Introduction aux Bases de données
Cours 4 : Introduction à SQL

UFR 919 – Licence

2e année

Plan

- Introduction
- Sélection et projection
- Tri des résultats
- Opérations ensemblistes
- Fonctions numériques, de caractères et de dates

SQL : Structured Query Language

- Langage d'interrogation pour les BD relationnelles
- Développé chez IBM (1970-80)
- Devenu une norme (ANSI/ISO) en 1986
- Malgré ça, implantations différentes selon SGBD
- Langage déclaratif (basé sur calcul relationnel de tuple)

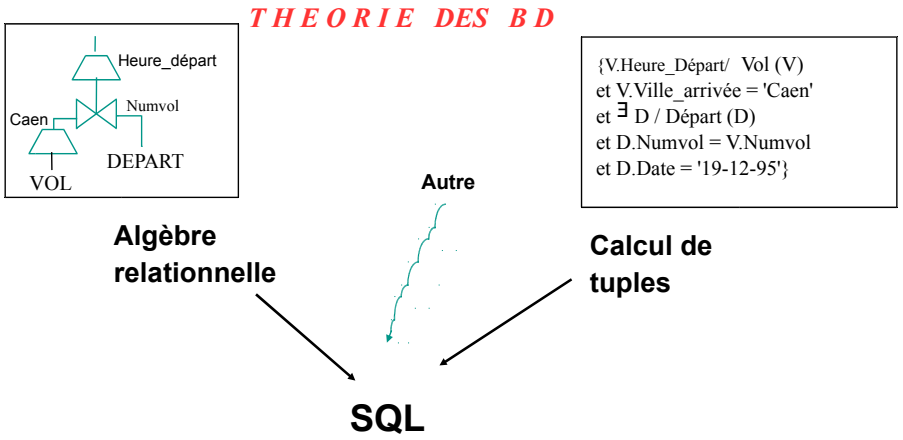
L'évolution des sta ndards SQL

- Début : SQL86
- SQL89 ou SQL1
- SQL92 ou SQL2
- SQL99 ou SQL3 (ajout récursivité, triggers, fonctions OO, types binaires,...)
- SQL2003 (ajout manipulation XML, auto-incrément...)
- SQL2008 (ajout des fonctions de fenêtrage, limite du nombre de résultats, ...)

Principaux rôles de SQL

1) Définir et modifier le schéma d'une BD
2) Manipuler les données (ajout, suppression, modification)
3) Interroger les données

D'où vient SQL ?



Syntaxe simplifée de SQL

SELECT liste-colonnes
FROM *Table*
[WHERE condition] ;

Retourne

- Les attributs de liste-colonnes
- Des enregistrements de la table *Table*
- qui vérifient condition

  La clause where est facultative mais très utile

Exemples

Schéma de la BD

Emp (Eno, Ename, Title, City)
Project(Pno, Pname, Budget, City)
Pay(Title, Salary)
Works(Eno, Pno, Resp, Dur)

Noms de tous les employés

Noms et budgets des projets

Remarques 1/2

- La clause from déclare les variables (calcul)
  - § Par défaut nom de la relation : from R, S
  - § on peut renommer : from R v1, S v2...
- Pour retourner toutes les colonnes
  - § Select *
- Sémantique « multi-ensembliste »:
  - § Possibilité d'avoir des doublons
    (parce que les éliminer coûte cher, parce qu'on peut vouloir les compter,...)
  - § Les éliminer avec le mot clé **distinct**
    select DISTINCT

Exemples

Schéma de la BD

Emp (Eno, Ename, Title, City)
Project(Pno, Pname, Budget, City)
Pay(Title, Salary)
Works(Eno, Pno, Resp, Dur)

Toutes les informations sur les employés

Toutes les villes où vivent des employés

L'ensemble des villes où vivent des employés

Remarques 2/2

- Possibilité d'exprimer des opérations arithmétiques
  - § (att1+att2, att*1.5, etc)
- Possibilité de retourner des chaines entre ' '
- Possibilité de préfixer les attributs par le nom de la table ou une variable
  - § Lever les ambiguïtés de noms d'attributs
- Possibilité de renommer une colonne dans le SELECT avec le mot-clé AS
  - § Lisibilité des résultats

