

Mémento SQL

Ce mémento a été composé à partir du livre de Philippe Mathieu : Bases de Données (De Merise à JDBC)

Documentation disponibles :

- http://wwwlsi.supelec.fr/www/yb/poly_bd/sql/tdm_sql.html
- <http://www-inf.int-evry.fr/COURS/BD/accueil.html>
- Et des milliers d'autres...

➤ Obtention des données

```
SELECT <liste des noms de colonnes> FROM <liste des noms de tables>

SELECT * FROM tab1                                (toutes les colonnes)
SELECT col1,col3 FROM tab1                        (une partie des colonnes)
SELECT DISTINCT col1 FROM tab1                   (élimine les doublons)
SELECT nom AS "nom personne" FROM tab1           (renommage des colonnes)
SELECT nom "nom personne" FROM tab1              (Oracle : mot clé AS facultatif)
```

➤ Expression des restrictions

```
SELECT * FROM tab1 WHERE <condition(s)>
SELECT * FROM tab1 WHERE ville IN ('Brest ','Rennes ','Paris ')
SELECT * FROM tab1 WHERE age NOT BETWEEN 15 AND 20
SELECT * FROM tab1 WHERE travail IS NULL
SELECT * FROM tab1 WHERE travail IS NOT NULL
```

```
> >= < <= = <>                                (comparateur arithmétiques)
AND OR NOT                                        (comparateur logique)
%                                                 (n'importe quelle séquence de car.)
_ (soulignement)                                (n'importe quel caractère)
```

➤ Tri et présentation des résultats

```
SELECT * FROM tab1 ORDER BY col1                  (tri ascendant par défaut)
SELECT * FROM tab1 ORDER BY col5,col7            (tri par col5 puis tri par col7)
SELECT * FROM tab1 ORDER BY age ASC, sexe DESC   (tri ascendant ou descendant)
```

➤ Expression des jointures

```
SELECT * FROM tab1,tab2 (jointure sans qualification =  
                        produit cartésien)  
SELECT * FROM tab1,tab2 WHERE tab1.col1=tab2.col2 (jointure avec égalité =  
                                                    équijointure)  
SELECT * FROM tab1 t1,tab2 t2,tab3 t3 WHERE t1.col1=t2.col2 AND t2.col2=t3.col3  
                                                    (jointures en cascades)
```

➤ Expression des manipulations de données

```
SELECT designation,prix*1.196 AS "P. TTC" FROM tab1  
YEAR, MONTH, DATE (date)  
SUBSTRING, UPPER, LOWER, CHARACTER_LENGTH (manipulation de chaînes de car.)
```

➤ Les fonctions statistiques

```
AVG (moyenne)  
COUNT (nombre d'éléments)  
MAX (maximum)  
MIN (minimum)  
SUM (somme)  
  
SELECT COUNT(*) FROM tab1  
SELECT SUM(col1) FROM tab2
```

➤ Regroupements

```
SELECT * FROM tab1 GROUP BY col1
```

➤ Sous-requêtes SQL

```
SELECT * FROM tab1 WHERE prix > (SELECT MIN(prix) FROM tab2)  
SELECT * FROM tab1 WHERE nom NOT IN (SELECT nom FROM tab2)  
SELECT * FROM tab1 WHERE prix > ALL (SELECT prix FROM tab2) (sup. à ttes les valeurs)  
SELECT * FROM tab1 WHERE prix > ANY (SELECT prix FROM tab2) (sup. à au moins 1)
```

➤ Opérateurs ensemblistes

S'intercalent entre deux sélections

```
UNION  
INTERSECT  
EXCEPT
```

➤ Insertions

```
INSERT INTO tab1 VALUES ('abc',5,7)
```

(toutes les valeurs doivent être renseignées)

```
INSERT INTO tab1(col1,col7) VALUES ('abc',7)
```

(on ne renseigne que les colonnes indiquées, les cols non précisées sont mises à NULL)

```
INSERT INTO tab1 SELECT * FROM tab2
```

➤ Mises à jour

```
UPDATE tab1 SET col7='abc' WHERE col1=1
```

➤ Suppressions

```
DELETE FROM tab1 WHERE col1=1
```

```
DELETE FROM tab1 WHERE col1 IN (SELECT no FROM tab2)
```

```
DELETE FROM tab1
```

(vide complètement la table)