Bases de données

Souhila KACI

Partie 2

Le langage SQL

Présentation

- SQL : **Strucured Query Language** (Langage de requêtes structurées).
- SQL est un langage de requêtes basé sur l'algèbre relationnelle.
- SQL permet
  - la description du schéma de la base de données : création des tables, suppression des tables, création des contraintes d'intégrité.
  - la modification du contenu des tables : ajout, suppression, modification des tuples des tables.
  - l'interrogation sur le contenu des tables : projection, sélection, jointure, tri, agrégation.
- SQL est un langage normé (norme ANSI).
- Les éditeurs de SGBDR proposent diverses extensions à cette norme.

Ouverture d'un compte Oracle

- Aller sur le site https://sapiens.umontpellier.fr/ et s'authentifier
- Aller sur "mon compte $->$ "service"
- Choisir Oracle et saisir un mot de passe

Le langage SQL

- **Description/Manipulation du schéma de la base de données : Langage de Description des Données**
- Modification du contenu des tables : Langage de Modification des Données
- Interrogation sur le contenu des tables : Langage d'Interrogation des Données

Création de tables

CREATE TABLE nomTable (
nomColonne1 TypeColonne1 [ContrainteCol1],
nomColonne2 TypeColonne2 [ContrainteCol2],
nomColonne3 TypeColonne3 [ContrainteCol3],
...
[Contraintes de table]
);

Les types / Les domaines des données

- Chaîne de caractères :
  - CHAR(n) : chaîne de caractères de taille fixe (n caractères)
  - VARCHAR(n) : chaîne de caractères de taille variable (n est la taille maximale)
- Numériques :
  - INTEGER ou INT
  - REAL
- Temps :
  - DATE (année,mois,jour)
  - TIME (heures,minutes,secondes)

Création de tables – Type de données

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

CREATE TABLE Employes (
Num INT ...,
Nom VARCHAR(30) ...,
Prenom VARCHAR(30) ...,
Age INT ...,
NomService VARCHAR(30) ...,
IndiceSal INT ...
);

Création de tables – Contrainte Clé Primaire

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

CREATE TABLE Employes (
Num INT PRIMARY KEY,
Nom VARCHAR(30) ...,
Prenom VARCHAR(30) ...,
Age INT ...,
NomService VARCHAR(30) ...,
IndiceSal INT ...
);

ATTENTION

Possible seulement si la clé primaire est formée d'un seul attribut.

Création de tables – Contrainte Clé Primaire

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

CREATE TABLE Employes (
Num INT,
Nom VARCHAR(30) . . . ,
Prenom VARCHAR(30) . . . ,
Age INT . . . ,
NomService VARCHAR(30) . . . ,
IndiceSal INT . . . ,
PRIMARY KEY (Num)
);

Toujours possible, que la clé primaire soit formée d'un seul attribut ou plusieurs attributs.

Création de tables – Contrainte Clé Primaire

Commande–Produit		
RefCmd	RefPdt	QuantiteCmdee
100	99	450
100	75	500
264	32	250
405	38	50

CREATE TABLE Commande–Produit (
RefCmd INT,
RefPdt INT,
QuantiteCmdee INT,
PRIMARY KEY (RefCmd,RefPdt)
);

Création de tables – Contrainte Clé Primaire

- Une table doit toujours posséder une clé primaire (et une seule).
- Les valeurs des attributs composant une clé primaire ne peuvent pas prendre la valeur NULL.
- Il ne peut y avoir de doublons pour les tuples formés des valeurs des attributs d'une clé primaire.
- Possibilité d'indiquer qu'on ne veut pas de doublons pour un autre ensemble d'attributs que ceux de la clé primaire : la contrainte UNIQUE.
- UNIQUE peut autoriser des valeurs NULL.

Création de tables – Contrainte UNIQUE

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

CREATE TABLE Employes (
Num INT PRIMARY KEY,
Nom VARCHAR(30) . . . ,
Prenom VARCHAR(30) . . . ,
Age INT . . . ,
NomService VARCHAR(30) . . . ,
IndiceSal INT . . . ,
UNIQUE(Nom,Prenom)
);

Création de tables – Contrainte NOT NULL

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

```
CREATE TABLE Employes (
Num INT PRIMARY KEY,
Nom VARCHAR(30) NOT NULL,
Prenom VARCHAR(30) NOT NULL,
Age INT,
NomService VARCHAR(30) NOT NULL,
IndiceSal INT NOT NULL,
UNIQUE(Nom,Prenom)
);
```

Création de tables – Contrainte CHECK

- La contrainte CHECK permet de spécifier une propriété que doit posséder un attribut ou un ensemble d'attributs.
- Syntaxe : CHECK(propriété), avec propriété qui doit être une expression booléenne (qui vaut vrai ou faux en considérant un tuple).

