

PREMIÈRES INTERROGATIONS SUR LA BASE ENTREPRISE

REQUÊTES D'INTERROGATION SUR UNE TABLE

Ecrivons la requête permettant d'obtenir le contenu de la table 'entreprise'.

select *

from entreprise;

Nous obtenons les résultats suivants :

id_ent	nom	id_com
1	AMBULANCES DA COSTA TRAVERT	1
2	TRANSPORTS LINGAT	5
3	STE TEXTILES DE SAINT LEGER	3
...
25	STE VALENTI ET CIE	15
26	STE TRANSPORT PETROLE PIPELINE (TRAPIL)	16
27	STE NOUVELLE CHEVALIER ET CIE	17

(27 lignes)

Qu'observons-nous ?

- Le nombre de lignes retournées est spécifié.
- Les entêtes sont les noms des colonnes.
- Les valeurs numériques sont alignées à droite, les valeurs alphanumériques à gauche.
- Le symbole '*' remplace l'ensemble des colonnes de la table.

Nous souhaitons n'obtenir que la colonne contenant les noms des entreprises. Il faut alors la spécifier.

```
select nom  
from entreprise;
```

Le résultat est prévisible :

```
----- nom -----  
AMBULANCES DA COSTA TRAVERT  
TRANSPORTS LINGAT  
...  
STE TRANSPORT PETROLE PIPELINE (TRAPIL)  
STE NOUVELLE CHEVALIER ET CIE  
(27 lignes)
```

Nous avons remplacé '*' par le nom de la colonne afin de ne sélectionner qu'une colonne. Cette opération s'appelle une projection en algèbre relationnel.

Nous pouvons restreindre les lignes à certaines d'entre elles répondant à un critère. Par exemple, nous recherchons les entreprises qui se situent dans la commune dont le code est 5. Nous allons utiliser la clause WHERE qui correspond à une opération de sélection en algèbre relationnel.

```
select *  
from entreprise  
where id_com = 5;
```

Nous obtenons

```
id_ent | nom | id_com  
-----+-----+-----  
2 | TRANSPORTS LINGAT | 5  
9 | THIEBAULT JEAN MICHEL (RECUP AUTO CASSE) | 5  
11 | STE SEDIP (SEDIP) | 5  
13 | STE MERGER BATIMENT (MODIPHARM OMNIBAT) | 5  
19 | STE CHAMPAGNE CEREALES | 5  
(5 lignes)
```

Remarquez que nous ne sommes pas obligés d'afficher les colonnes sur lesquelles on pose une condition :

```
select id_ent, nom
      from entreprise
      where id_com = 5;
```

```
id_ent |          nom
-----+-----
      2 | TRANSPORTS LINGAT
      9 | THIEBAULT JEAN MICHEL (RECUP AUTO CASSE)
     11 | STE SEDIP (SEDIP)
     13 | STE MERGER BATIMENT (MODIPHARM OMNIBAT)
     19 | STE  CHAMPAGNE CEREALES
(5 lignes)
```

Je sais que la commune 5 est 'CHALONS EN CHAMPAGNE', je peux ajouter un littéral aux colonnes retenues :

```
select nom, 'CHALONS EN CHAMPAGNE'
      from entreprise
      where id_com = 5;
```

```
          nom | ?column?
-----+-----
TRANSPORTS LINGAT | CHALONS EN CHAMPAGNE
THIEBAULT JEAN MICHEL (RECUP AUTO CASSE) | CHALONS EN CHAMPAGNE
STE SEDIP (SEDIP) | CHALONS EN CHAMPAGNE
STE MERGER BATIMENT (MODIPHARM OMNIBAT) | CHALONS EN CHAMPAGNE
STE  CHAMPAGNE CEREALES | CHALONS EN CHAMPAGNE
(5 lignes)
```

Nous verrons par la suite que d'autres moyens sont à ma disposition pour obtenir ce résultat.

Dans un dernier temps, je souhaite faire apparaître un autre en-tête que ?column? dans le résultat, je peux renommer les colonnes grâce à la clause **AS**.

```
select nom AS "NOM DE L'ENTREPRISE",  
       'CHALONS EN CHAMPAGNE' AS "VILLE"  
from entreprise  
where id_com = 5;
```

NOM DE L'ENTREPRISE	VILLE
TRANSPORTS LINGAT	CHALONS EN CHAMPAGNE
THIEBAULT JEAN MICHEL (RECUP AUTO CASSE)	CHALONS EN CHAMPAGNE
STE SEDIP (SEDIP)	CHALONS EN CHAMPAGNE
STE MERGER BATIMENT (MODIPHARM OMNIBAT)	CHALONS EN CHAMPAGNE
STE CHAMPAGNE CEREALES	CHALONS EN CHAMPAGNE

(5 lignes)

1.1.1. LES CARACTÈRES GÉNÉRIQUES

Recherchons la ligne correspondant à la commune BETHENY.

```
select *  
from commune  
where nom = 'BETHENY';
```

id_com	nom	id_dept
2	BETHENY	51

(1 ligne)

Ceci va bien, mais imaginons que je ne me rappelle plus comment se termine BETHENY, est-ce BETHENI, BETHENIE, ... Je dispose alors pour m'aider de caractères génériques :

- % (Symbole pour cent) qui remplace n'importe quelle chaîne
- _ (Trait bas) qui remplace n'importe quel caractère.

