

الفصل الخامس

عرض البيانات من اكثر من جدول

الأهداف:

1. استخدام جملة SELECT لاستخلاص بيانات من أكثر من جدول وذلك باستخدام طرق الربط المختلفة .
2. إستخلاص البيانات التي لا تقابل شرط الربط باستخدام Outer Join.
3. ربط عمودين بنفس الجدول ويسمى Self Join.

الحصول علي البيانات من أكثر من جدول

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
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

فقد تم اختيار العمود DEPARTMENT_NAME من جدول DEPARTMENTS واختيار العمود EMPLOYEE_ID والعمود DEPARTMENT_ID من جدول EMPLOYEES فتم عرض هذه البيانات معا في جدول جديد وهذا ما يسمى بالربط. Cartesian Products عندما لانضع شرط الربط او نضع شرط ربط خاطئ وتكون النتيجة عرض عدد كبير من الصفوف وتكون النتيجة غير ذات معني. كما في المثال التالي:

Cartesian Products

EMPLOYEES (20 rows)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

20 rows selected.

DEPARTMENTS (8 rows)

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

8 rows selected.

Cartesian
product: →
20x8=160 rows

EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
100	90	1700
101	90	1700
102	90	1700
103	60	1700
104	60	1700
107	60	1700
...		

160 rows selected.

وكما ذكرنا ففي هذا المثال بسبب حذف الشرط WHERE تم ربط كل الصفوف في جدول الموظفين مع كل الصفوف في جدول الاقسام و بذلك يكون الناتج 160 صفوف.

```
SELECT last_name, department_name dept_name
FROM employees, departments;
```

LAST_NAME	DEPT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration

160 rows selected.

أنواع الروابط

Types of Joins	
Oracle Proprietary Joins (8i and prior): <ul style="list-style-type: none"> • Equijoin • Nonequijoin • Outer join • Self join 	SQL: 1999 Compliant Joins: <ul style="list-style-type: none"> • Cross joins • Natural joins • Using clause • Full or two sided outer joins • Arbitrary join conditions for outer joins

يستخدم الربط لعرض البيانات من أكثر من جدول

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column1 = table2.column2;
```

يستخدم الأمر SELECT عن طريق كتابة اسم الجدول المراد الاختيار منه table1 واسم العمود المراد عرضه column1 ويفصل بينهم بنقطة .
ثم الجدول الثاني المراد ربطه واسم العمود الذي يراد عرضه column2 ويفصل بينهم بنقطة .

ثم الشرط WHERE ليتم ربط الجدولين معا حيث يتم مساواة قيم العمودين في الجدولين.
وهنا نجد ان عمود رقم الإدارة في الجدول الأول للموظفين يتساوى مع عمود رقم الإدارة في الجدول الثاني للإدارات.

*لكي ترتبط مجموعة من الجداول معا ، كمثال لربط أربعة جداول، يتطلب ذلك ثلاثة روابط كحد أدنى.

ما هو الربط المتساوي EquiJoin؟

EMPLOYEES

EMPLOYEE_ID	DEPARTMENT_ID
200	10
201	20
202	20
124	50
141	50
142	50
143	50
144	50
108	60
104	60
107	60
149	80
174	80
175	80
...	...

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
20	Marketing
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
60	IT
60	IT
60	IT
80	Sales
80	Sales
80	Sales
...	...

Foreign key Primary key

الربط المتساوي

وفية يتم تساوي بين قيم عمود بالجدول الأول مع قيم عمود بالجدول الثاني .
ويتضمن هذا النوع استخدام المفتاح الرئيسي والمفتاح الأجنبي
(Primary Key و Foreign Key) ويسمي هذا النوع من بالربط البسيط أو الربط الداخلي (Simple Join و Inner Join).

مثال يوضح الربط بطريقة EquiJoin

```
SELECT employees.employee_id, employees.last_name,
       employees.department_id, departments.department_id,
       departments.location_id
FROM   employees, departments
WHERE  employees.department_id = departments.department_id;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourges	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500
144	Vargas	50	50	1500
...

