

Le langage SQL

Qu'est-ce que le SQL ?

- **Structured Query Language** = Langage d'interrogation structuré
- Permet des requêtes de consultation (sélection) mais aussi de créer, modifier, supprimer tables et enregistrements
- Langage normalisé mais implémentations parfois différentes (dialectes)
- L4G (langage de quatrième génération)
- Langage déclaratif, non procédural (procédural : C, Pascal...)
- Inspiré du modèle d'algèbre relationnel de Codd
- Utilisé sur Oracle, MySQL, PostgreSQL...

Historique - Evolution

- 1970 Développement de langages de requête
- 1976 Sequel (IBM)
- 1982 Sequel -> SQL
- 1986 SQL1 (SQL-86) -> norme ANSI
- 1987 SQL1 -> norme ISO
- 1989 **SQL1 révisé (SQL-89) -> norme ANSI/ISO**
- 1992 **SQL2 (SQL-92) -> norme ANSI/ISO**
- 1999 **SQL-99 -> norme ANSI/ISO**
- 2003 **SQL:2003 -> norme ANSI/ISO**

Classification des commandes du langage SQL

L= langage (language)		D= Donnée (Data)		
SQL				
interactif			intégré	dynamique
LDD (DDL) Définition	LMD (DML) Manipulation	LCD (DCL) Contrôle		
CREATE	INSERT	GRANT	DECLARE	PREPARE
DROP	DELETE	REVOKE	CURSOR	DESCRIBE
ALTER	UPDATE	CONNECT	FETCH	EXECUTE
	Interrogation SELECT	COMMIT ROLLBACK SET		

Principales commandes du SQL

- Sélection de champs/lignes `SELECT col1, col2, col3 FROM table1 WHERE <condition>;`
- Suppression de lignes `DELETE FROM table1 WHERE <condition>;`
- Mise à jour de données `UPDATE table1 SET col1=val1, col2=val2 WHERE <condition>;`
- Insertion de lignes `INSERT INTO table1 VALUES (val1, 'val2');`
- Création de table `CREATE TABLE table1 (col1 type1 PRIMARY KEY, col2 type2);`
- Suppression de table `DROP TABLE table1;`

Exemple de table : etud

num	nom	prenom	dateN	note	dep	licence
1	Martin	Véra	85-10-31	13.5	77	2
2	Martin	Annie	85-12-31	15.5	75	1
3	Dupont	Sylvie	83-02-03	15.0	77	2
4	Martin	Annie	83-10-22	05.7	93	1
5	Dupond	Laurent			92	2
6	Lefèvre	Laurent		11.5		3

7

Ecriture du code SQL

- **A éviter**
 SELECT licence, MIN(note), MAX(note),AVG(note) AS
 Moyenne FROM etud WHERE licence=1 GROUP BY
 licence ORDER BY Moyenne ;
- **A privilégier : saut de ligne, indentation**
 SELECT licence,
 MIN(note), MAX(note),
 AVG(note) AS Moyenne
 FROM etud
 WHERE licence=1
 GROUP BY licence
 ORDER BY Moyenne ;

8

Syntaxe du SQL

- Séparateur d'instructions : point virgule
- Séparateur de mots-clés : espace
- Mots-clés du SQL non sensibles à la casse
 -> Convention : mots-clés en majuscules
- Noms d'objets (table, champ...) sensibles à la casse
- Eviter les mots réservés pour les noms d'objets :
 -> action, as, date, from, is, section, session, table, union, work, zone...
- Entourer de '...' ou "..." les noms d'objets comportant un espace
 ex : 'moyenne des étudiants'
- Valeurs de chaînes et date entre '' dans les expressions
- Des commentaires sont possibles sous plusieurs formes :
 /* commentaire sur plusieurs lignes */
 -- espace puis commentaire sur une ligne
 # commentaire sur une ligne

