

---

## TD5 : introduction à SQL

---

### Passage de l'algèbre relationnelle au SQL

Voiture(imm,marq,mod,coul,cv)  
 Conducteur(num\_permis,nom,prenom)  
 Possede(#imm, #num\_permis, date\_achat)

#### La projection

**Exemple 1.** Quelles sont les immatriculation et les marq des voitures ?

$Proj_{[imm,marq]}(Voiture)$

```
SELECT imm,marq FROM Voiture;
```

**Attention :** faux ami (ça sera le seul) la projection s'écrit *SELECT* en *SQL*.

**Attention :** en SQL, la projection ne supprime pas automatiquement les doublons, si votre projection ne comporte pas la clé primaire vous devez ajouter le mot clé DISTINCT

**Exemple 2.** Quelles sont les couleurs des voitures ?

$Proj_{[coul]}(Voiture)$

```
SELECT DISTINCT coul FROM Voiture;
```

#### La selection

**Exemple 3.** Quelles sont les voitures de couleur Rouge et dont la puissance est supérieure à 5 cv ?

$Selection_{(coul='Rouge' \wedge cv > 5)}(Voiture)$

```
SELECT * FROM Voiture WHERE coul='Rouge' AND cv > 5;
```

Le \* signifie qu'on garde tous les attributs (donc pas de projection).

Evidement comme en algèbre on peut combiner selection et projection dans une même requête.

**Exemple 4.** Quelles sont les marques des voitures de couleur Rouge ?

$Projection_{[marq]}(Selection_{(coul='Rouge')}(Voiture))$

```
SELECT DISTINCT marq FROM Voiture WHERE coul='Rouge' ;
```

**Exemple 5.** Quelles sont les marques des voitures dont le nom contient la lettre 'a' ?

```
SELECT DISTINCT marq FROM Voiture WHERE marq LIKE '%a%' ;
```

Ici le % est le caractère joker (vous verrez ça en cours de systeme). LIKE 'Re%' qui commencent par 'Re'. LIKE '%a' qui finissent par un 'a'.

#### La jointure

**Exemple 6.** Quels sont les numéros de permis des heureux possesseurs de Kangoo orange ?

$Project_{[num\_permis]}(Select_{mod='Kangoo' \wedge coul='Orange'}(Voiture \bowtie_{imm=imm} Possede))$

```
SELECT DISTINCT num_permis
FROM voiture JOIN possede ON voiture.imm = possede.imm
WHERE mod='Kangoo' AND coul='Orange' ;
```

Notons que l'utilisation d'alias pour les tables permet de raccourcir l'expression des requêtes :

```
SELECT DISTINCT num_permis
FROM voiture v JOIN possede p ON v.imm = p.imm
WHERE mod='Kangoo' AND coul='Orange' ;
```

On remarque que comme en algèbre, une requête SQL "standard" peut comprendre une projection, une sélection et une jointure.

On peut bien sûr faire plusieurs jointures dans une même requête.

**Exemple 7.** Quels sont les noms, prénoms des heureux possesseurs de Kangoo orange ?

$$Project_{[nom,prenom]}(Select_{mod='Kangoo' \wedge coul='Orange'}(Voiture \bowtie_{imm=imm} Possede \bowtie_{num\_p=num\_p} Conducteur))$$

```
SELECT DISTINCT nom,prenom
FROM conducteur c JOIN possede p ON p.num_p = c.num_p
      JOIN voiture v ON v.imm = p.imm
WHERE mod='Kangoo' AND coul='Orange' ;
```

## Les opérateurs ensemblistes

**Exemple 8.** Quelles sont les marques qui NE fabriquent PAS des voitures de couleur Rouge ?

$$Proj_{[marq]}(Voiture) \\ - \\ Proj_{[marq]}(Selection_{(coul='Rouge')}(Voiture))$$

```
SELECT marq FROM voiture
MINUS
SELECT marq FROM voiture WHERE coul='Rouge';
```

**Exemple 9.** Quelles sont les marques qui fabriquent des voitures de couleur Rouge ou des voitures de couleur Noir ?

$$Proj_{[marq]}Selection_{(coul='Rouge')}(Voiture) \\ \cup \\ Proj_{[marq]}Selection_{(coul='Noir')}(Voiture)$$

```
SELECT marq FROM voiture WHERE coul='Noir'
UNION
SELECT marq FROM voiture WHERE coul='Rouge';
```