19 rows selected

وكما نري في المثال السابق فى فقرة WHERE يحدد شرط الربط المستخدم هو إشارة التساوي (=) وهو:

EMPLOYEES.DEPARTMENT_ID = DEPARTMENTS.DEPARTMENT_ID

لأن عمود DEPARTMENT_ID موجود فى كلتا الجدولين، ويجب أن يكون مُسبق بأسم الجدول الخاص به كي نتجنب الغموض.

إضافة شرط آخر من شروط الربط عن طريق استخدام المعامل and

EMPLOYEES

LAST_NAME	DEPARTMENT_ID
Whalen	10
Hartstein	20
Fay	20
Mourges	50
Rajs	50
Davies	50
Matos	50
Vargas	50
Hunold	60
Ernst	60

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
20	Marketing
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
60	IT
60	IT

من الممكن إضافة شرط آخر عن طريق استخدام معامل and فمثلا:
AND last_name = 'Matos'

حيث يتم إضافة AND بعد الشرط فى WHERE
ثم يلي AND الشرط الجديد.

```
SELECT last_name, employees.department_id,
       department_name
FROM employees, departments
WHERE employees.department_id = departments.department_id
AND last_name = 'Matos';
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Matos	50	Shipping

ففي المثال السابق تم عرض الاسم و رقم القسم واسم القسم للموظف Matos.

استخدام الأسماء المستعارة للجداول

- * استخدام الأسماء المستعارة للجداول يبسط عملية الاستعلام.
- * يحسن الأداء باستخدام اول الحروف من اسماء الجداول.

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM   employees e, departments d
WHERE  e.department_id = d.department_id;
```

من الممكن استخدام الأسماء المستعارة للجداول المراد ربطها وهي أسماء سهلة مؤقتة بالجداول. تستخدم فقط أثناء تنفيذ أمر SELECT حيث تفيد في اختصار الوقت وكتابة الأمر. ويتم تعريف Alias في أمر FROM لكل جدول. كما هو موضح بالمثال

FROM employees e , departments d

بحيث يكتب أسم الجدول كاملا بتبعية الاسم المستعار لة ففي المثال السابق كان أسم الجدول employees و الاسم المستعار لة e وأسم الجدول الثاني department والاسم المستعار لة d .

1. الأسماء المستعارة للجداول يجب أن تكون في حدود 30 حرف في الطول لكن إذا كانت أقصر في الطول كان ذلك أفضل.
2. الاسم المستعار للجداول يكون في فقرة FROM ، إذن ذلك الاسم المستعار للجداول يجب أن يسبق اسم العمود المراد عرضه.
3. الأسماء المستعارة للجداول يجب أن تكون ذات مغزى. من الممكن الربط بين أكثر من جدول

EMPLOYEES		DEPARTMENTS		LOCATIONS	
LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID	LOCATION_ID	CITY
King	90	10	1700	1400	Southlake
Kochhar	90	20	1800	1500	South San Francisco
De Haan	90	50	1900	1700	Seattle
Hunold	60	60	1400	1800	Toronto
Ernst	60	60	2500	2500	Oxford
Lorentz	60	90	1700		
Maugos	50	110	1700		
Rajs	50	190	1700		
Davies	50	8 rows selected.			
Matos	50				
Vargas	50				
Zlotkey	60				
Abel	60				
Taylor	60				

20 rows selected.

ولكي تكون هذه العملية صحيحة فإن عدد عمليات الربط دائما تكون أقل من عدد الجداول المستخدمة بواحد.

حيث أن هناك ثلاث جداول تم الربط بينهم بعمليتي ربط فقط وهما:

```
e.department_id = d.department_id
d.location_id = l.location_id
```

```
SELECT e.last_name, d.department_name, l.city
FROM   employees e, departments d, locations l
WHERE  e.department_id = d.department_id
AND    d.location_id = l.location_id;
```

LAST_NAME	DEPARTMENT_NAME	CITY
Hunold	IT	Southlake
Ernst	IT	Southlake
Lorentz	IT	Southlake
Mourgos	Shipping	South San Francisco
Rajs	Shipping	South San Francisco
Davies	Shipping	South San Francisco

19 rows selected.