9

Interroger une base de données : SELECT

- SELECT permet d'interroger une base de données
- Le résultat est une nouvelle table (temporaire)
- Sélection de colonnes (projection) dans l'ordre indiqué
- Sélection de lignes (restriction / sélection)
- Des clauses supplémentaires permettent de :
 - faire des calculs sur des regroupements de lignes
 - trier les lignes
 - afficher un certain nombre de lignes de résultats

10

Quelques exemples simples de SELECT

- SELECT * FROM table1;
 On obtient toute la table
- SELECT champ3 FROM table1;
 On obtient uniquement le champ3 de la table
- SELECT champ4, champ3, champ1 FROM table1;

11

Syntaxe du SELECT

- Syntaxe minimale
 SELECT ... FROM table;
- Syntaxe complète (**respecter l'ordre des clauses**)
 SELECT
 [**DISTINCT** | **ALL**] (* | champ [[**AS**] alias], ...)
 FROM table [[**AS**] alias], ...
 [**WHERE** { condition | sous condition}]
 [**GROUP BY** champ , ...]
 [**HAVING** condition]
 [**ORDER BY** {champ | num}{**ASC** | **DESC**}, ...]
 [**LIMIT** [deb,] nb];

12

Sélection de plusieurs colonnes

- Note, nom et prénom de tous les étudiants
 SELECT note, nom, prenom
 FROM etud;

note	nom	prenom
13.5	Martin	Véra
15.5	Martin	Annie
15.0	Dupont	Sylvie
05.7	Martin	Annie
	Dupond	Laurent
11.5	Lefèvre	Laurent

13

Opérateur * : toutes les colonnes

- Liste de toutes les colonnes de la table
SELECT * FROM etud;
- Identique à :
SELECT num, nom, prenom, date_n, note, dep, licence FROM etud;

num	nom	prenom	dateN	note	dep	licence
1	Martin	Véra	85-10-31	13.5	77	2
2	Martin	Annie	85-12-31	15.5	75	1
3	Dupont	Sylvie	83-02-03	15.0	77	2
4	Martin	Annie	83-10-22	05.7	93	1
5	Dupond	Laurent			92	2
6	Lefèvre	Laurent		11.5		3

14

Alias de noms de colonnes

Prénom	Nom de famille
...	...

- En utilisant AS
SELECT prenom AS Prénom,
nom AS 'Nom de famille'
FROM etud;
- Sans utiliser AS (facultatif)
SELECT prenom Prénom,
nom 'Nom de famille'
FROM etud;

15

Caractère d'échappement

- SELECT nom 'Nom d'étudiant' ... -> erreur
- SELECT nom "Nom d'étudiant" ...
- SELECT nom 'Nom d\'étudiant' ...
- SELECT nom 'Nom d\`étudiant' ...

16

Gérer les doublons : ALL (facultatif) et DISTINCT

Nom	prenom	...	dep
Martin	Véra	...	77
Martin	Annie	...	75
Dupont	Sylvie	...	92
Martin	Annie	...	77
Dupond	Laurent	...	75
Lefèvre	Laurent	...	null
...	93
...	77
...	92
...	null

dep
77
75
92
77
75
null
93
77
92
null

dep
77
75
92
null
93

17

Gérer les doublons : DISTINCT s'applique par ligne

dep	licence
93	2
75	1
75	2
77	1
75	3
75	2
null	3
93	2
93	2

dep	licence
93	2
75	1
75	2
77	1
75	3
75	2
null	3
93	2
93	2

18

Tri par ordre croissant (ascendant)

- SELECT * FROM etud ORDER BY note;
- SELECT * FROM etud ORDER BY note ASC;

num	nom	prenom	dateN	note	dep	licence
5	Dupond	Laurent			92	2
4	Martin	Annie	83-10-22	05.7	93	1
6	Lefèvre	Laurent		11.5		3
1	Martin	Véra	85-10-31	13.5	77	2
3	Dupont	Sylvie	83-02-03	15.0	77	2
2	Martin	Annie	85-12-31	15.5	75	1