Exemples

Schéma de la BD

Emp (Eno, Ename, Title, City)
Project(Pno, Pname, Budget, City)
Pay(Title, Salary)
Works(Eno, Pno, Resp, Dur)

Salaires mensuel par titre(considérer que Salary est pour un an)

Toutes les villes où vivent des employés

Noms et budgets des projets

Exemple

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

SELECT PNO, BUDGET
FROM PROJ :

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

SELECT PNAME FROM PROJ :

PNAME
Database Develop.
Instrumentation
CAD/CAM
Maintenance
CAD/CAM

SELECT DISTINCT PNAME
FROM PROJ :

PNAME
Maintenance
CAD/CAM
Database Develop.
Instrumentation

WHERE : Prédicats

Prédicats simples :

- Expression1 θ Expression2
  - où Expression1 peut être un attribut ou une expression arithmétique impliquant des attributs, $\theta \in \{<, >, =, <=, >=, <>\}$ et Expression2 une expression ou une valeur de domaine
- Exemples :
  - R.Name = 'J. Doe'
  - (S.Age + 30) >= 65
  - R.A = S.B

Prédicats composés :

- prédicats simples combinés avec les connecteurs logiques AND, OR, NOT

Exemple de sélection

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

SELECT * FROM EMP
WHERE TITLE = 'Elect. Eng.' ;

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E6	L. Chu	Elect. Eng.

Requêtes avec prédicats

Emp (Eno, Ename, Title, City)
Pay(Title, Salary)
Project(Pno, Pname, Budget, City)
Works(Eno, Pno, Resp, Dur)

Professions qui gagnent plus de 50 000 € par an ?

Numéros des managers d'un projet pendant plus de 17 mois?

IN, BETWEEN, LIKE

- Appartenance à un ensemble de valeurs :

  Att IN (Const1, Const2, ...)
- Appartenance à un intervalle de valeurs :

  Att BETWEEN Constante1 AND Constante2
- Ressemblance à un motif :

  Att LIKE 'MOTIF'

§ où MOTIF combine des chaînes et des joker
- % pour une chaîne quelconque (y compris vide)
- _ pour un caractère quelconque et un seul

Requêtes avec prédicats (2)

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Nom des projets de Paris, Lyon ou Nantes ?

Comment le faire sans IN ?

Nom des projets ayant un budget compris entre 5M et 10M euros?

Comment le faire sans BETWEEN ?

Requêtes avec prédicats (3)

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Nom des employés commençant pas C?

Nom des employés dont le 2ème numéro est un 5?

Nom des employés habitant une ville composé de 2 mots (ex : Chatenay Malabry)?

Valeurs nulles

- La valeur de certains attributs peut
  - ne pas être connue (ex. : année de construction du Louvre)
  - ou ne pas avoir de sens (ex. : nom de jeune fille pour un homme)

on parle alors de valeurs nulles (mot-clé NULL)

- NULL n'est pas une valeur mais une absence de valeur! Les opérations ou les comparaisons ne peuvent lui être appliqué
- Toute opération (+,-,/,*,substr, to_char,...) appliquée à NULL donne NULL
- Toute comparaison avec NULL donne ni vrai ni faux, mais INCONNU

Notions de sémantique Tri-Valuée abordées en cours 8 : compléments SQL

Syntaxe du tri

SELECT liste-colonnes
FROM nomtable
WHERE condition
ORDER BY liste-colonnes ;

- Dans la clause ORDER BY, on peut avoir des :
  - des noms de colonnes
  - des expressions avec noms de colonnes
  - des numéros de position des colonnes dans la clause SELECT.
- On précise le sens : ASC (par défaut) ou DESC
- Les valeurs nulles sont à la fin par ordre croissant, au début par ordre décroissant.

