

# Le langage SQL

05 février 2007

Florence Petit

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

## 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



## NULL : absence de valeur

- Valeur inconnue  
ex : note, date de naissance...
- Valeur non pertinente  
ex : nom de jeune fille pour un garçon
- NULL ≠ zéro, NULL ≠ 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

8



## 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 ;

9



## Syntaxe du SQL

- Séparateur d'instructions : point virgule
- Séparateur de mots-clés : espace, tab, fin de ligne
- 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 '' ou "" dans les expressions
- Des commentaires sont possibles sous plusieurs formes :  
/\* commentaire sur plusieurs lignes \*/  
-- espace puis commentaire sur une ligne  
# commentaire sur une ligne

10



## Création de table avec CREATE

- Exemple  
CREATE TABLE etud (  
  num INT UNSIGNED NOT NULL PRIMARY KEY,  
  nom VARCHAR(30) NOT NULL,  
  prenom VARCHAR(25) NOT NULL,  
  dateN DATE,  
  note DECIMAL(3,1) UNSIGNED ZEROFILL,  
  dep CHAR(2),  
  licence TINYINT(1)  
);

11



## Suppression de table avec DROP

- Syntaxe générale  
DROP TABLE *nom\_table* ;

12



## Suppression d'enregistrements : DELETE

- Syntaxe générale  
DELETE FROM *nom\_table*  
  WHERE *condition* ;
- Suppression de toutes les lignes de la table (-> table vide) !  
DELETE FROM etud ;
- Suppression des lignes d'étudiants de licence 3  
DELETE FROM etud  
  WHERE licence=3 ;
- Suppression des lignes d'étudiants nommés Jean Dupont  
DELETE FROM etud  
  WHERE nom='Dupont' AND prenom='Jean' ;

13

### Insertion d'enregistrements : INSERT

- Syntaxe générale n°1  
`INSERT INTO nom_table  
VALUES (val1, val2,...,val7) ;`
- Exemple de syntaxe n°1  
`INSERT INTO etud  
VALUES (7,'Martin','Véra', '1988-01-01',13.5,75,1) ;`
- Syntaxe générale n°2  
`INSERT INTO nom_table (champ3, champ2,champ7)  
VALUES (val3, val2, val7) ;`
- Exemple de syntaxe n°2  
`INSERT INTO etud (licence, nom, prenom )  
VALUES (1, 'Martin', 'Véra') ;`

14

### Mise à jour : UPDATE... SET

- Changer une note :  
`UPDATE etud SET note=18  
WHERE num=6;`
- Changer le prénom :  
`UPDATE etud SET prenom='Véro'  
WHERE nom='Martin' AND prenom='Véra' ;`
- Changer le prénom et le département :  
`UPDATE etud SET prenom='Véro', dep=NULL  
WHERE num=4 ;`
- Augmenter toutes les notes de 2 :  
`UPDATE etud SET note=note+2 ;`

15

### 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

16

### 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;`

champ1	champ2	champ3	champ4	champ5
...				

  

champ4	champ3	champ1
...		

17

### 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];`

18

### 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

19

### 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

20

### Alias de noms de colonnes

- 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;

Prénom	Nom de famille
...	...

21

### Caractère d'échappement

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

22

### Gérer les doublons : ALL (facultatif) et DISTINCT

SELECT \* FROM etud ;

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

SELECT dep FROM etud ;

dep
77
75
92
77
75
null
93
77
92
null

SELECT DISTINCT dep FROM etud ;

dep
77
75
92
null
93

23

### Gérer les doublons : DISTINCT s'applique par ligne

SELECT ALL dep,licence FROM etud;

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

SELECT DISTINCT dep,licence FROM etud;

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

24

### 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	Dupond	Sylvie	83-02-03	15.0	77	2
2	Martin	Annie	85-12-31	15.5	75	1