19

Tri par ordre décroissant

- SELECT * FROM etud ORDER BY note DESC;

num	nom	prenom	dateN	note	dep	licence
2	Martin	Annie	85-12-31	15.5	75	1
4	Dupont	Sylvie	83-02-03	15.0	77	2
1	Martin	Véra	85-10-31	13.5	77	2
6	Lefèvre	Laurent		11.5		3
4	Martin	Annie	83-10-22	05.7	93	1
5	Dupond	Laurent			92	2

20

Tri sur colonnes multiples

- SELECT * FROM etud ORDER BY nom, prenom, dateN DESC;

num	nom	prenom	dateN	note	dep	licence
5	Dupond	Laurent			92	2
3	Dupont	Sylvie	83-02-03	15.0	77	2
6	Lefèvre	Laurent		11.5		3
2	Martin	Annie	85-12-31	15.5	75	1
4	Martin	Annie	83-10-22	05.7	93	1
1	Martin	Véra	85-10-31	13.5	77	2

21

Tri des lignes : ORDER BY

Tri numérique, alphabétique ou chronologique

- SELECT * FROM t1 ORDER BY note;
- SELECT * FROM t1 ORDER BY note ASC;
- SELECT * FROM t1 ORDER BY note DESC;
- SELECT * FROM t1 ORDER BY nom, prenom, note;
- SELECT * FROM t1 ORDER BY 1, 2, 4;
- SELECT * FROM t1 ORDER BY note DESC, nom;
- SELECT nom, note, dep FROM t1 ORDER BY 3, 1;

22

Limiter le nombre d'enregistrements affichés

- SELECT * FROM etud ORDER BY note DESC LIMIT 2; équivalent à :
- SELECT * FROM etud ORDER BY note DESC LIMIT 0,2 ;

nom	prenom	dateN	note	dep	licence
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Véra	85-10-31	13.5	77	2
Lefèvre	Laurent		11.5		3
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2

23

Prélever n ligne(s) à partir de la ligne i + 1

- SELECT * FROM etud LIMIT i,n ;
- SELECT * FROM etud ORDER BY note DESC LIMIT 3,2 ;
- Les lignes sont notées à partir de i=0 et non de i=1 !

i	rang	nom	prenom	dateN	note	dep	licence
0	1	Martin	Annie	85-12-31	15.5	75	1
1	2	Dupont	Sylvie	83-02-03	15.0	77	2
2	3	Martin	Véra	85-10-31	13.5	77	2
3	4	Lefèvre	Laurent		11.5		3
4	5	Martin	Annie	83-10-22	05.7	93	1
5	6	Dupond	Laurent			92	2

2 enregistrements

24

SELECT FROMWHERE

- SELECT -> sélection de colonnes (projection)
- WHERE -> sélection de lignes (sélection = restriction)
- Exemples :
 - SELECT * FROM t1 WHERE nom='Martin';
 - SELECT * FROM t1 WHERE note >=10;
 - SELECT * FROM t1 WHERE note >10 AND dep=77;

champ1	champ2	champ3	champ4	champ5

25

Sélection avec comparaison de valeurs

- Quels sont les étudiants dont le nom est Martin ?
SELECT * FROM etud WHERE nom='Martin' ;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

26

Expression des valeurs constantes

- Chaîne : valeur entre guillemets simples ' ' ou doubles " "
- Date : valeur entre guillemets en présence de séparateur - sans séparateur : guillemets facultatifs
- Echappement de l'apostrophe dans la valeur

Exemples :

Nombre

```
... num=3
... note=13.5
... num='3'
... note='13.5'
```

Chaîne

```
... nom='Martin'
... nom="Martin"
... nom='\'Martin\''
... nom='L\'Hote'
... nom="L'Hote"
... nom='L"Hote"
... nom="L\"Hote\""
```

Date avec séparateur (comme une chaîne)

```
... date_n='1980-12-31'
... date_n='1980/12/31'
```

Date sans séparateur

```
... date_n='19801231'
... date_n=19801231
```

27

Sélection avec connecteur logique AND

- Liste des étudiants de licence 2 dont le nom est Martin
SELECT * FROM etud
WHERE nom='Martin' AND licence = 2 ;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

28

Sélection avec connecteur logique OR

- Liste des étudiants de nom Dupont ou Dupond
SELECT * FROM etud
WHERE nom='Dupont' OR nom='Dupond' ;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

29

Pause....