**Exemple 10.** Quelles sont les marques qui fabriquent des voitures de couleur Rouge et des voitures de couleur Noir ?

$$Proj_{[marq]}Selection_{(coul='Rouge')}(Voiture) \\ \cap \\ Proj_{[marq]}Selection_{(coul='Noir')}(Voiture)$$

```
SELECT marq FROM voiture WHERE coul='Noir'
INTERSECT
SELECT marq FROM voiture WHERE coul='Rouge';
```

## L'autojointure

**Exemple 11.** Quelles sont les marques qui fabriquent au moins 2 modèles de voitures ?

$$Voiture2 \leftarrow Voiture \\ Proj_{[Voiture.marq]}Selection_{(v.mod \neq v2.mod)}(Voiture \bowtie_{marq=marq} Voiture2)$$

```
SELECT v.marq FROM voiture v
      JOIN voiture v2 ON v.marq = v2.marq
WHERE v.mod != v2.mod ;
```

L'utilisation des alias permet de s'affranchir de la ligne  $Voiture2 \leftarrow Voiture$ .

## Le produit cartésien

**Exemple 12.** Quels sont tous les couples (marq, num\_permis) possibles ?

$$Proj_{[marq,num\_permis]}(Voiture \times Conducteur)$$

```
SELECT marq, num_permis FROM voiture, conducteur;
```

**Exemple 13.** Les numéro de permis des conducteurs qui possèdent tous les marques de voitures :

$$\begin{aligned}
 & Projection_{[num\_p]}(Conducteur) \\
 - & \\
 & Projection_{[num\_p]}( \\
 & \quad Projection_{[num\_p,marq]}(Conducteur \times Voiture) \\
 - & \\
 & \quad Projection_{[num\_p,marq]}(Possede \underset{(imm=imm)}{\bowtie} Voiture))
 \end{aligned}$$

```

SELECT num_p FROM conducteur
MINUS
SELECT num_p FROM (
    SELECT num_p, marq FROM conducteur, voiture
    MINUS
    SELECT num_p, marq FROM possede p JOIN voiture v ON p.imm = v.imm) ;

```

## 1 Se connecter à Oracle avec isqlplus

Dans un navigateur, ouvrez l'url <http://orainfo.iutmontp.univ-montp2.fr:5560/isqlplus/> (accessible aussi de chez vous) et identifiez vous avec votre login, utilisez votre numéro INE comme mot de passe, et mettez IUT comme identificateur de connexion.

Puis changez son mot de passe SQL : Allez dans "Préférences" (en haut à droite), puis "Modification du mot de passe" dans le menu de gauche. Attention à ne pas mettre de @ de \ ou de / dans votre mot de passe.

## 2 En début de chaque session

Taper SET AUTOCOMMIT ON pour que chaque transaction (notamment les insertions dans la base soient exécutées immédiatement)

## 3 Requêtes SQL de Produit, Achat, Client

Comme énoncé en cours, SQL est un langage de requêtes est aussi un langage de définition des tables et des contraintes d'intégrité. Mais le but de ce TD étant la partie langage de requêtes, nous vous fournissons les *scripts* de création et de remplissage des tables

- Téléchargez le script PAC.sql à l'url <http://www.lirmm.fr/~coletta/Cours/BD/PAC.sql>
- Copier/coller le contenu de ce fichier dans le shell SQL.
- Executer le contenu du shell
- Effacer le contenu du shell (on ne l'exécute pas à chaque fois).
- Tapez votre première requête : Exemple : `SELECT * FROM client WHERE nom='Rifflet';` **Attention, il faut mettre les chaînes de caractères entre quotes et respecter le casse.**

## 4 Requetes

**Rappel du schéma relationnel :**

```

PRODUIT(NP, LIB, COUL, QS)
ACHAT(# NCLI, # NP, QA)
CLIENT(NCLI, NOM, ADR)

```

**Q1** Editer (c.a.d. donner la liste) les libellés des produits.

```

LIB
---
Crayon
Encre
Lampe
Crayon Luxe
Agrafeuse
Pese-lettre
Calculatrice