25

### 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

26

### 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

27

### 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;

28

### 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

29

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

2 enregistrements

30

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

- 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

31

### 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

32

### 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="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

33

### 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

34

### 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

35

### 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

36

### Sélection et valeur Null

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

37

### Tables de vérité à 3 valeurs (dont NULL)

E	NOT
V	F
F	V
N	N

E1	E2	AND
V	V	V
V	F	F
F	F	F
V	N	N
F	N	F
N	N	N

E1	E2	OR
V	V	V
V	F	V
F	F	F
V	N	V
F	N	N
N	N	N

### Opérations algébriques avec NULL

E	Calcul sur E : +-*/
N	N

38

### 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	

39

### 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 >= (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	

40

### 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

41

### Intervalle avec BETWEEN (nombre)

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

équivalent à :

- SELECT \* FROM etud  
 WHERE note >= 11.5 AND note <=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)

42

### 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

43

### 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

44

### 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				

45

### Caractère % pour 0, 1 ou n caractères(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				

46

### 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,...

47

### Sélection avec 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

48

### Opérateurs

<b>Opérateurs arithmétiques</b>	+ - * / % (MOD)
<b>Opérateurs de comparaison</b>	
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
<b>Opérateurs logiques</b>	
et non ou ou exclusif	&& !    XOR
<b>Connecteurs logiques</b>	AND OR NOT
Gestion de la priorité	( )
Caractère d'échappement (avant ' " %)	\

49

### Ordre de priorité des opérateurs

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

50

### 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

51

### Fonction de concaténation et alias

- SELECT nom, prenom, CONCAT(nom, ' ', prenom) 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

52

### 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

pour obtenir une note sur 40

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

53

### Fonctions de groupe (d'agrégat ou d'ensemble)

- COUNT** nombre d'enregistrement
  - nombre d'étudiants total
  - nombre d'étudiant dont la note est connue
- SUM** somme
  - somme des prix dans une facture
- MAX** maximum
  - liste des étudiants qui ont la meilleure note
- MIN** minimum
  - liste des étudiants qui ont la note la plus basse
- AVG** moyenne (average) pour des nombres
  - moyenne des notes de licence 1

54

### Que comptent les fonctions d'agrégation ?

- SELECT COUNT(\*) FROM etud ;
- SELECT COUNT(nom) FROM etud ;
- SELECT COUNT(dep) FROM etudiant ;
- SELECT COUNT(DISTINCT dep) FROM etud ;

COUNT(*)
6

COUNT(nom)
6

COUNT(dep)
5

COUNT(DISTINCT dep)
4

- Compte les enregistrements (même Null)
- Nb d'enregistrements de nom "non Null"
- Nb d'enregistrements de département "non Null"
- Les doublons ne sont comptés qu'une fois

55

### Alias pour les noms de champs : AS

- SELECT COUNT(\*) FROM etud ;
- SELECT COUNT(\*) AS 'Nb d'étudiants' FROM etud ;  
-> s'il y a des espaces, entourer de '...' ou "..."
- SELECT dep AS Département, nom AS Nom, prenom AS Prénom FROM etud ;
- AS est facultatif (selon les implémentations)  
SELECT dep Département, nom Nom, prenom Prénom FROM etud ;

COUNT(*)	
	6

Nb d'étudiants	
	6

Département	Nom	Prénom
...	...	....

56

### Erreur d'utilisation des fonctions d'agrégation

- SELECT AVG(note) FROM etud ;  
- Moyenne des notes (non NULL)
- SELECT MIN(note), AVG(note), MAX(note) FROM etud ;

AVG(note)
12.24

MIN(note)	AVG(note)	MAX(note)
5.7	12.24	15.5

Incorrect :

- ~~SELECT nom, MIN(note) FROM etud ;~~  
- nom, note des étudiants ayant la note la plus basse
- ~~- pas de mélange entre Agrégat et Champ !~~

57

### Groupements de données : GROUP BY

Pour la lisibilité de la réponse

- SELECT dep, count(\*) FROM etud GROUP BY dep ;

nom	prenom	date_n	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

dep	count(*)
null	1
75	1
77	2
92	1
93	1

58

### Groupements de données : exercices

- Par année de licence, présenter par moyenne croissante, le nombre d'étudiants notés, la moyenne, le min, le max.
- SELECT licence, COUNT(note) AS Nb, AVG(note) AS Moyenne, MIN(note) AS Min, MAX(note) AS Max FROM etud GROUP BY licence ORDER BY Moyenne ;

licence	Nb	Moyenne	Min	Max
1	2	10.60000	05.7	15.5
3	1	11.50000	11.5	11.5
2	2	14.25000	13.5	15.0

