يعيدها الاستعلام.

مثال:

لنفرض أننا نبحث عن أسماء الموظفين الذين يعملون في قسم المحاسبة.

عندها سيكون الاستعلام على الشكل التالي:

```
SELECT DeptName, FirstName, LastName
FROM Employee, Department
WHERE Employee.DeptID = Department.DeptID
AND DeptName = 'Accounting'
```

مثال 1: عرض أسماء الموظفين الذين لديهم ولد اسمه وسيم:

```
SELECT FirstName, LastName
FROM Employee, Children
WHERE Employee.EmpID = Children.EmpID
AND ChildName = 'وسيم'
```

مثال 2: أسماء الأبناء الذكور للموظفين العاملين في قسم التسويق.

حيث نلاحظ هنا أننا نحتاج بيانات من جدول الأولاد و الشرط من جدول الأقسام و بسبب عدم وجود رابط مباشر بين الجدولين سنلجأ للربط مع جدول ثالث هو جدول الموظفين الذي يرتبط بكل من الجدولين السابقين و يصبح الاستعلام كالاتي :

```
SELECT ChildName
FROM Department, Employee, Children
WHERE Department.DeptID = Employee.DeptID
AND Employee.EmpID = Children.EmpID
AND Children.Gender = 'Male'
AND DeptName = 'Marketing'
```

مثال 3: أسماء الموظفين وأعداد أولاد الموظفين (في جميع الأقسام) (جميع الموظفين)

حيث سنحتاج هنا إلى الربط بين جدولي الموظفين و الأولاد و استخدام تابع التجميع count لحساب عدد الأولاد لكل موظف

```
SELECT FirstName, LastName, Count(ChildID) As ChildrenCount
FROM Employee, Children
WHERE Employee.EmpID = Children.EmpID
GROUP BY FirstName, LastName
```

ملاحظة :

عند إضافة العلاقة Primary key – Foreign key، يتم استعمال اسم الجدول ملحقاً بنقطة '، لمنع الالتباس عند تشابه أسماء الحقول الموجودة في عدة جداول.

كما يجب استخدام اسم الجدول ملحقاً بنقطة ضمن التعبير Select أو Where أيضاً عند إمكانية حدوث التباس بالحقول. انظر المثال 2 في الجدول السابق

الربط اللامتساوي Non-Equi Join

يعتمد الربط بالمساواة على استخدام المساواة في شرط التعبير WHERE ولكن هذا لا يعني أننا لا نستطيع استخدام عمليات المقارنة الأخرى (أكبر، أصغر، وغيرها) كما في الصيغة التالية:

```
SELECT Table1.Column1, Table2.Column2
FROM Table1, Table2
WHERE Table1.Column1 < Table2.Column2;
```

مثال:

تحديد الاستعلام الذي يعطي أسماء الأقسام وأسماء الموظفين الذين لا يعملون فيها.

عندها سيكون الاستعلام على الشكل التالي:

```
SELECT DeptName, FirstName, LastName
FROM Employee, Department
WHERE Employee.DeptID <> Department.DeptID
```

الربط الداخلي Inner Join

يعطي الربط الداخلي نفس النتيجة التي يعطيها الربط بالمتساوي، الفرق فقط بالصيغة. إذ لا تزودنا جميع أنواع أنظمة إدارة قواعد المعطيات بالربط الداخلي. فنسخ Oracle ما قبل 9 لا تدعم الربط الداخلي.

بالنسبة للصيغة، يوجد اختلافان:

- يفصل بين أسماء الجداول الكلمات Inner Join بدلاً من الفواصل '،'.
- يتغير موضع تحديد العلاقة بين الجداول من Where إلى On، تاركين بذلك التعبير Where للشروط التقليدية.

الأمثلة الواردة في الجدول التالي هي نفسها الواردة في فقرة الربط بالمتساوي، وبالتالي يمكنك المقارنة.