```

**Q2** Editer la liste des noms des clients

```
NOM
---
Jamar
Johnson
Vilarem
Marchand
Rifflet
Defrere
Garey
Jacob
Ullman
Smith
```

**Q3** Editer tous les produits dont la quantité en stock est supérieure ou égale à 50.

```
NP  LIB  COUL  QS
-----
1  Agrafeuse  Rouge  180
2  Calculatrice  Noir  200
4  Lampe  Rouge  99
5  Lampe  Blanc  100
6  Lampe  Bleu  105
7  Lampe  Vert  105
14 Crayon  Luxe  Noir  50
```

**Q4** Editer tous les achats dont la quantité achetée est strictement supérieure à 12.

```
NCLI  NP  QA
-----
101  1  13
101  3  15
101  4  15
```

**Q5** Editer les quantités en stock et les couleurs des lampes (i.e. des produits dont le libellé est 'Lampe')

```
QS  COUL
-----
99  Rouge
100 Blanc
105 Bleu
105 Vert
```

**Q6** Editer les numéros des crayons de luxe dont la quantité en stock est supérieure ou égale à 20.

```
NP
--
11
13
14
```

**Q7** Editer les numéros des clients ayant fait au moins un achat de quantité supérieure ou égale à 10.

```
NCLI
--
108
101
107
104
106
```

**Q8** Editer les numéros des clients dont le nom commence par un J et contient la lettre a.

```
NCLI
--
107
108
```

**Q9** Dresser la liste des numéros des clients qui ont acheté des lampes.

NCLI  
---  
108  
101  
107  
104  
105  
103  
106

**Q10** Dresser la liste des numéros des clients ayant acheté des lampes rouges

NCLI  
----  
101  
105  
103

**Q11** Dresser la liste des produits (libellé) dont au moins un achat est de quantité supérieure ou égale à 13.

LIB  
----  
Encre  
Lampe  
Agrafeuse

**Q12** Dresser la liste des noms des clients qui ont acheté des crayons de luxe en quantité supérieure ou égale à 2.

NOM  
---  
Jamar  
Vilarem  
Defrere  
Garey

**Q13** Numéros des clients n'ayant pas acheté de crayon.

NCLI  
----  
60  
102  
103  
104  
105  
107  
108  
109

**Q14** Nom et numéro des clients ayant fait des achats (au moins un ) et toujours fait des achats dont la quantité est supérieure (strict) à 3.

| NOM     | NCLI |
|---------|------|
| -----   |      |
| Jacob   | 107  |
| Johnson | 105  |

**Q15** Numéro des produits qu'aucun client n'a acheté.

NP  
---  
9

**Q16** Nom et numéro des clients ayant acheté une Lampe et une Agrafeuse.

```

NOM      NCLI
-----
Defrere  101
Ullman   103

```

**Q17** Nom et numéro des clients ayant acheté tous les produits.

aucune ligne sélectionnée

**Q18** Insérer un achat pour le client 101 sur le produit 9.

```
INSERT INTO achat VALUES (101,9,12);
```

En ré-exécutant la Q17, vous devez désormais obtenir :

```

NOM
----
Defrere

```

**Q19** Numéros des produits qui ont été achetés par au moins deux clients.

```

NP
1
6
11
14
4
5
8
3
7
10
12

```

**Q20** (★) Libellé et couleur des produits qui ont été achetés par au moins deux clients.

```

LIB      COUL
-----
Encre    Bleu
Agrafeuse Rouge
Lampe    Rouge
Lampe    Blanc
Lampe    Vert
Crayon Luxe Vert
Crayon Luxe Noir
Lampe    Bleu
Pese-lettre Noir
Crayon    Bleu
Crayon Luxe Rouge

```

**Q21** (★) Numéro du produit, dont la quantité en stock est maximale.

```

NP
--
2

```

**Q22** (★) Nom des clients qui ont acheté les produits achetés par le client 103 (client 103 compris).

```

NOM
----
Jamar
Johnson
Vilarem
Defrere
Ullman

```

**Q23** (★) Nom et numéro des clients qui ont acheté tous les produits achetés par le client 103.

```

NOM
----
Defrere
Ullman

```