59

### Erreur d'utilisation du GROUP BY

- ~~SELECT nom, prenom FROM etud GROUP BY dep ;~~
- ~~SELECT \* FROM etud GROUP BY dep ;~~

nom	prenom	date_n	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

60

### Condition sur le regroupement : HAVING

- Ressemble au WHERE mais appliqué au regroupement
- Départements où résident plus de 20 étudiants
- SELECT licence, count(num) FROM etud GROUP BY dep HAVING count(num) > 20
- S'il y a une clause WHERE, elle sera appliquée avant le calcul d'agrégat.

dep	count(num)
75	500
76	10
77	612

61

## Méthodologie d'écriture de requête SELECT

- Liste ordonnée des champs à afficher  
-> SELECT champ1, champ2, fonction de groupe  
Préfixer les champs dont le nom est identique dans 2 tables
- Liste des tables dont on cherche les champs  
Jointure nécessaire ?  
-> FROM table1, table2 WHERE condition de jointure  
-> FROM table1 JOIN table2 ON condition de jointure
- Conditions de recherche sur les enregistrements ?  
-> WHERE condition
- Fonction de groupe ?  
-> GROUP BY ..., ...
- Conditions de recherche sur les groupes  
-> HAVING condition
- Ordre de présentation des enregistrements ?  
-> ORDER BY champ1, alias2, (alias si fonction d'agrégat)
- Ajouter des alias si nécessaire (présentation, agrégat, autojointure...)

62

## Quelques fonctions

- ROUND(nb,n) -> arrondi à n décimales
  - ROUND(24.16, 1) = 24.2
  - ROUND(24.16, 0) = ROUND(24.16) = 24
- RAND(nb) -> nb aléatoire de 0 à 1 (plafonné à 0,n)
- UPPER(col) UCASE(col) -> chaîne en majuscule
- LOWER(col) LCASE(col) -> chaîne en minuscule
- CONCAT(exp1,exp2...) -> concaténation
- LENGTH(col) -> Nb de caractères
- SUBSTRING(col,n,p) -> sous-chaîne (du n<sup>ème</sup>, Lg=p)

64

## Récupération de date/temps

- CURRENT\_DATE CURRENT\_DATE() CURDATE() -> AAAA-MM-JJ
- CURRENT\_TIME CURRENT\_TIME() CURTIME() -> HH-MM-SS
- NOW() -> AAAA-MM-JJ HH-MM-SS
- YEAR(date) -> année (4 chiffres)
- MONTH(date) -> mois (chiffre)
- DAYOFMONTH() -> 26 (n<sup>ème</sup> jour du mois)
- DAYOFYEAR() -> 310 (n<sup>ème</sup> jour de l'année)
- EXTRACT(exp FROM date) -> exp=month, year...
- DATE\_FORMAT(date, format) -> date avec formatage demandé
- TO\_DAYS(date) -> nb jours écoulés entre date et an 0
- FROM\_DAYS(nb) -> inverse de To\_DAYS(), retourne la date
- UNIX\_TIMESTAMP(date) -> nb jours écoulés entre date et 1970-01-01

65

## Motifs de format pour DATE\_FORMAT(date, 'format')

- %y an (2 chiffres) : 97 %Y an (4 chiffres) : 1997
- %m mois (01 à 12) : 04 %M mois (long) : April
- %b mois (court) : Apr %e jour (1 à 31) : 8
- %d jour (01 à 31) : 08 %W jour (long) : Saturday
- %a jour (court) : Sat %w jour de semaine (Di=0, Lu=6)
- %j N°jour (1 à 365)
- %U semaine de l'année (début=di) %u idem (début=lu)
- %r heure (12h) : 11:59:30 PM %T heure (24h) : 23:59:30
- %l heure (12h) : 11 %H heure (24h) : 23
- %p AM ou PM
- %i minutes (2 chiffres) : 51
- %S secondes (2 chiffres) : 53