L'âge doit être strictement compris entre 0 et 150 ans :
CHECK((age>0) AND (age<150))

ATTENTION

Si présence de NULL alors le CHECK est vérifié, comme si la propriété retourne true.

Création de tables – Contrainte CHECK

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

```
CREATE TABLE Employes (
Num INT PRIMARY KEY,
Nom VARCHAR(30) NOT NULL CHECK(CHAR_LENGTH(Nom)>3),
Prenom VARCHAR(30) NOT NULL,
Age INT CHECK((Age>0) AND (Age<150)),
NomService VARCHAR(30) NOT NULL,
IndiceSal INT NOT NULL,
UNIQUE(Nom,Prenom),
CHECK(NomService IN ('Informatique','Juridique','Comptabilite'))
);
```

Rappel

Une clé étrangère est un attribut dont les valeurs appartiennent aux valeurs d'une clé primaire d'une autre table.

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

Salaires	
Indice	Montant
128	1200
234	1800
350	3300
484	4700

Création de tables – Contrainte Clé Etrangère

Employes					
Num	Nom	Prenom	Age	NomService	IndiceSal
1	'Martin'	'Paul'	23	'Informatique'	234
2	'Durand'	'Sandrine'	NULL	'Juridique'	234
3	'Berthe'	'Jasmine'	34	'Informatique'	128
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'	234

Salaires	
Indice	Montant
128	1200
234	1800
350	3300
484	4700

CREATE TABLE Employes (
Num INT PRIMARY KEY,
Nom VARCHAR(30) NOT NULL CHECK(CHAR_LENGTH(Nom)>3),
Prenom VARCHAR(30) NOT NULL,
Age INT CHECK((Age>0) AND (Age<150)),
NomService VARCHAR(30) NOT NULL,
IndiceSal INT REFERENCES Salaires(Indice),
UNIQUE(Nom,Prenom),
CHECK(NomService IN ('Informatique','Juridique','Comptabilite'))
);

Création de tables - Valeur par défaut

- Rajouter le mot clé DEFAULT suivi d'une valeur :
  IndiceSal INT DEFAULT 128

- Valeur générée par incrémentation de 1 de la valeur précédente de l'attribut (uniquement pour les clés primaires) :
  Num INT AUTO_INCREMENT (MySQL)
  Num INT AUTOINCREMENT (SQLite)

Suppression de tables

- Suppression d'une table : DROP TABLE nomTable;
  DROP TABLE Employes;
- Suppression d'une table lorsqu'elle existe : DROP TABLE IF EXISTS nomTable;
  DROP TABLE IF EXISTS Employes;

Fichier de création d'une BD

```
DROP TABLE IF EXISTS nomTable4;
DROP TABLE IF EXISTS nomTable3;
DROP TABLE IF EXISTS nomTable2;
DROP TABLE IF EXISTS nomTable1;
CREATE TABLE nomTable1(...);
CREATE TABLE nomTable2(...);
CREATE TABLE nomTable3(...);
CREATE TABLE nomTable4(...);
```

- Description/Manipulation du schéma de la base de données : Langage de Description des Données
- **Modification du contenu des tables : Langage de Modification des Données**
- Interrogation sur le contenu des tables : Langage d'Interrogation des Données

Ajout de tuples dans une table

- Pour insérer un tuple complet :
  INSERT INTO Table VALUES (val1,. . . , valn);
- Pour insérer un tuple non complet ou avec des valeurs ne suivant pas l'ordre des attributs :
  INSERT INTO Table (liste_attributs) VALUES (val1,. . . , valn);
- Pour insérer des tuples résultant d'une requête interrogative :
  INSERT INTO Table
  SELECT . . .
  FROM . . .
  . . .