Attention, leur usage nécessite de remplacer l'opérateur '=' par l'opérateur **LIKE**.

Voyons la requête :

```
select *  
  from commune  
 where nom like 'BETHEN%';
```

Recherchons maintenant toutes les communes commençant par la lettre 'B'.

```
select *  
  from commune  
 where nom like 'B%';
```

```
id_com |   nom   | id_dept  
-----+-----+-----  
    1 | BAZEILLES |    2  
    2 | BETHENY  |   51  
    3 | BOURGOGNE |   51  
(3 lignes)
```

Plus difficile, recherchons les communes dont la première lettre est un 'B' et contenant un 'E' qui n'est pas en fin de mot.

Solution en page suivante.

Nous trouvons grâce au caractère générique remplaçant une lettre.

```
select *  
  from commune  
  where nom like 'B%E_%';
```

id_com	nom	id_dept
1	BAZEILLES	2
2	BETHENY	51

(2 lignes)

Je peux aussi combiner des conditions grâce à certains opérateurs logiques. Voyons déjà le **AND** et le **OR**.

Cherchons les communes ayant pour code de département le '10' et commençant par la lettre 'S'.

```
select *  
  from commune  
  where id_dept = 10  
  and nom like 'S%';
```

id_com	nom	id_dept
12	ST ANDRE LES VERGERS	10
13	ST JULIEN LES VILLAS	10
14	ST PARRES AUX TERTRES	10
16	STE SAVINE	10

(4 lignes)

Cherchons les communes ayant pour code de département le '10' et les communes commençant par la lettre 'S' dans une même requête.

```
select *  
  from commune  
  where id_dept = 10  
  or nom like 'S%';
```

id_com	nom	id_dept
4	CHALETTE SUR VOIRE	10
9	LA CHAPELLE ST LUC	10
...
18	TROYES CEDEX	10

(10 lignes)

1.1.2. INTERROGATION SUR PLUSIEURS TABLES (PREMIER PAS)

Reprenons la recherche concernant les entreprises qui se situent dans la commune de 'CHALONS EN CHAMPAGNE'.

Nous n'avons plus cette fois-ci le code. Je dois donc réaliser cette recherche en 2 étapes :

- Rechercher le code de 'CHALONS EN CHAMPAGNE'.
- Rechercher les entreprises qui ont ce code comme valeur de clef étrangère.

```
select id_com
from commune
where nom = 'CHALONS EN CHAMPAGNE';
```

```
id_com
-----
      5
(1 ligne)
```

Puis recherchons les entreprises qui ont pour code la valeur retournée (en l'occurrence: 5).

```
select *
from entreprise
where id_com = 5;
```

```
id_ent |          nom          | id_com
-----+-----+-----
      2 | TRANSPORTS LINGAT    |      5
      9 | THIEBAULT JEAN MICHEL (RECUP AUTO CASSE) |      5
     11 | STE SEDIP (SEDIP)    |      5
     13 | STE MERGER BATIMENT (MODIPHARM OMNIBAT) |      5
     19 | STE CHAMPAGNE CEREALES |      5
(5 lignes)
```

Il est possible de tout faire en une seule requête contenant une autre requête que nous qualifions d'imbriquée.

```
select *
from entreprise
where id_com =
(select id_com
from commune
where nom = 'CHALONS EN CHAMPAGNE');
```

1.1.3. EXERCICE

Réalisez les requêtes suivantes :

1. Liste des employés dont le nom commence par 'MA'.
2. Liste des employés dont le nom contient un 'R' à la troisième position.
3. Liste des employés dont le nom contient la lettre 'D'.
4. Liste des employés gagnant moins de 2000€.
5. Liste des communes de la MARNE.
6. Liste des employés du service 'DIRECTION'.
7. Liste des entreprises des 'ARDENNES'.
8. Liste des employés du service DIRECTION gagnant moins de 1800€.

Solution pages suivantes.

