

# مبادئ قواعد المعطيات

المحاضرة السادسة  
د. سمر الحلبي

قواعد المعطيات العلاقاتية  
لغة الاستفسارات البنيوية SQL  
(Structured Query Language)  
الجزء الثاني

3

الاستفسار من عدة جداول

## استخراج المعطيات من عدة جداول

جدول الموظفين

| EMPLOYEE_ID | LAST_NAME | DEPARTMENT_ID |
|-------------|-----------|---------------|
| 100         | King      | 90            |
| 101         | Kochhar   | 90            |
| 205         | Higgins   | 110           |
| 206         | Gietz     | 110           |

20 rows selected.

جدول الأقسام

| DEPARTMENT_ID | DEPARTMENT_NAME | LOCATION_ID |
|---------------|-----------------|-------------|
| 10            | Administration  | 1700        |
| 20            | Marketing       | 1800        |
| 50            | Shipping        | 1500        |
| 190           | Contracting     | 1700        |

8 rows selected.

| EMPLOYEE_ID | DEPARTMENT_ID | DEPARTMENT_NAME |
|-------------|---------------|-----------------|
| 200         | 10            | Administration  |
| 201         | 20            | Marketing       |
| 202         | 20            | Marketing       |
| 124         | 50            | Shipping        |
| 141         | 50            | Shipping        |

|     |     |            |
|-----|-----|------------|
| 205 | 110 | Accounting |
| 206 | 110 | Accounting |

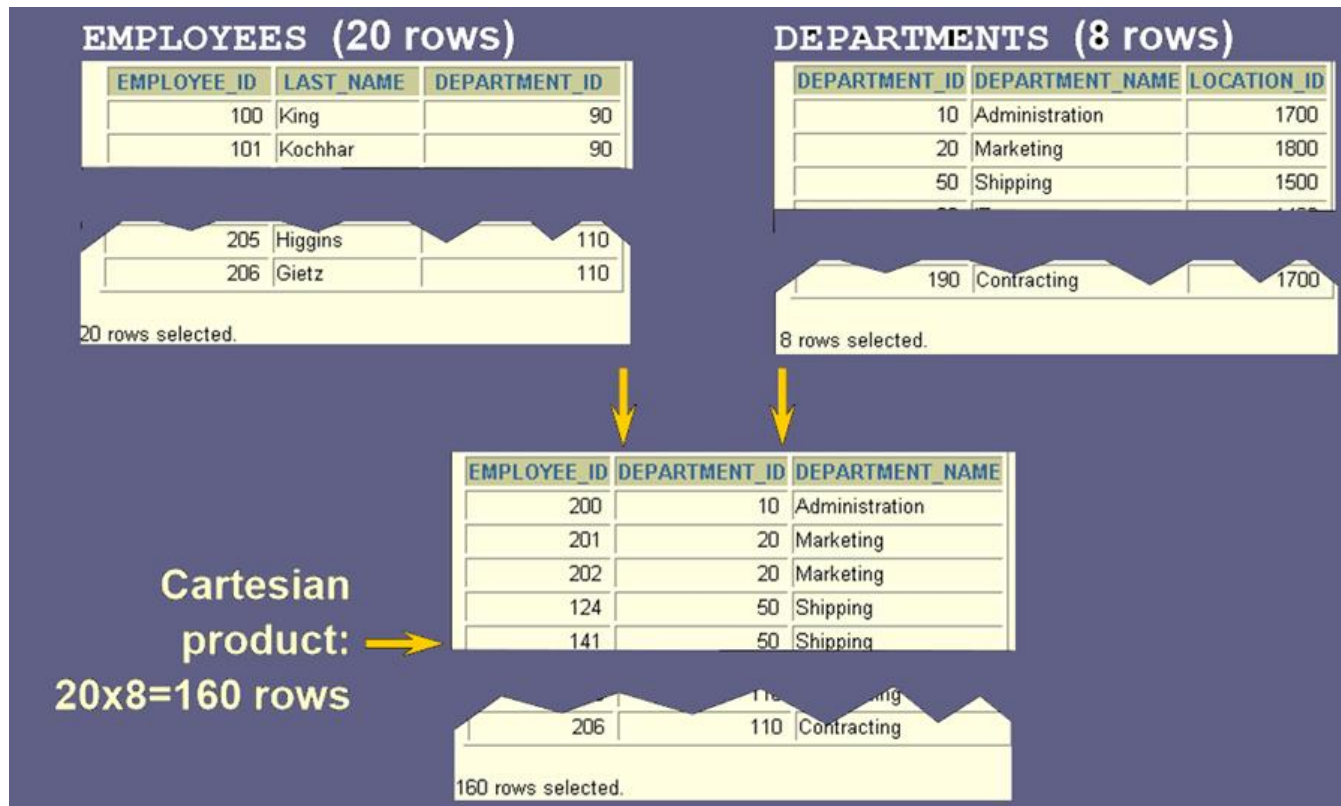
19 rows selected.