Ajout de tuples dans une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	NULL	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'

```
INSERT INTO Employes VALUES (5,'Samuel','Paul',45,'Informatique');
```

Ajout de tuples dans une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	NULL	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'
5	'Samuel'	'Paul'	45	'Informatique'

```
INSERT INTO Employes VALUES (5,'Samuel','Paul',45,'Informatique');
```

Ajout de tuples dans une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	NULL	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'
5	'Samuel'	'Paul'	45	'Informatique'

- INSERT INTO Employes VALUES (5,'Samuel','Paul',45,'Informatique');
- INSERT INTO Employes (Nom,Num,Prenom,NomService,Age) VALUES ('Samuel',5,'Paul','Informatique',45);

Ajout de tuples dans une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	NULL	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'
5	'Samuel'	'Paul'	NULL	'Informatique'

- INSERT INTO Employes (Num,Nom,Prenom,NomService) VALUES (5,'Samuel','Paul','Informatique');
- Insertion de la valeur NULL pour les valeurs des attributs manquant (ou d'une valeur par défaut si spécifié lors de la création de la table)

Ajout de tuples dans une table

- Pour insérer des tuples complets résultant d'une requête interrogative :
  INSERT INTO Table
  SELECT ...
  FROM ...
  ...
- Pour insérer des tuples non complets résultant d'une requête interrogative :
  INSERT INTO Table(liste d'attributs)
  SELECT ...
  FROM ...
  ...

Ajout de tuples dans une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	NULL	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	NULL	'Juridique'
5	'Samuel'	'Paul'	NULL	'Informatique'

Services	
NomService	NbEmp

```
INSERT INTO Services(NomService,NbEmp)
SELECT NomService,count(*)
FROM Employes
GROUP BY NomService;
```

Services	
NomService	NbEmp
'Informatique'	3
'Juridique'	2

Suppression de tuples d'une table

Suppression de tuples satisfaisant un critère

DELETE FROM Table
WHERE condition;

DELETE FROM Employes
WHERE Age>=30;

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
~~2~~	~~'Durand'~~	~~'Sandrine'~~	~~40~~	~~'Juridique'~~
~~3~~	~~'Berthe'~~	~~'Jasmine'~~	~~34~~	~~'Informatique'~~
4	'Schwind'	'Jean-Marc'	28	'Juridique'
~~5~~	~~'Samuel'~~	~~'Paul'~~	~~45~~	~~'Informatique'~~

Mise à jour des tuples d'une table

- UPDATE table
  SET Att1 = va11,Att2=val2,Att3=val3,. . . ;
- UPDATE table
  SET Att1 = va11,Att2=val2,Att3=val3,. . .
  WHERE condition;
- UPDATE table
  SET (Att1,Att2,Att3,. . .) = (SELECT . . .)
  WHERE condition;

Mise à jour des tuples d'une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	40	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	28	'Juridique'
5	'Samuel'	'Paul'	45	'Informatique'

```
UPDATE Employes
SET NomService='Info'
WHERE NomService='Informatique';
```

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Info'
2	'Durand'	'Sandrine'	40	'Juridique'
3	'Berthe'	'Jasmine'	34	'Info'
4	'Schwind'	'Jean-Marc'	28	'Juridique'
5	'Samuel'	'Paul'	45	'Info'

Mise à jour des tuples d'une table

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	23	'Informatique'
2	'Durand'	'Sandrine'	40	'Juridique'
3	'Berthe'	'Jasmine'	34	'Informatique'
4	'Schwind'	'Jean-Marc'	28	'Juridique'
5	'Samuel'	'Paul'	45	'Informatique'

```
UPDATE Employes
SET Age=Age*2;
```

Employes				
Num	Nom	Prenom	Age	NomService
1	'Martin'	'Paul'	46	'Informatique'
2	'Durand'	'Sandrine'	80	'Juridique'
3	'Berthe'	'Jasmine'	68	'Informatique'
4	'Schwind'	'Jean-Marc'	56	'Juridique'
5	'Samuel'	'Paul'	90	'Informatique'

- Description du schéma de la base de données : Langage de Description des Données.
- Modification du contenu des tables : Langage de Modification des Données.
- **Interrogation sur le contenu des tables : Langage d'Interrogation des Données.**

Requête interrogative

- Le résultat d'une requête interrogative sera toujours une table/une relation.
- Une requête interrogative (simple) est de la forme :
  SELECT ⟨**liste d'attribut(s)**⟩ : colonnes/attributs que l'on veut garder pour affichage
  FROM ⟨**liste de table(s)**⟩ : tables sur lesquelles porte la requête
- Par la suite nous verrons qu'une requête interrogative peut contenir d'autres clauses que les clauses SELECT et FROM.

La projection

- Afficher certaines colonnes d'une table.
- Les colonnes sélectionnées doivent obligatoirement appartenir à la table.

```
SELECT TABLE.att1, TABLE.att2 ...  FROM TABLE;
```

SELECT ProduitVrac.designation FROM ProduitVrac;

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	'sel'

$\Rightarrow$

designation
'sucre'
'poivre'
'sel'

Lorsqu'il n'y a pas d'ambiguïté pour la désignation d'un attribut il est possible d'omettre le nom de la table dans sa désignation :

SELECT designation FROM ProduitVrac;

Cas particulier

- Le caractère générique *
- Il sert à sélectionner toutes les colonnes pour la projection.