مثال 1: المطلوب عرض أسماء الموظفين الذين لديهم ولد اسمه وسيم باستخدام inner join.

```
SELECT FirstName, LastName
FROM Employee Inner Join Children
ON Employee.EmpID = Children.EmpID
WHERE ChildName = 'وسيم';
```

مثال 2: أسماء الأبناء الذكور للموظفين العاملين في قسم التسويق.

```
SELECT ChildName
FROM Department
Inner Join Employee
ON Department.DeptID = Employee.DeptID
Inner Join Children
ON Employee.EmpID = Children.EmpID
WHERE Children.Gender = 'Male'
AND DeptName = 'Marketing'
```

ملاحظة:

رأينا فيما سبق كيفية تغيير اسم حقل أو عمود ضمن التعبير **Select** في أي استعلام، وذلك عبر ما يسمى **Alias**. يمكننا أيضاً تغيير أسماء الجداول ضمن **SQL**، بهدف تقصير طول الاستعلام أو تسهيل قراءته.

لاستعمال **alias** ضمن التعبير **From**، أتبع الاسم الحقيقي للجدول بفراغ ثم الاسم الجديد الذي ترغب بالتعامل معه. ويمكنك إضافة **As** بين الاسم الحقيقي للجدول والاسم الجديد له.

وعند تغيير تسمية جدول، لا يمكن استخدام الاسم الحقيقي في بقية الاستعلام. يبين الجدول التالي عدة أمثلة على استخدام إعادة تسمية الجداول.

أسماء الموظفين الذين لديهم ولد يدعى وسيم

```
SELECT FirstName, LastName
FROM Employee E, Children C
WHERE E.EmpID = C.EmpID AND ChildName = 'وسيم';
```

الربط الخارجي Outer Join

في حالة **INNER JOIN**، كانت السجلات التي أرجعها الاستعلام، هي السجلات التي تحقق شرط الربط الذي يظهر بعد تعبير **ON**، حيث تم إسقاط السجلات غير المتطابقة من جدول النتائج. أما في حالة الربط الخارجي **Outer Join** فلا يتم إسقاط السجلات غير المتطابقة. للربط الخارجي ثلاثة أنواع: **LEFT**, **RIGHT**, **FULL**.

:Left Outer Join

لأخذ جميع السجلات من الجدول الأول **Table1** و فقط السجلات من الجدول الثاني **Table2** التي تتطابق فيها قيمة الحقل **Column1** من الجدول **Table1** مع قيمة الحقل **Column2** من الجدول الثاني **Table2**، نكتب الصيغة:

```
SELECT * FROM Table1
LEFT OUTER JOIN Table2
ON Table1.Column1 = Table2.Column2;
```

مثال : المطلوب عرض اسم المديرية و اسم و كنية الموظفين بحيث تظهر أسماء كل المديريات و فقط الموظفين المحققين لشرط الربط

```
SELECT DeptName, FirstName, LastName
FROM Department LEFT JOIN Employee
ON Department.DeptID = Employee.DeptID
```

:Right Outer Join

لأخذ جميع السجلات من الجدول الثاني **Table2** و فقط السجلات من الجدول الأول **Table1** التي تتطابق فيها قيمة الحقل **Column1** من الجدول **Table1** مع قيمة الحقل **Column2** من الجدول الثاني **Table2**، نكتب الصيغة:

```
SELECT * FROM Table1
RIGHT OUTER JOIN Table2
ON Table1.Column1 = Table2.Column2;
```

مثال : المطلوب عرض اسم المديرية و اسم و كنية الموظفين بحيث تظهر أسماء كل الموظفين و فقط الأقسام المحققة لشرط الربط

```
SELECT DeptName, FirstName, LastName
FROM Department RIGHT JOIN Employee
ON Department.DeptID = Employee.DeptID
```


: Full Outer Join

لأخذ جميع السجلات من الجدول الثاني Table2 وجميع السجلات من الجدول الأول Table1 بحيث تتوضع السجلات التي تتطابق فيها قيمة الحقل Column1 من الجدول Table1 مع قيمة الحقل Column2 من الجدول الثاني Table2 في نفس السجل من جدول القيم المعادة، نكتب الصيغة:

```
SELECT *
FROM Table1 FULL OUTER JOIN Table2
ON Table1.Column1 = Table2.Column2;
```