66

## Copie totale/partielle de table avec CREATE

- Syntaxe générale  
CREATE TABLE table2  
AS requête ;
- Copie de la table des étudiants de licence  
CREATE TABLE etud\_copie  
AS SELECT \* FROM etud ;
- Copie partielle de table : uniquement les étudiants de licence 1  
CREATE TABLE etud\_lic1  
AS SELECT nom, prenom, date\_n, note  
FROM etud  
WHERE licence=1 ;

67

## Notation pointée des champs

- Les champs peuvent être préfixés pour le nom de la table ou de son alias
  - SELECT etud.nom FROM etud WHERE etud.note>10 ;
  - SELECT e.nom FROM etud e WHERE e.note>10 ;
- Notation obligatoire quand les noms sont identiques
  - dans les champ de jointure  
SELECT \* FROM etud, departement  
WHERE etud.dep=departement.dep ;
  - dans les champs à afficher  
SELECT nom, etud.dep FROM etud, departement  
WHERE etud.dep=departement.dep ;
  - dans une auto-jointure  
SELECT \* FROM etud e1, etud e2  
WHERE e1.num=e2.binome ;

68

### Exemple de BD à tables multiples

etud

nom	prenom	date_n	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

departement

dep_num	dep_nom
01	Ain
...	...
75	Paris
...	...
77	Seine-et-Marne
92	Hauts-de-Seine
93	Seine-Saint-Denis

69

### Jointure de table avec WHERE (SQL1)

- SELECT \* FROM etud, departement WHERE dep=dep\_num ; Ne pas oublier la clause WHERE
- SELECT \* FROM etud, departement WHERE etud.dep=departement.dep\_num ; Attention aux NULL
- SELECT \* FROM etud e, departement d WHERE e.dep=d.dep\_num ;

etud						departement	
nom	prenom	dateN	note	dep	licence	dep_num	dep_nom
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	75	Paris
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	93	Seine-Saint-Denis
Dupond	Laurent			92	2	92	Hauts-de-Seine
Lefèvre	Laurent	null	11.5	null	3	null	null
null	null	null	null	null	null	01	Ain

70

### Produit cartésien de table

nb enregistrements = 6 étudiants x n départements

- SELECT \* FROM etud, departement ;

nom	prenom	dateN	note	dep	licence	dep_num	dep_nom
Martin	Véra	85-10-31	13.5	77	2	01	Ain
Martin	Annie	85-12-31	15.5	75	1	01	Ain
Dupont	Sylvie	83-02-03	15.0	77	2	01	Ain
Martin	Annie	83-10-22	05.7	93	1	01	Ain
Dupond	Laurent			92	2	01	Ain
Lefèvre	Laurent		11.5		3	01	Ain
...	...	...	...	...	...	...	...
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	77	Seine-et-Marne
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	77	Seine-et-Marne
Dupond	Laurent			92	2	77	Seine-et-Marne
Lefèvre	Laurent		11.5		3	77	Seine-et-Marne
...	...	...	...	...	...	...	...

71

### Jointure avec JOIN (SQL2)

- Produit cartésien
  - SELECT \* FROM etud, departement ;
  - SELECT \* FROM etud CROSS JOIN departement ;
  - SELECT \* FROM etud JOIN departement ;
- Jointure
  - SELECT \* FROM etud, departement WHERE dep=dep\_num ;
  - SELECT \* FROM etud JOIN departement ON dep=dep\_num ;
- Jointure avec restriction
  - SELECT \* FROM etud, departement WHERE dep=dep\_num AND note > 10 ;
  - SELECT \* FROM etud JOIN departement ON dep=dep\_num WHERE dep=dep\_num AND note > 10 ;

72

### Jointure de table et sélection d'enregistrement

- SELECT \* FROM etud, departement WHERE dep=dep\_num AND (dep=77 or dep=75) AND licence=2 ; Ne pas oublier la clause WHERE de jointure

etud						departement	
nom	prenom	dateN	note	dep	licence	dep_num	dep_nom
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	75	Paris
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	93	Seine-Saint-Denis
Dupond	Laurent			92	2	92	Hauts-de-Seine
Lefèvre	Laurent		11.5		3		

