Afficher des données de plusieurs tables

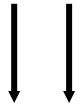
Obtenir des données de plusieurs tables

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

Joindre des tables à l'aide de la syntaxe SQL : 1999

Utilisez une jointure pour interroger des données provenant de plusieurs tables :

```
SELECT   table1.column, table2.column
FROM     table1
[JOIN table2
  ON (table1.column_name = table2.column_name)]
[JOIN table3
  ON (table2.column_name = table3.column_name)]
;
```

Joindre des noms de colonne

EMPLOYEES

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
149	80
174	80
176	80

...

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
60	IT
80	Sales
80	Sales
80	Sales

...

Clé étrangère **Clé primaire**

Créer des jointures avec la clause ON

- La condition de la jointure naturelle est une équijointure de toutes les colonnes portant le même nom.
- Utilisez la clause ON pour indiquer des conditions arbitraires ou pour désigner les colonnes à joindre.
- La condition de jointure est distincte des autres conditions de recherche.
- La clause ON facilite la compréhension du code.

Extraire des enregistrements avec la clause ON

```
SELECT employee_id, last_name, employees.department_id,
 departments.department_id,location_id
FROM   employees JOIN departments
ON     (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

...

19 rows selected.

Différencier les noms de colonne

- **Utilisez des préfixes qui précisent le nom de la table pour différencier les noms de colonne présents dans plusieurs tables.**
- **L'utilisation de préfixes désignant la table améliore les performances.**
- **Utilisez des alias de colonne pour distinguer les colonnes qui présentent des noms identiques, mais qui résident dans des tables différentes.**

Extraire des enregistrements avec la clause ON

```
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	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

...

19 rows selected.

Appliquer des conditions supplémentaires à une jointure

```
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

Créer des jointures à trois liens avec la clause ON

```
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.