Exemple de tri

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms, budgets et villes des projets de budget supérieur à 250 000 euros, en ordonnant le résultat par ordre décroissant de budget puis par nom par ordre alphanumérique croissant ?

Exemple tri lexicographique

NUMPROJ	NOMPROJ	BUDGET
1	aa	20
2	bb	100
3	ab	30
4	aa	200

select * from project order by nomproj , budget desc;

NUMPROJ	NOMPROJ	BUDGET
4	aa	200
1	aa	20
3	ab	30
2	bb	100

Exemple de tri (2)

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms, budgets TTC (TVA 20%) et villes des projets, en ordonnant le résultat par ordre décroissant de budget TTC ?

Noms, budgets et villes des projets en ordonnant le résultat par ordre décroissant de budget TTC ?

Opérations ensemblistes

On peut réaliser des opérations ensemblistes sur les clauses SELECT.

3 opérations ensemblistes

UNION	union de deux ensembles
INTERSECT	intersection de deux ensembles
MINUS	différence de deux ensembles (norme : EXCEPT)

Principe

Pour les opérations ensemblistes :

- Pas de lien entre les objets sélectionnés dans les 2 requêtes
- Même schéma dans les SELECT des deux requêtes : c'est à dire même nombre d'attributs et chacun du même type (par forcément le même nom)
- Le schéma en sortie correspond au schéma de la première requête
- Par défaut, les opérations ensemblistes éliminent les doublons (ensemble). Pour garder les doublons (multi-ensemble), il faut ajouter ALL après l'opérateur : UNION ALL, EXCEPT ALL, INTERSECT ALL

UNION

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, Town)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms des villes où habitent des employés ou où sont localisés des projets?

INTERSECTION

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, Town)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms des villes où habitent des employés et où sont localisés des projets?

DIFFERENCE

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, Town)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms des villes où habitent des employés mais où n'est localisé aucun projet?

Fonctions

De nombreuses fonctions existent pour :

- Manipuler les dates
- Manipuler les chaînes de caractères
- Manipuler les chiffres

Cependant, bien qu'une norme existe, elles diffèrent souvent d'un SGBD à l'autre...

Quelques exemples suivent.

Les fonctions de date

Fonction(s)	Desc	Norme	Access	MySQL	Sql Srv	Oracle
Current_date	Date courante	O	N	O	N	N
Current_time	Heure courante	O	N	O	N	N
Getdate	Heure et date courante	N	N	N	O	N
Now	Heure et date courante	N	O	O	O	N
Sysdate	Date et heure courante	N	N	O	N	O
Day/month/year	Sélectionne le jour/mois/an	N	O	O	O	N
To_char(f1,f2)	Conversion de date ou numérique en string	N	N	N	N	O

Les fonctions sur chaînes de caractères

Fonction	Desc	Norme	Access	MySQL	Sql Srv	Oracle
Lower/Upper	Mise en minuscules/majusc	O	N	O	O	O
Substring	Extraction sous-chaîne	O	N	O	N	N
Substr	Extraction sous-chaîne	N	N	N	N	O
Position	Position d'une chaîne dans une autre	O	N	O	N	N
Locate	Position d'une chaîne dans une autre	N	O	O	O	O

Les fonctions numériques

Fonction	Desc	Norme	Access	MySQL	Sql Srv	Oracle
Abs	Valeur absolue	N	O	O	O	O
Ceiling	Valeur approchée haute	N	O	O	O	N
Ceil	Valeur approchée haute	N	N	N	N	O
Floor	Valeur approchée basse	N	O	O	O	O
Cos, sin, tan, exp, log,mod, power, sqrt	Opérations diverses	N	O	O	O	O

Fonctions sur dates (1)

Les dates ont un format de stockage optimisé et un format d'affichage/saisie par défaut qui dépend de la configuration du SGBD

- On peut afficher une date (attribut date) au format voulu :

  to_char(att_Date, 'masque')

- Ou saisir une date sans connaître le format de saisie par défaut :

  to_date('chaîne', 'masque')