ففى المثال السابق تم الربط بين ثلاث جداول Employees ,Department Location وذلك لعرض الاسم و اسم الادارة والمدينة التى بها تلك القسم.

الربط الغير متساوي NonEquiJoin

JOB_GRADES

GRA	LOWEST SAL	HIGHEST SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	16000	24999
F	26000	40000

EMPLOYEES

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hunold	9000
Ernst	6000
Lorentz	4200
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Zlotkey	10500
Abel	11000
Taylor	8600

20 rows selected.

المرتبات في جدول الموظفين يجب أن تكون بين أقل مرتب وأعلى مرتب في جدول JOB_GRADES .

الربط الغير متساوي (NonEquiJoin) توجد به علاقة غير مباشرة لربط جدولين ولا تستخدم فيه إشارة التساوي (=). والعلاقة بين عمود المرتبات في جدول الموظفين يجب أن تكون بين أقل مرتب وأعلى مرتب في جدول JOB_GRADES . ففى المثال التالى يقوم بمعرفة مركز او موقع مرتب كل موظف وذلك بمعرفة نطاق كل شريحة من المرتبات (بدايتها ونهايتها) ومقارنتها بمرتب كل موظف فى جدول الموظفين. إذا كان ضمن الشريحة فإنه يعرض اسم الشريحة. كما فى المثال التالى...

مثال يوضح الربط الغير المتساوي NonEquiJoin

```
SELECT e.last_name, e.salary, j.grade_level
FROM   employees e, job_grades j
WHERE  e.salary
      BETWEEN j.lowest_sal AND j.highest_sal;
```

LAST_NAME	SALARY	GRA
Matos	2600	A
Vargas	2500	A
Lorentz	4200	B
Mourgos	5800	B
Rajs	3500	B
Davies	3100	B
Whalen	4400	B
Hunold	9000	C
Ernst	6000	C

20 rows selected.

حيث لا توجد علاقة مباشرة بين الجدولين employees e, job_grades بينهم ولذلك تم استخدام BETWEEN بدلا من يساوي في شرط الربط. لاحظ ان قيمة المرتب للموظف تمتد بين أقل قيمة وأعلى قيمة مرتب في جدول درجات المرتب. كل مرتبات الموظفين ممتدة بواسطة جدول درجات. الوظيفة JOB_GRADE فلا يوجد موظف يكسب أقل من أصغر قيمة في عمود LOWEST_SAL ولا يوجد موظف يكسب أكثر من العمود HIGHEST_SAL .

ملاحظة:

(<=) و (>=) يمكن أن تستخدم و لكن BETWEEN للتبسيط . تذكر عند تحديد القيم يجب ان تكون أول قيمة منخفضة وثاني قيمة مرتفعة وذلك عند استخدام BETWEEN.

الربط الخارجي (outer join)

يستخدم لعرض صفوف موجودة بالجدول ولكن لا ينطبق عليها شرط الربط ويتم ذلك باستخدام إشارة الجمع (+) مع شرط الربط وتوضع في جهة العمود المراد عرض بياناته .

DEPARTMENTS

DEPARTMENT_NAME	DEPARTMENT_ID
Administration	10
Marketing	20
Shipping	30
IT	40
Sales	50
Executive	60
Accounting	70
Contracting	80

8 rows selected.

EMPLOYEES

DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
90	Hunold
90	Ernst
90	Lorentz
90	Mourgos
90	Rajs
90	Davies
90	Mates
90	Vargas
90	Turney

20 rows selected.

لاحظ عدم وجود موظفين في القسم 190

بالنظر الى الجدولين نجد ان هناك ادارة موجودة في جدول الاقسام وغير موجودة في جدول الموظفين. فلنفترض ان هذه الادارة جديدة ولم يعين بها موظفين حتى الان. و اردنا ان نرى تلك الادارة بطريقة الربط Equijoin كما في المثال التالي فنجد اننا لا يمكن رؤيتها بتلك الطريقة ولذلك يتم استخدام طريقة الربط Outer join كما سوف نرى.

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Harstein	20	Marketing
Fay	20	Marketing
Mourgos	30	Shipping

19 rows selected.