## الجداء الديكارتي والدمج

- يجري حساب الجداء الديكارتي عندما تتضمن عبارة From أكثر من جدول أي بالشكل التالي:

```
SELECT    table1.column, table2.column
FROM      table1, table2
```

- مثال:



## دمج الجداول

- لتجنب الجداء الديكارتى لابد من وجود شرط دمج بين الجداول ضمن عبارة

where

- أنواع دمج الجداول:

١. الدمج الطبيعي (الداخلي) (inner join)

٢. الدمج الخارجي (outer join)

## الدمج الطبيعي

- يستخدم الدمج الطبيعي لاستخراج المعطيات من أكثر من جدول، تأخذ عبارة select الشكل التالي:

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

- يوجد شرط الدمج الطبيعي بين جدولين ضمن عبارة where
- **ملاحظة:** يوضع اسم الجدول واسم العمود إذا كان هناك عمودان يحملان نفس الاسم من الجدولين و إلا يكتفى باسم العمود.

## الدمج الطبيعي

- آلية الدمج الطبيعي:

- ✓ يطابق سجلات الجدولين مع بعضها اعتماداً على عمود أو أكثر.
- ✓ ( أي تظهر السجلات من الجدول الأول التي لها مقابلات من الجدول الثاني وبالعكس).



# استخدام الدمج الطبيعي

مثال:

جدول الموظفين

| EMPLOYEE_ID | DEPARTMENT_ID |
|-------------|---------------|
| 200         | 10            |
| 201         | 20            |
| 202         | 20            |
| 124         | 50            |
| 141         | 50            |
| 142         | 50            |
| 143         | 50            |
| 144         | 50            |
| 103         | 60            |
| 104         | 60            |
| 107         | 60            |
| 205         | 110           |
| 206         | 110           |

19 rows selected.

Foreign key

جدول الأقسام

| 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              |
| 110           | Accounting      |
| 110           | Accounting      |

Primary key

# استخدام الدمج الطبيعي

مثال:

الاستفسار عن رقم الموظف واسمه الأخير ورقم قسمه بالإضافة إلى رقم القسم ورقم مكان القسم .

```
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         | Mourgos   | 50            | 50            | 1500        |
| 141         | Rajs      | 50            | 50            | 1500        |
| 142         | Davies    | 50            | 50            | 1500        |
| 143         | Matos     | 50            | 50            | 1500        |
| 205         | Higgins   | 110           | 110           | 1700        |
| 206         | Gietz     | 110           | 110           | 1700        |

19 rows selected.

ملاحظة: تم المطابقة بين رقم القسم من جدول الموظفين مع رقم القسم من جدول الأقسام .

## حل الالتباس في أسماء الأعمدة

- استخدام اسم الجدول إضافة إلى اسم العمود.
- استخدام الأسماء الرديفة للجداول والأعمدة .

مثال:

الاستفسار عن رقم الموظف واسمه الأخير ورقم قسمه بالإضافة إلى رقم القسم ورقم مكان القسم .