30

Ajout de parenthèses si besoin

- A SELECT * FROM etud
WHERE (licence=2 OR licence=1)
AND (note < 10 OR note IS NULL) ; -> 2 enregistrements
- B SELECT * FROM etud
WHERE licence=2 OR (licence=1
AND note < 10) OR note IS NULL ; -> 4 enregistrements

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

A → (rows 1, 2, 3, 4)
B → (rows 1, 2, 3, 4)

31

NULL : absence de valeur

- Valeur inconnue
ex : note, date de naissance...
- Valeur non pertinente
ex : nom de jeune fille pour un garçon
- NULL \neq zéro, NULL \neq chaîne vide
ex : note NULL (inconnue) est différent de la note 0
- Null ne se traite pas comme une autre valeur
... = NULL (uniquement pour l'affectation)
... IS NULL (pour des comparaisons)
... IS NOT NULL

32

Sélection et valeur Null

- SELECT * FROM etud
WHERE note \geq 10 OR note $<$ 10 ;
-> Les lignes de valeurs NULL ne sont pas comptées !

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

- SELECT * FROM etud
WHERE note \geq 10 OR note $<$ 10 OR note IS NULL ;
-> pour retrouver toutes les valeurs, y compris NULL

33

Sélection avec comparaison de colonnes

- Quels étudiants ont des résultats identiques en maths et en bio ?
SELECT * FROM etud1
WHERE note_m = note_b ;

nom	prenom	dateN	note_m	note_b
Martin	Véra	85-10-31	13.5	15.0
Martin	Annie	85-12-31	15.5	10.0
Dupont	Sylvie	83-02-03	15.0	15.0
Martin	Annie	83-10-22		
Dupond	Laurent			02.0
Lefèvre	Laurent		11.5	

34

Sélection avec opération sur les colonnes

- Quels étudiants ont des résultats meilleurs en maths (+ 2 pts) qu'en bio ?
SELECT * FROM etud1
WHERE note_m \geq (note_b + 2) ;

nom	prenom	dateN	note_m	note_b
Martin	Véra	85-10-31	13.5	15.0
Martin	Annie	85-12-31	15.5	10.0
Dupont	Sylvie	83-02-03	15.0	15.0
Martin	Annie	83-10-22		
Dupond	Laurent			02.0
Lefèvre	Laurent		11.5	

35

Liste de valeurs avec IN

- SELECT * FROM etud
WHERE dep IN (77,92,93);

Pour les chaînes de caractères,
(val1',val2',val3')

équivalent à :

- SELECT * FROM etud
WHERE dep= 77 OR dep= 92 OR dep= 93;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

36

Intervalle avec BETWEEN (nombre)

- SELECT * FROM etud
WHERE note BETWEEN 11.5 AND 15 ;

équivalent à :

- SELECT * FROM etud
WHERE note \geq 11.5 AND note \leq 15 ;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

- Respecter l'ordre des bornes (inférieure puis supérieure)

37

Intervalle avec BETWEEN (chaîne)

- SELECT * FROM etud
WHERE nom BETWEEN 'D' AND 'M' ;

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

38

Intervalle avec BETWEEN (date)

- SELECT * FROM etud
WHERE date_n BETWEEN '83-02-01' AND '83-02-15' ;
- Ne pas oublier les '...'