لاحظ ان تلك الصفوف ليست كاملة حيث انه لم يعرض الادارات التي ليس بها موظفين. ولمعالجة ذلك يتم استخدام Outer Join .

الربط الخارجي Outer Join

* يستخدم لكي يمكن رؤية الصفوف التي لا تقابل شرط الربط.
* معامل الربط الخارجي هو إشارة الزائد (+).

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column(+) = table2.column;

SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column = table2.column(+);
```


يتم استخدام إشارة الجمع جهة العمود المراد عرض جميع بياناته أثناء عملية الربط بين العمودين وتوضع علامة (+) جهة العمود الذي يوجد فيه النقص. ويجب وضع علامة(+) بين قوسين .
(+) *table2.column* رمز الربط الخارجي (+) يمكن أن يوضع علي جانب واحد في شرط فقرة WHERE لكن لايمكن ان يوجد على كلا من الجانبين.

استخدام الربط الخارجي Outer Join

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id(+) = d.department_id ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mourgos	50	Shipping
Rajs	50	Shipping
Davies	50	Shipping
Mates	50	Shipping
...		
Gietz	110	Accounting
		Contracting

20 rows selected.

لا يوجد موظف في هذه الخانة

في هذا **المثال** حيث أنه تم استخدام إشارة الجمع جهة العمود المراد عرض جميع بياناته أثناء عملية الربط بين العمودين وتوضع علامة (+) جهة العمود الذي يوجد فيه النقص. والمثال يعرض last_name, department_id, department_name.

لاحظ ان القسم Contracting لا يشتمل علي أي موظفين. وإن القيمة المعروضة فارغة وظاهرة في الناتج.

الربط الداخلي Self Join

EMPLOYEES (WORKER)

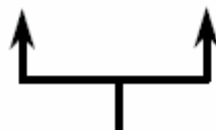
EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

...



رقم الموظف في جدول الموظفين
يساوي رقم الموظف في جدول المدير

الربط الداخلي Self Join يستخدم لربط عمود بعمود آخر في نفس الجدول. وهنا في هذه الحالة يجب ان تتخيل الجدول الواحد(الموظفين) جدولين. الجدول الاول للموظفين والثاني للمديرين حيث ان العمود Employee_ID يعتبر Primary Key للعمود Manager_ID يعتبر Foreign Key لة.
وبهذا يمكنك التعامل مع تلك الجدولين بالطرق Equijoin او Outer join.

```
SELECT worker.last_name || ' works for '
       || manager.last_name
FROM   employees worker, employees manager
WHERE  worker.manager_id = manager.employee_id ;
```

WORKER LAST_NAME 'WORKS FOR' MANAGER LAST_NAME
Kochhar works for King
De Haan works for King
Mougaes works for King
Zlotkey works for King
Hartstein works for King
Whalen works for Kochhar
Higgins works for Kochhar
Hunold works for De Haan
Ernst works for Hunold

19 rows selected.

في المثال السابق يتم إظهار كل موظف بجانب مديرة وبما أن الربط بين عمودين في نفس الجدول فقد تم استخدام الاسم المستعار Employees worker, employees manager

وتم استخدامها كي يتم اعتبار الجدول عبارة عن جدولين منفصلين ويمكن ربطهما بين عمودين داخل نفس الجدول وقد سمي جدول الموظفين باسمين الأول وهو Worker ويمثل الموظفين والثاني Manager وهو المدير .

حيث أن رقم المدير هو أصلاً رقم موظف لذا فقد تم مساواة رقم الموظف بعمود أرقام الموظفين في جدول المديرين Manager بأرقام المديرين بعمود المديرين في جدول الموظفين Worker حيث أن الموظف في جدول المديرين هو مدير في جدول الموظفين.

* تم إظهار 19 موظف فقط ، مع ان هناك 20 موظف و هذا حدث لأن الموظف King هو الرئيس ولا يرئسة احد وبالتالي لا يوجد له رقم مدير في العمود Manager_ID ولعرض KING لابد من استخدام Outer Join .

ربط الجداول بطرق الربط بعد سنة 1999

الشكل الاساسى لتلك الطرق:

```
SELECT table1.column, table2.column
FROM   table1
[CROSS JOIN table2] |
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
 ON(table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
 ON (table1.column_name = table2.column_name)];
```

- * CROSS JOIN يساوي Cartesian product.
- * NATURAL JOIN يربط جدولين معتمد على نفس أسم العمود بالجدولين.
- * JOIN table USING column_name يساوي الربط البسيط (Equijoin) معتمد علي أسم عمود.
- * JOIN table ON يساوي الربط البسيط ايضاً معتمد علي الشرط بعد ON .
- * LEFT/RIGHT/FULL OUTER Join وهو نفس فكرة Outer Join .

(cross join)

وهي نفس Cartesian Product .