مثال : المطلوب عرض اسم المديرية و اسم و كنية الموظفين بحيث تظهر أسماء كل الموظفين و كل الأقسام المحققة لشرط الربط و غير المحققة لشرط الربط

```
SELECT DeptName, FirstName, LastName
FROM Department FULL JOIN Employee
ON Department.DeptID = Employee.DeptID
```

ملاحظة:

ينتج عن عمليات الربط الخارجي، في الحالة العامة، سجلات تحتوي في حقول معينة القيمة NULL بسبب اختلاف عدد السجلات التي نريد ربطها، وهذا ما سنوضحه بالتفصيل لاحقاً مع مثال مناسب لكل نوع من أنواع الربط الخارجي .

الربط الذاتي Self-Join

الربط الذاتي هو ربط جدول مع نفسه. ويمكن ذلك عبر Inner Join أو Outer Join. لناخذ بنية معدلة عن الجدول Employee الذي عرّفناه في بداية الجلسة، وذلك بإضافة ManagerID إليه كما هو موضح في الجدول التالي.

```
Alter table employee add ManagerID int
```

```
update Employee set ManagerID = 1
```

```
where EmpID in (2,3,4)
```

Employee					
EmpID	FirstName	LastName	Gender	DeptID	ManagerID
1	عمر	السهي	male	actg	NULL
2	طارق	العواد	male	mktg	1
3	سامية	سمان	female	mktg	1
4	حنان	آغا	female	NULL	1

يعبر الحقل الجديد ManagerID عن الرقم EmpID لمدير الموظف. وإذا كانت قيمته NULL فهذا يعني أن الموظف الحالي هو مدير.

تمرين: اكتب الاستعلام الذي يعطي أسماء جميع الموظفين مع اسم المدير (إن كان موجوداً).

```
Select E.FirstName,E.LastName,
M.FirstName, M.LastName
From Employee E LEFT JOIN Employee M
ON E.ManagerID = M.EmpID;
```

لا حظ الاستخدام الإجباري لإعادة التسمية (Aliasing) ضمن عبارة From في التمرين السابق.



الاستعلامات الفرعية Sub Queries

- الاستعلام الفرعي هو تعليمة انتقاء SELECT تعيد قيمة وحيدة
- الاستعلام الفرعي يمكن تضمينه ضمن تعليمة SELECT, INSERT, UPDATE, DELETE
- الاستعلام الفرعي يمكن تضمينه ضمن استعلام فرعي آخر
- يسمى الاستعلام الفرعي بالاستعلام الداخلي Inner Query
- يسمى الاستعلام الأب بالاستعلام الخارجي Outer Query

أنواع الاستعلامات الفرعية :

- الاستعلامات الفرعية المرتبطة باستعلام رئيسي :
- الاستعلامات الفرعية الغير مرتبطة (المستقلة)
- الاستعلام الفرعي كعمود من أعمدة الاستعلام الرئيسي

استعمال الاستعلامات الفرعية ضمن شرط WHERE

مثال :المطلوب عرض اسم الكتاب الذي يملك أعلى سعر في جدول الكتب في قاعدة البيانات pubs

```
Select title from titles where price = (select max(price) from titles)
```

استعمال الاستعلامات الفرعية بدلاً من عامود

مثال : المطلوب عرض اسم الموظف من جدول الموظفين وتاريخ الطلبة من جدول الطلبات و ذلك للطلبة رقم 10250 حيث يمكن القيام بذلك باستخدام الربط بين الجدولين و لكن يمكن القيام بذلك باستخدام الاستعلام الفرعي كما هو مبين بالشكل

```
select (select FirstName from Employees where EmployeeID  
orders.EmployeeID),OrderDate  
from orders where OrderID = 10250
```

الاستعلامات الفرعية التي يمكن أن تعيد عدة قيم:

الاستعلامات الفرعية مع التعبير IN, NOT IN
اشترطنا حتى الآن في الاستعلامات الفرعية، سواء تلك المستخدمة كأعمدة في الاستعلام الرئيسي أو تلك المستخدمة ضمن شرط التعبير WHERE، أن يعيد الاستعلام الفرعي قيمة وحيدة كي لا يفشل الاستعلام. لكي تتمكن من استخدام الاستعلامات الفرعية التي تعيد أكثر من قيمة نستخدم التعبير IN وذلك وفق الصيغة التالية:

```
SELECT columnA, columnB  
FROM Table_Name  
WHERE columnC IN (Subquery);
```

لكن يجب أن نراعي أن يعيد الاستعلام الفرعي المستخدم مع التعبير IN حقلاً واحداً فقط (نقصد هنا حقلاً واحداً وليس قيمة واحدة). (بينما من الممكن له أن لايعيد أي سطر أو يعيد أي عدد من الأسطر من ضمن ذلك الحقل الواحد) أي كما يلي:

```
SELECT column1 FROM Table1;
```

مثال:

نريد قائمة بأسماء الناشرين الذين ينشرون كتباً في مجال إدارة الأعمال:

```
SELECT PUB_NAME
FROM Publishers
WHERE PUB_ID IN (SELECT PUB_ID FROM TITLES
WHERE TYPE = 'BUSINESS')
```

ومرة أخرى أسماء الناشرين الذين لا ينشرون كتباً في مجال إدارة الأعمال:

```
SELECT PUB_NAME
FROM Publishers
WHERE PUB_ID NOT IN (SELECT PUB_ID FROM TITLES
WHERE TYPE = 'BUSINESS')
```

الاستعلامات الفرعية في تعليمات الإضافة، الحذف والتعديل

يمكن استخدام الاستعلامات الفرعية مع جميع تعليمات DML الأربعة

مثال:

نريد مضاعفة أسعار جميع الكتب للناشرين الموجودين في الولايات المتحدة فقط

```
UPDATE titles
SET price = price * 2
WHERE pub_id IN
(SELECT pub_id FROM publishers WHERE country = 'USA')
```

ملاحظات

يتضمن الاستعلام الفرعي الفقرات التالية:

- فقرة الانتقاء النظامية **Select**
- فقرة جداول الانتقاء **From**
- فقرة الفلتر **Where** وهي فقرة اختيارية
- يجب أن يحاط الاستعلام الفرعي دوماً بأقواس ()
- يمكن للاستعلام الفرعي أن يظهر في مكان أي تعبير شريطة أن يعيد قيمة واحدة.
- يمكن تضمين استعلام فرعي ضمن استعلام فرعي آخر حتى 32 مستوى

التعبير EXISTS:

يستخدم التعبير EXISTS للتحقق من إعادة الاستعلام الفرعي الذي يليه لأي سجل. ويأخذ التعبير كاملاً القيمة TRUE في حال أرجع الاستعلام الفرعي سجلاً أو أكثر، والقيمة FALSE إذا لم يُرجع الاستعلام الفرعي أي سجل.

مثال:

المطلوب عرض الشركات و المدينة للزبائن الذين لديهم طلبيات منذ بداية عام 1998 و ذلك في قاعدة البيانات northwind

```
select CompanyName,ContactName,city from Customers c
where exists (select * from orders o where o.CustomerID=c.CustomerID and o.OrderDate >
'01/01/1998')
```

أسئلة عن المحاضرة :

من الجداول الموجودة في المحاضرة العملية المطلوب :

- 1- كتابة استعلام يعيد عدد الأولاد للعاملين بكل قسم أي يعطي اسم القسم و مجموع عدد الأولاد للأباء المنتمين لنفس القسم
- 2- المطلوب عرض اسم الموظف الذي لديه أكثر عدد من الأولاد
- 3- المطلوب عرض اسماء كل الموظفين و فقط الأقسام التي تحوي موظفين
- 4- المطلوب حذف سجلات الأولاد الذين أبائهم يعملون في قسم actg أي قسم الحسابات

جامعة المنارة