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3

39

Caractère _ pour un caractère et un seul

- SELECT * FROM etud
WHERE nom LIKE 'Dupon_';
- Attention LIKE au lieu de =

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3
Dupontel	Anne				
Mattel	Laurence				

40

Caractère % pour 0, 1 ou n caractère(s)

- SELECT * FROM etud
WHERE nom LIKE 'Dupon%';
- Attention LIKE au lieu de =

nom	prenom	dateN	note	dep	licence
Martin	Véra	85-10-31	13.5	77	2
Martin	Annie	85-12-31	15.5	75	1
Dupont	Sylvie	83-02-03	15.0	77	2
Martin	Annie	83-10-22	05.7	93	1
Dupond	Laurent			92	2
Lefèvre	Laurent		11.5		3
Dupontel	Anne				
Mattel	Laurence				

41

Exemples de filtres avec LIKE

- ... WHERE prenom LIKE 'Ann_';
-> Anne, Anna mais pas Annie, Annette
- ... WHERE prenom LIKE 'Laure _ _ _';
-> Laurence, Laurette mais pas Laurent, Laure
- ... WHERE prenom LIKE 'Ann%';
-> Ann, Anne, Annie, Annette...
- ... WHERE prenom LIKE '%e';
-> Laurence, Sylvie, Anne, Annie,...

42

Caractères Joker et valeur Null

Table de vérité du NULL
NOT NULL -> NULL

- SELECT * FROM etud WHERE dep LIKE '%';
-> Les NULL ne sont pas sélectionnés
-> Les valeurs vides sont affichées
- SELECT * FROM etud WHERE dep NOT LIKE '%';
-> Les NULL ne sont pas sélectionnés
-> Les valeurs vides ne sont pas sélectionnés
- SELECT * FROM etud WHERE dep IS NULL ;
-> Les NULL sont sélectionnés
-> Les valeurs vides ne sont pas sélectionnés

43 Opérateurs

Opérateurs arithmétiques	+ - * / % (MOD)
Opérateurs de comparaison	
égalité / non égalité	= <> !=
supériorité / infériorité	> < >= <=
appartenance à une liste de valeurs	IN ('val1','val2','val3')
intervalle de valeurs	BETWEEN val1 AND val2
motif de chaîne	LIKE % _
valeur Null	IS NULL IS NOT NULL
Opérateurs logiques	
et non ou ou exclusif	&& ! XOR
Connecteurs logiques	AND OR NOT
Gestion de la priorité	()
Caractère d'échappement (avant ' " %)	\

44 Ordre de priorité des opérateurs

NOT
- (négatif)
^
* / % MOD
+ -
= <> != > < >= <= IS LIKE REGEXP IN
BETWEEN
AND &&
OR

45 Concaténation : CONCAT(exp1, exp2, exp3)

- SELECT nom, prenom, CONCAT(nom, ' ', prenom) FROM etud ;

nom	prenom	CONCAT(nom, ' ', prenom)
Martin	Véra	Martin Véra
Martin	Annie	Martin Annie
Dupont	Sylvie	Dupont Sylvie
Martin	Annie	Martin Annie
Dupond	Laurent	Dupond Laurent
Lefèvre	Laurent	Lefèvre Laurent

46 Fonction de concaténation et alias

- SELECT nom, prénom, CONCAT(nom, ' ', prénom) as 'Nom complet' FROM etud ;

nom	prenom	Nom complet
Martin	Véra	Martin Véra
Martin	Annie	Martin Annie
Dupont	Sylvie	Dupont Sylvie
Martin	Annie	Martin Annie
Dupond	Laurent	Dupond Laurent
Lefèvre	Laurent	Lefèvre Laurent

47 Exemples avec fonction et calcul

- SELECT upper(nom), prenom, note, note*2 FROM etud;

upper(nom)	prenom	note	note*2
MARTIN	Véra	13.5	27.0
MARTIN	Annie	15.5	31.0
DUPONT	Sylvie	15.0	30.0
MARTIN	Annie	05.7	11.4
DUPOND	Laurent		
LEFÈVRE	Laurent	11.5	23.0

mise en majuscule

pour obtenir une note sur 40

- Autres exemples de champs calculés :
 - SELECT salaire + prime ...
 - SELECT prix * 19.6 / 100
 - SELECT prix - remise

48

- Fin de la première partie.
- SQL: l'ordre SELECT simple.