```
SELECT last_name, department_name
FROM employees
CROSS JOIN departments ;
```

LAST_NAME	DEPARTMENT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration
Hunold	Administration
...	...

160 rows selected.

المثال يعطي نفس النتائج باستخدام الطريقة الاولى Cartesian Product

```
SELECT last_name, department_name
FROM employees, departments;
```

LAST_NAME	DEPARTMENT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration
Hunold	Administration
Ernst	Administration
...	...

160 rows selected.

Natural Join

* يعتمد هنا على أعمدة الربط في الجدولين الذين لها نفس الاسم.
* يختار الصفوف من الجدولين التي تكون قيمتها متساوية في كل الأعمدة المتناظرة.
* إذا كانت الأعمدة لها نفس الأسماء ولكن نوع البيانات مختلف يحدث خطأ عند الاستخلاص.

```
SELECT department_id, department_name,
       location_id, city
FROM departments
NATURAL JOIN locations ;
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle
110	Accounting	1700	Seattle
190	Contracting	1700	Seattle
20	Marketing	1800	Toronto
80	Sales	2500	Oxford

8 rows selected.

وسوف نعرض نفس النتيجة باستخدام Equijoin

```
SELECT department_id, department_name,
       departments.location_id, city
FROM departments, locations
WHERE departments.location_id = locations.location_id;
```

شرط إضافي علي Natural Join في فقرة WHERE.
فيتم تحديد صفوف الناتج للأقسام 20 أو 50.

```
SELECT department_id, department_name,
       location_id, city
FROM departments
NATURAL JOIN locations
WHERE department_id IN (20, 50);
```

إنشاء الربط باستخدام USING

- * إذا كانت الأعمدة لها نفس الأسماء لكن أنواع البيانات غير مطابقة فيمكنك أن تستخدم USING في تحديد الأعمدة التي يجب أن يتم الربط بها .
- * تستخدم فقرة USING كي تطابق عمود واحد فقط عندما يكون أكثر من عمود متماثل.
- * لا يستخدم أسم الجدول أو الأسم المستعار في الأعمدة التي إسترجعت.

Natural Join تستخدم الأعمدة بمطابقة الأسماء وأنواع البيانات لربط الجداول.
USING يمكن أن تستخدم كي تحدد فقط تلك الأعمدة التي يجب الربط بها.
كمثال

```
SELECT l.citya, d.department_name
FROM locations l JOIN departments d USING (location_id)
WHERE location_id = 1400;
```

لكن هذا المثال خطأ لأنه لا يمكن استخدام Alias في الفقرة

```
SELECT l.city, d.department_name
FROM locations l JOIN departments d USING (location_id)
WHERE d.location_id = 1400;
```

لا يمكن أن تشتمل
 علي Alias

ORA-25154: column part of USING clause cannot have qualifier

مثال على USING

```
SELECT e.employee_id, e.last_name, d.location_id
FROM employees e JOIN departments d
USING (department_id);
```

EMPLOYEE_ID	LAST NAME	LOCATION_ID
200	Whalen	1700
201	Hartstein	1800
202	Fay	1800
124	Mourgos	1900
141	Rajs	1500
142	Dawes	1500
143	Mates	1500
144	Vargas	1500
103	Hunold	1400

19 rows selected.