```
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;
```

**ملاحظة:**

تم استخدام أسماء رديفة للجداول والأعمدة.

## الدمج الطبيعي لأكثر من جدول

مثال: الاستفسار عن الاسم الأخير للموظف واسم مكان قسمه .

جدول الموظفين (employees)

| LAST_NAME | DEPARTMENT_ID |
|-----------|---------------|
| King      | 90            |
| Kochhar   | 90            |
| De Haan   | 90            |
| Hunold    | 60            |
| Ernst     | 60            |
| Lorentz   | 60            |
| Grant     | 10            |
| Whalen    | 10            |
| Hartstein | 20            |
| Fay       | 20            |
| Higgins   | 110           |
| Gietz     | 110           |

20 rows selected.

(department)

جدول الأقسام

| DEPARTMENT_ID | LOCATION_ID |
|---------------|-------------|
| 10            | 1700        |
| 20            | 1800        |
| 50            | 1500        |
| 60            | 1400        |
| 80            | 2500        |
| 90            | 1700        |
| 110           | 1700        |
| 190           | 1700        |

8 rows selected.

(location)

جدول أماكن الأقسام

| LOCATION_ID | CITY                |
|-------------|---------------------|
| 1400        | Southlake           |
| 1500        | South San Francisco |
| 1700        | Seattle             |
| 1800        | Toronto             |
| 2500        | Oxford              |

## الدمج الطبيعي لأكثر من جدول

مثال: الاستفسار عن الاسم الأخير للموظف واسم مكان قسمه .

```
select e.LAST_NAME ,l.City
from
    employee e ,department d, location l
where
    e. DEPARTMENT_ID= d. DEPARTMENT_ID
    and d.LOCATION_ID=l. LOCATION_ID
```

## الدمج الخارجي

### DEPARTMENTS

| DEPARTMENT_NAME | DEPARTMENT_ID |
|-----------------|---------------|
| Administration  | 10            |
| Marketing       | 20            |
| Shipping        | 50            |
| IT              | 60            |
| Sales           | 80            |
| Executive       | 90            |
| Accounting      | 110           |
| Contracting     | 190           |

8 rows selected.

### EMPLOYEES

| DEPARTMENT_ID | LAST_NAME |
|---------------|-----------|
| 90            | King      |
| 90            | Kochhar   |
| 90            | De Haan   |
| 60            | Hunold    |
| 60            | Ernst     |
| 10            | Whalen    |
| 20            | Hartstein |
| 20            | Fay       |
| 110           | Higgins   |
| 110           | Gietz     |

20 rows selected.



There are no employees in department 190.

## الدمج الخارجي

- آلية الدمج الخارجي:

✓ الحصول على سجلات الجدول الأول والتي ليس لها مقابل من الجدول الثاني وذلك اعتماداً على عمود محدد.

- أنواع الدمج الخارجي:

✓ الدمج الخارجي اليساري (left outer join)

✓ الدمج الخارجي اليميني (right outer join)

✓ الدمج الخارجي الكامل (full outer join)

## التعبير عن الدمج الخارجي

- الشكل العام للدمج الخارجي:

Select <select list>

from

table1 [**left | right | full**] **outer join** table2

**On**

table1.primarykey=table2.foreignkey



## استخدام الدمج الخارجي اليساري

مثال:

الاستفسار عن الاسم الأخير للموظف ورقم واسم قسمه شرط أن لا ينتمي الموظف إلى قسم.

```
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 |
|-----------|---------------|-----------------|
| King      | 90            | Executive       |
| Kochhar   | 90            | Executive       |

|           |     |                |
|-----------|-----|----------------|
| Ernst     | 60  | IT             |
| Grant     |     |                |
| Whalen    | 10  | Administration |
| Hartstein | 20  | Marketing      |
| Fay       | 20  | Marketing      |
| Higgins   | 110 | Accounting     |
| Gietz     | 110 | Accounting     |

20 rows selected.

## استخدام الدمج الخارجي اليميني

مثال:

الاستفسار عن الاسم الأخير للموظف ورقم واسم قسمه شرط أن يكون القسم لا يوجد موظف ينتمي له.

```
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 |
|-----------|---------------|-----------------|
| Whalen    | 10            | Administration  |
| Hartstein | 20            | Marketing       |
| Fay       | 20            | Marketing       |
| Mourgos   | 50            | Shipping        |
| Rajs      | 50            | Shipping        |
| Davies    | 50            | Shipping        |
| Matos     | 50            | Shipping        |

|       |     |             |
|-------|-----|-------------|
| Gietz | 110 | Accounting  |
|       |     | Contracting |

20 rows selected.

# استخدام الدمج الخارجي الكامل

مثال:

الاستفسار عن الاسم الأخير للموظف ورقم واسم قسمه شرط أن يكون القسم لا يوجد موظف ينتمي له والموظف لا ينتمي إلى قسم.