A noter qu'on peut soustraire deux dates pour obtenir une durée, ou additionner une durée à une date pour obtenir une autre date

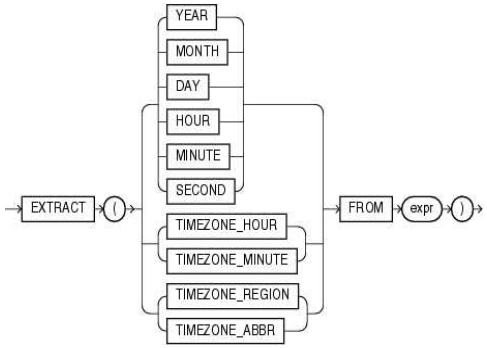
Fonctions sur dates (2)

Le masque se compose de :

- YEAR, YY, YYYY pour l'année en toutes lettres, sur 2 chiffres ou sur 4
- MONTH, MON, MM pour le mois en entier, abrégé, sur 2 chiffres
- DAY, DDD, DD, D pour le jour en lettre, le jour de l'année, du mois ou de la semaine
- HH, HH24, MI, SS pour heure sur 12H, sur 24H, les minutes, les secondes
- - ; / , _ , etc pour les séparateurs

Fonctions sur dates (3)

- La fonction Extract permet d'extraire un élément d'une date



SELECT EXTRACT(YEAR FROM DATE '1998-03-07') FROM DUAL;

EXTRACT(YEARFROMDATE'1998-03-07')

1998

Exemple

```
SELECT last_name, employee_id, hire_  date
   FROM employees
   WHERE EXTRACT(YEAR FROM
   TO_DATE(hire_date, 'DD-MON-RR')) > 1998
   ORDER BY hire_date;

LAST_NAME                 EMPLOYEE_ID HIRE_DATE
------------------------- ----------- ---------
Landry                            127 14-JAN-99
Lorentz                           107 07-FEB-99
Cabrio                            187 07-FEB-99
```


Exemples sur Oracle (1/3)

```
SELECT  TO_CHAR
     (SYSDATE, 'MM-DD-YYYY HH24:MI:SS')
 FROM DUAL;

SQL> SQL>   2    3
TO_CHAR(SYSDATE,'MM-DD-
YYYYHH24:MI:SS')
-------------------------------------------------------
02-19-2015 12:50:46
1 ligne sélectionnée
```


Exemples sur Oracle (2/3)

```
SELECT TO_DATE ('02-19-2015' )
FROM DUAL;

SQL> SQL>   2    3        ('02-19-2015' )
    *
ERREUR à la ligne 2 :
ORA-01843: ce n'est pas un mois valide
```


Exemples sur Oracle (3/3)

```
SELECT TO_DATE
     ('02-19-2015','MM-DD-YYYY' )
     FROM DUAL;

SQL> SQL>   2    3
TO_DATE('0
----------
19/02/2015

1 ligne sélectionnée.
```

Exemples de fonctions sur dates

Emp (Eno, Ename, Title, City)
Pay(Title, Salary)
Project(Pno, Pname, Budget, City, StartingDate)
Works(Eno, Pno, Resp, Dur)

Nom des projets ayant commencé avant le 1er janvier 2015?

```
Attention !! avec StartingDate < '01/01/2015' comparaison de
   chaines de caractères (ordre alphanumérique) et non de dates
```

Afficher tous les projets commencé ce mois ?

Trouver le jour (lundi, mardi, etc) de votre naissance ?

Exemple Oracle

Jour de votre date de naissance (1 juillet 1992)

```
SELECT TO_CHAR(TO_DATE('01-07-1992','DD-
MM-YYYY'), 'DAY-DD-MM-YYYY')
    FROM DUAL;

SQL> SQL>  2
TO_CHAR(TO_DATE('01-07-1992','DD-MM-
YYYY'),'DAY-DD-MM-YYY
----------------------------------------------------
MERCREDI-01-07-1992
1 ligne sélectionnée.
```