المثال يظهر ربط عمود رقم القسم بجدول الموظفين بجدول الأقسام وتعرض لنا موقع مكان عمل الموظف.
 وهذا المثال يمكن تنفيذه بـ **Equijoin** كما يلي:

```
SELECT employee_id, last_name,
employees.department_id, location_id
FROM employees, departments
WHERE employees.department_id = departments.department_id;
```

مثال على ON

تستخدم فقرة ON كي تحدد شرط الربط بين الجدولين.

```
SELECT e.employee_id, e.last_name, e.department_id,
d.department_id, d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id);
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Harstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1900
141	Rajs	50	50	1900
142	Davies	50	50	1900
143	Mates	50	50	1900

19 rows selected.

فقرة ON يمكن أن تربط أعمده بها أسماء مختلفة:

```
SELECT e.last_name emp, m.last_name mgr
FROM   employees e JOIN employees m
ON     (e.manager_id = m.employee_id);
```

EMP	MGR
Kochhar	King
De Haan	King
Mourgos	King
Zlotkey	King
Harstein	King
Whalen	Kochhar

19 rows selected.

إن **المثال** السابق يعد Self Join اى (ربط داخلي) لجدول الموظفين بنفسه معتمد علي أعمدة رقم الموظف ورقم المدير بنفس الجدول .

امكانية استخدام اكثر من طريقة ربط

```
SELECT employee_id, city, department_name
FROM   employees e
JOIN   departments d
ON     d.department_id = e.department_id
JOIN   locations l
ON     d.location_id = l.location_id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
103	Southlake	IT
104	Southlake	IT
107	Southlake	IT
124	South San Francisco	Shipping
141	South San Francisco	Shipping
142	South San Francisco	Shipping
143	South San Francisco	Shipping
144	South San Francisco	Shipping

19 rows selected.

ويمكن تنفيذ نفس **المثال** السابق بطريقة Equijoin :

```
SELECT employee_id, city, department_name
FROM   employees, departments, locations
WHERE  employees.department_id = departments.department_id
AND    departments.location_id = locations.location_id;
```

ويمكن تنفيذ نفس **المثال** بطريقة الربط USING :

```
SELECT e.employee_id, l.city, d.department_name
FROM   employees e
JOIN   departments d
USING (department_id)
JOIN   locations l
USING (location_id);
```

قبل سنة 1999

بعد سنة 1999

Oracle	SQL: 1999
Equi-Join	Natural/Inner Join
Outer-Join	Left Outer Join
Self-Join	Join ON
Non-Equi-Join	Join USING
Cartesian Product	Cross Join

Left Outer Join

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Harstein	20	Marketing
...		
De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

20 rows selected

وهذا يعنى ان النقص فى d.department_id وبدلا من وضع (+) عند تلك العمود كما فى الطريقة Outer Join يتم استخدام LEFT OUTER JOIN.

ملحوظة: هنا توضع علامة(+) يسار الشاشة وليس يسار رؤيتك.
والمثال التالى يوضح ذلك:

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e, departments d
WHERE d.department_id (+) = e.department_id;
```

Right Outer Join

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
...		
Whalen	10	Administration
Harstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Gietz	110	Accounting
		Contracting

20 rows selected

وهذا يعنى ان النقص فى e.department_id وبدلا من وضع (+) عند تلك العمود كما فى الطريقة Outer Join يتم استخدام Right OUTER JOIN.

ملحوظة: هنا توضع علامة(+) يمين الشاشة وليس يمين رؤيتك.
والمثال التالى يوضح ذلك:

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e, departments d
```

WHERE d.department_id = e.department_id (+);

Full outer join

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e
FULL OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
...		
De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
		Contracting

21 rows selected.

هذا الاستعلام يسترجع كل الصفوف في الجدولين حتى إذا كانت غير متماثلة
شروط إضافية

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id)
AND e.manager_id = 149 ;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
174	Abel	80	80	2500
176	Taylor	80	80	2500

يمكنك ان تستخدم بعض الشروط الاضافية بعد الربط بين الجدولين لتحديد عدد معين من الصفوف.

أنواع الروابط	
Equijoins	الروابط المتساوية (البسيطة)
Non-equijoins	الروابط الغير متساوية
Outer joins	الروابط الخارجية
Self joins	الروابط الداخلية
Cross joins	الروابط التقاطعية
Natural joins	الروابط الطبيعية
Full or outer joins	الروابط المتكاملة أو الروابط الخارجية