73

### Jointure naturelle de table

- Jointure avec les colonnes communes de même nom
- SELECT \* FROM etud NATURAL JOIN departement
- SELECT \* FROM etud e, departement d WHERE e.dep=d.dep ;

etud						departement	
nom	prenom	dateN	note	dep	licence	dep	dep_nom
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	75	Paris
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	93	Seine-Saint-Denis
Dupond	Laurent			92	2	92	Hauts-de-Seine
Lefèvre	Laurent		11.5	null	3	null	

74

### Jointure externe (FULL | LEFT | RIGHT OUTER JOIN ...ON...)

- SELECT \* FROM etud LEFT OUTER JOIN departement ON dep=dep\_num ;

etud				departement			
nom	prenom	dateN	note	dep	licence	dep_num	dep_nom
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	75	Paris
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	93	Seine-Saint-Denis
Dupond	Laurent	null	null	92	2	92	Hauts-de-Seine
Lefèvre	Laurent	null	11.5	null	3	null	null
null	null	null	null	null	null	07	Ardèche

- par défaut, une jointure est une jointure interne (INNER JOIN)

75

### Jointure externe droite (RIGHT OUTER JOIN ...ON...)

- SELECT \* FROM etud RIGHT OUTER JOIN departement ON dep=dep\_num ;

etud				departement			
nom	prenom	dateN	note	dep	licence	dep_num	dep_nom
Martin	Véra	85-10-31	13.5	77	2	77	Seine-et-Marne
Martin	Annie	85-12-31	15.5	75	1	75	Paris
Dupont	Sylvie	83-02-03	15.0	77	2	77	Seine-et-Marne
Martin	Annie	83-10-22	05.7	93	1	93	Seine-Saint-Denis
Dupond	Laurent	null	null	92	2	92	Hauts-de-Seine
null	null	null	null	null	null	07	Ain
null	null	null	null	null	null	07	Ardèche
Lefèvre	Laurent	null	11.5	null	3	null	null

76

### Jointure externe et fonction d'agrégat

- SELECT dep\_num, count(nom) FROM etud RIGHT OUTER JOIN departement ON dep=dep\_num GROUP BY dep\_num ;

dep	count(*)
01	0
07	0
75	1
77	2
92	1
93	1

- SELECT dep\_num, count(nom) FROM etud, departement WHERE dep=dep\_num GROUP BY dep\_num ;

dep	count(*)
01	0
07	0
75	1
77	2
92	1
93	1

77

### Auto-jointure de table

- Jointure de la table sur elle-même
- Utilisation de 2 alias pour la même table
- SELECT \* FROM etud e1, etud e2 WHERE e1.num=e2.binome ;

e1				e2			
num	nom	prenom	binome	num	nom	prenom	binome
1	Martin	Véra	2	2	Martin	Annie	1
2	Martin	Annie	1	1	Martin	Véra	2
3	Dupont	Sylvie	4	4	Martin	Annie	3
4	Martin	Annie	3	3	Dupont	Sylvie	4
5	Dupond	Laurent	6	6	Lefèvre	Laurent	5
6	Lefèvre	Laurent	5	5	Dupond	Laurent	6

78

### Auto-jointure de table

- Etudiants dont la note est inférieure à la note de l'étudiant n°4
- SELECT e1.\* FROM etud e1, etud e2 WHERE e1.note < e2.note AND e2.num=4 ;

e1				e2			
num	nom	prenom	Note	num	nom	prenom	binome
1	Martin	Véra	13.5	2	Martin	Annie	15.5
2	Martin	Annie	15.5	1	Martin	Véra	13.5
3	Dupont	Sylvie	15.0	3	Dupont	Sylvie	15.0
4	Martin	Annie	05.7	4	Martin	Annie	05.7
5	Dupond	Laurent	null	5	Dupond	Laurent	null
6	Lefèvre	Laurent	13.5	6	Lefèvre	Laurent	13.5