

## ■ Historique

1950-1960: gestion par simple fichier texte

1960: COBOL (début de notion de base de données)

1968: premier produit de sgbdr structuré (IBM -> IDMS)

1970-74: élaboration de l'outil d'analyse (modèle entité-association)

1974-76: sgbdr avec le langage SEQUEL

1976-79: début du SQL (encore baptisé QUEL et en 1977, QBE)

1981-82: IBM sort ORACLE basé sur le langage SQL

1982: normalisation du SQL en tant quel tel

1984-86: BGSQ, Sybase, Informix, DBASE

1986: normalisation SQL86

1992: normalisation SQL2 (toutes les appli sont basées dessus)

1999: SQL3 (quelques fonctions orientées objet ... controversées...)

2003: SQL2003 (ajout manipulation XML)

2008: SQL2008 (améliorations mineurs)...

## ■ le langage SQL

il est basé sur des instructions structurées.

les DDL (data definition langage)

- CREATE, ALTER, DROP

les DML (data manipulation Langage)

- INSERT, UPDATE, DELETE, SELECT

les DCL (Data control langage)

- GRANT, REVOKE

les TCL (Transaction control langage)

- SET, TRANSACTION, COMMIT



# Exemple de départ

- Gestion d'une entreprise de dépannage :
  - Dictionnaire des données :

code	description	type	taille
dep_id	identifiant <u>depanneur</u>	N	entier
dep_nom	nom du <u>depanneur</u>	A	30
dep_prenom	prenom du <u>depanneur</u>	A	30
dep_age	age du <u>depanneur</u>	N	entier
dep_tel	tel du <u>depanneur</u>	A	14
categorie	<u>categorie</u>	A	20
salaire	salaire du <u>depanneur</u>	N	Reel
ent_id	code de l'entreprise cliente	N	entier
ent_nom	nom de l'entreprise	A	30
ent_tel	tel de l'entreprise	A	14
date_inter	date d'intervention	D	JJ/MM/AAA
inter_action	type d'intervention	A	30
inter_dist	distance client / notre <u>société</u>	N	entier

- MCD : à faire...
- MLD : à faire...
- Déployer la base de données : à faire...

# les commandes

- création d'une base (interdit sur les serveur mmi)

```
CREATE DATABASE mabase;
```

(votre base est déjà créée et vide sur le serveur mmi)

- création d'une table

```
CREATE TABLE depanneurs(  
    dep_id INT AUTO_INCREMENT PRIMARY KEY,  
    dep_nom CHAR(20),  
    dep_prenom CHAR(20),  
    dep_age INT,  
    dep_tel CHAR(15)  
);
```

- suppression d'une table

```
DROP TABLE depanneurs;
```



# les commandes

## modification d'une structure de table

```
ALTER TABLE depanneurs ADD COLUMM qualif CHAR(5);  
ALTER TABLE depanneurs DROP COLUMM qualif;  
ALTER TABLE depanneurs ALTER COLUMM qualif CHAR(10);
```

## ajout d'une données (enregistrement ou ligne)

```
INSERT INTO depanneurs(dep_nom,dep_prenom,dep_age,dep_tel)  
values('Dupond','Bob',34,'02.25.42.42.42');  
INSERT INTO depanneurs  
values('','Dupond','Bob',34,'02.25.42.42.42');
```

## suppression d'une ou plusieurs données

```
DELETE FROM depanneurs WHERE dep_id=5;  
DELETE FROM depanneurs WHERE dep_age>20 AND age<30;  
DELETE FROM depanneurs WHERE dep_nom='Dupond';
```

## modification d'un enregistrement

```
UPDATE depanneur  
  SET dep_nom='Bubu', dep_prenom='Toto'  
  WHERE dep_id=5;  
UPDATE depanneur SET dep_age=dep_age+1 WHERE dep_id=5;  
UPDATE depanneur SET dep_age=dep_age+1 WHERE dep_age>25;
```



- afficher des enregistrements

SELECT ..... FROM ..... WHERE .....

afficher la liste des dépanneurs ayant entre 20 et 30 ans

```
SELECT dep_nom, dep_prenom, dep_age  
FROM depanneurs  
WHERE dep_age>20  
AND dep_age<30;
```

SELECT \*

```
FROM depanneurs  
WHERE dep_age>20  
AND dep_age<30;
```

## ■ les jointures:

afficher la liste des interventions (date, distance et code) avec le nom du dépanneur et le tel des entreprises pour les interventions ayant été faites à plus de 50 Km

écriture acceptée (SQL1):

```
SELECT dep_nom, ent_nom, date_inter, inter_id  
FROM   depanneurs, interventions, entreprises  
WHERE  inter_dist>50  
       AND depanneurs.dep_id=interventions.dep_id  
       AND entreprises.ent_id=interventions.ent_id;
```



- les jointures (suite SQL2):

```
SELECT dep_nom, inter_date, ent_nom  
FROM interventions  
INNER JOIN depanneurs  
ON interventions._dep_id=depanneurs.dep_id  
INNER JOIN entreprises  
ON interventions._ent_id=entreprises.ent_id  
WHERE inter_dist>30;
```

INNER JOIN  
OUTER JOIN  
FULL, LEFT et RIGHT (exemple: LEFT INNER JOIN)

# les commandes

## ■ les tris:

```
SELECT * FROM depanneurs ORDER BY dep_nom ASC
```

```
SELECT * FROM depanneurs ORDER BY dep_nom DESC
```



# les commandes

## ■ voir aussi:

COUNT  
SUM  
AVG  
GROUP BY  
HAVING  
UNION  
LIKE  
AS

Illustration : Voir base

[http://195.83.128.55/adminsql/index.php?db=fmeuzeret\\_src1](http://195.83.128.55/adminsql/index.php?db=fmeuzeret_src1)

gestion des dates (INTERVAL...)

gestion des caractères spéciaux (\' \'\" \n ...)

...

- SUM, AVG, AS... :

Age moyen de tous les dépanneurs

```
SELECT AVG(dep_age) FROM depanneurs
```

```
SELECT AVG(dep_age) AS age_moyen FROM depanneurs
```

Somme de tous les salaires mensuels des dépanneurs

```
SELECT SUM(dep_salaire) FROM depanneurs
```



- COUNT... :

Nombre de dépanneurs de plus de 40 ans

```
SELECT COUNT(dep_id) FROM depanneurs WHERE dep_age >40
```

- GROUP BY combiné avec AVG :

Age moyen des dépanneurs par type de diplôme

```
SELECT dep_diplo , AVG(dep_age) FROM depanneurs GROUP BY dep_diplo
```

- **mysql:**

[www.mysql.com/doc/fr/](http://www.mysql.com/doc/fr/)

[www.mysql.com/doc/fr/Reference.html](http://www.mysql.com/doc/fr/Reference.html)