1.1.4. CORRECTION DE L'EXERCICE

1. Liste des employés dont le nom commence par 'MA'. Ici, on utilise le caractère générique '%'

```
select *  
  from employe  
  where nom like 'MA%';
```

id_employe	nom	salaire	id_service	id_chef
1	MARTUSE André	840.00	1	2
8	MARIN Dominique	1296.00	2	7

(2 lignes)

2. Liste des employés dont le nom contient un 'R' à la troisième position. Deux lettres : caractère générique '_' puis le 'R' et n'importe quelle chaîne '%' (y compris la chaîne vide).

```
select *  
  from employe  
  where nom like '__R%';
```

id_employe	nom	salaire	id_service	id_chef
1	MARTUSE André	840.00	1	2
8	MARIN Dominique	1296.00	2	7
9	DURDUX Monique	1537.50	2	7

(3 lignes)

3. Liste des employés dont le nom contient la lettre 'D'. Cette requête pose un problème.

```
select *  
  from employe  
  where nom like '%D%';
```

id_employe	nom	salaire	id_service	id_chef
3	SILLARD Laurence	1867.50	1	2
8	MARIN Dominique	1296.00	2	7
9	DURDUX Monique	1537.50	2	7
14	DUMOITIERS Lucille	2526.00	4	15

(4 lignes)

La requête précédente n'est satisfaisante que si l'on considère nom comme le nom complet : présence de 'MARIN Dominique' acceptable. Dans le cas contraire, il faut recourir à une autre requête.

```

select *
  from employe
  where nom like '%D% %';

```

id_emp	nom	salaire	id_service	id_chef
3	SILLARD Laurence	1867.50	1	2
9	DURDUX Monique	1537.50	2	7
14	DUMOITIERS Lucille	2526.00	4	15

(3 lignes)

En intégrant une espace, on sépare les noms des prénoms à condition que les prénoms composés ne soient pas séparés par une espace, sinon, le résultat sera incorrect. Si ce n'est pas le cas, il faut conclure de cette requête que le modèle de données n'est pas pertinent (rappelez-vous le côté pointilleux de l'analyste). Dans ce domaine de gestion, le nom et le prénom aurait alors dû être séparés.

- Liste des employés dont le nom contient la lettre 'D' gagnant plus de 2000€. On utilise alors l'opérateur logique **AND**.

```

select *
  from employe
  where nom like '%D% %'
  and salaire_mensuel > 2000;

```

num_emp	nom	salaire_mensuel	code_s	num_chef
3	SILLARD Laurence	12450.00	1	2
14	DUMOITIERS Lucille	16840.00	4	15

(2 lignes)

- Liste des communes de la MARNE. Il est nécessaire d'utiliser une requête imbriquée. Il est possible d'utiliser l'opérateur = à la place de like.

```

select *
  from commune
  where id_dept =
  (select id_dept
   from departement
   where nom like 'MARNE');

```

id_com	nom	id_dept
2	BETHENY	51
3	BOURGOGNE	51
5	CHALONS EN CHAMPAGNE	51
10	REIMS	51

(4 lignes)

6. Liste des employés du service 'DIRECTION'. Même opération.

```
select *
  from employe
  where id_service =
  (select id_service
   from service
   where libelle like 'DIRECTION');
```

id_emp	nom	salaire	id_service	id_chef
10	CANILLE Walter	4170.00	3	
11	BOUQUET Antoinette	2178.00	3	10
12	GAUTON Nadine	1342.50	3	10
13	LEGROS Christian	1131.00	3	10

(4 lignes)

7. Liste des entreprises des 'ARDENNES'. Une requête imbriquée qui pose problème ... C'est l'objet de la suite du cours. Recherchons d'abord les communes des Ardennes.

```
select *
  from commune
  where id_dept =
  (select id_dept
   from departement
   where nom like 'ARDENNES');
```

id_com	nom	id_dept
--------	-----	---------

(0 ligne)

Il n'y a pas de département 'ARDENNES', il n'y a donc pas de commune et a fortiori aucune entreprise dans la base. Nous voyons toutefois que l'opérateur '=' s'est acquitté d'une comparaison entre un id_dept et rien (ce que renvoie la requête imbriquée). Tous les SGBDR ne sont pas aussi tolérants et ils ont raison. Rien n'indique ici que le département n'existe pas alors que le renvoi d'une erreur aurait été plus explicite : zéro ligne aurait signifié alors aucune commune dans ce département. Si "select null" renvoie 1 ligne, la comparaison renvoie une erreur.

```
select *
  from commune
  where id_dept = (select null);
```

ERREUR: failed to find conversion function from unknown to integer

8. Liste des employés du service DIRECTION gagnant moins de 1800€. On peut sans problème lier une expression contenant une requête imbriquée avec une autre condition.

```
select *
  from employe
     where id_service = (select id_service
                        from service
                        where libelle like 'DIRECTION')
     and salaire_mensuel < 10000;
```

id_emp	nom	salaire	id_service	id_chef
12	GAUTON Nadine	1342.50	3	10
13	LEGROS Christian	1131.00	3	10

(2 lignes)