```
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 |
|-----------|---------------|-----------------|
| Abel      | 80            | Sales           |
| Davies    | 50            | Shipping        |
| De Haan   | 90            | Executive       |
| Ernst     | 60            | IT              |
| Fay       | 20            | Marketing       |
| Gietz     | 110           | Accounting      |
| Grant     |               |                 |
| Hartstein | 20            | Marketing       |

|         |    |             |
|---------|----|-------------|
| Zlotkey | 80 | Sales       |
|         |    | Contracting |

21 rows selected.

# طريقة ثانية للتعبير عن الدمج الطبيعي

- طريقة ثانية للدمج الطبيعي:

```
Select <select list>  
from  
    table1 join table2  
On  
    table1.primarykey=table2.foreignkey
```

مثال:

الاستفسار عن رقم الموظف واسمه الأخير ورقم قسمه بالإضافة إلى رقم القسم ورقم مكان القسم .

```
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)
```

4

تجميع المعطيات

استخدام توابع التجميع

## مبدأ عمل توابع التجميع

تعالج توابع التجميع مجموعات من الأسطر وتعيد قيمة وحيدة لكل مجموعة.

### EMPLOYEES

| DEPARTMENT_ID | SALARY |
|---------------|--------|
| 90            | 24000  |
| 90            | 17000  |
| 90            | 17000  |
| 60            | 9000   |
| 60            | 6000   |
| 60            | 4200   |
| 50            | 5800   |
| 50            | 3500   |
| 50            | 3100   |
| 50            | 2600   |
| 50            | 2500   |

|     |       |
|-----|-------|
| 110 | 12000 |
| 110 | 8300  |

20 rows selected.

The maximum salary in the EMPLOYEES table.

MAX(SALARY)

24000

## أنواع توابع التجميع

- `max(column-name)`: يستخدم للحصول على اكبر قيمة في العمود.
- `min(column-name)`: يستخدم للحصول على اصغر قيمة في العمود.
- `count(column-name)`: يستخدم للحصول على عدد القيم في العمود.
- `avg(column-name)`: يستخدم للحصول على معدل القيم في العمود.
- `sum(column-name)`: يستخدم للحصول على مجموع القيم في العمود.

## قواعد كتابة توابع التجميع

```
SELECT      [column,] group_function(column), ...  
FROM        table  
[WHERE      condition]  
[GROUP BY   column]  
[ORDER BY   column];
```



## استخدام توابع التجميع

مثال: الاستفسار عن معدل الرواتب والراتب الأعلى والراتب الأدنى بالإضافة إلى مجموع رواتب الموظفين

```
SELECT AVG(salary) , MAX(salary) ,  
       MIN(salary) , SUM(salary)  
FROM   employees
```

| AVG(SALARY) | MAX(SALARY) | MIN(SALARY) | SUM(SALARY) |
|-------------|-------------|-------------|-------------|
| 8150        | 11000       | 6000        | 32600       |

## استخدام توابع التجميع

- كل تابع تجميع يتجاهل القيم الفارغة (null) ، ماعدا التابع count .
- count (\*) يعيد عدد الأسطر في الجدول مع مراعاة القيم الفارغة.
- count(column-name) يعيد عدد الأسطر في الجدول بدون القيم الفارغة.

```
SELECT COUNT(*)  
FROM employees  
WHERE department_id = 50;
```

| COUNT(*) |
|----------|
| 5        |

## استخدام توابع التجميع

- لعد القيم المختلفة ضمن عمود معين نستخدم التابع count مع

الخيار distinct

مثال:

الاستفسار عن عدد الأقسام من جدول الموظفين.

```
SELECT COUNT(DISTINCT department_id)
FROM employees;
```

```
COUNT(DISTINCTDEPARTMENT_ID)
```