SELECT * FROM ProduitVrac;

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	'sel'

$\Rightarrow$

codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	'sel'

Doublons

- Par défaut les doublons ne sont pas supprimés.

```
SELECT codePV FROM ProduitCond;
```

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

codePV
'P01'
'P01'
'P01'
'P02'

Suppression des doublons

- Pour supprimer les doublons : Utiliser le mot clé **DISTINCT**

```
SELECT DISTINCT codePV FROM ProduitCond;
```

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

codePV
'P01'
'P02'

Suppression des doublons ...2

SELECT DISTINCT codePC,codePV FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

codePC	codePV
'C012'	'P01'
'C253'	'P01'
'C258'	'P01'
'C693'	'P02'

Un peu d'ordre

- Par défaut, le résultat des requêtes n'est pas ordonné.
- Heureusement, il est possible de choisir un ordre.
- On peut ordonner suivant plusieurs colonnes.
- Le mot clé : **ORDER BY**
- Pour chaque critère, le tri peut être ascendant ou descendant `DESC`.
- Par défaut il est ascendant.
- Le tri peut également être fait à partir d'une expression construite à partir d'attributs.

Un peu d'ordre ... 2

ORDER BY nom_col1 | num_col1 [DESC] ...

SELECT * FROM Commande ORDER BY DateCom;
SELECT * FROM Commande ORDER BY 4;

Commande			
numCom	codePC	qte	DateCom
'CD01'	'C012'	25	12/05/2002
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD05'	'C693'	71	09/01/2003

$\Rightarrow$

numCom	codePC	qte	DateCom
'CD01'	'C012'	25	12/05/2002
'CD04'	'C012'	11	02/06/2002
'CD03'	'C012'	4	03/07/2002
'CD02'	'C693'	14	09/11/2002
'CD05'	'C693'	71	09/01/2003

Un peu d'ordre ...3

SELECT * FROM Commande ORDER BY codePC,DateCom DESC;

Commande			
numCom	codePC	qte	DateCom
'CD01'	'C012'	25	12/05/2002
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD05'	'C693'	71	09/01/2003

⇒

numCom	codePC	qte	DateCom
'CD05'	'C693'	71	09/01/2003
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD01'	'C012'	25	12/05/2002

Un peu d'ordre ...4

SELECT * FROM Commande ORDER BY codePC ASC,DateCom DESC;

Commande			
numCom	codePC	qte	DateCom
'CD01'	'C012'	25	12/05/2002
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD05'	'C693'	71	09/01/2003

⇒

numCom	codePC	qte	DateCom
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD01'	'C012'	25	12/05/2002
'CD05'	'C693'	71	09/01/2003
'CD02'	'C693'	14	09/11/2002

La restriction - Sélection

- Permet l'affichage de certaines lignes, qui vérifient un critère donné.
- Le critère est une expression booléenne plus ou moins compliquée.
- Le mot clé WHERE

SELECT att1,att2 ... FROM table WHERE condition;

SELECT * FROM ProduitCond WHERE (codePV='P01');

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'

Opérateurs de comparaison

Opérateur	Numérique	Chaîne de caractères	Date/Heure
=	égal	identique	en même temps
<>	différent	différent	pas en même temps
>	supérieur	supérieure	après
<	inférieur	inférieure	avant
		...	

- Et les connecteurs logiques habituels : AND, OR, NOT.
- Attention aux priorités.
- Il en existe d'autres, nous les verrrons plus tard.

Un petit exercice (1)

Voiture	Immatriculation	Marque	Annee	Prix	IdProprio
	'1111AA01'	'Toyota'	1997	16 000	'Id01'
	'2222BB02'	'Peugeot'	2000	31 200	'Id01'
	'3333CC03'	'Fiat'	1997	2 000	'Id03'
	'4444DD13'	'Fiat'	1995	30 300	'Id02'
	'5555EE62'	'Renault'	1997	21 000	'Id02'
	'6666FF59'	'Opel'	1999	2 900	'Id01'
	'7777ZZ75'	'Ford'	1998	22 222	'Id03'

Personnes	IdProprio	Nom	Prenom	Naissance
	'Id01'	'Martin'	'Paul'	01/02/1967
	'Id02'	'Duval'	'Jean'	03/09/1980
	'Id03'	'Dupond'	'Laurence'	01/01/1945
	'Id04'	'Durand'	'Julie'	03/03/1985

Un petit exercice (2)

1. Liste des immatriculations.
2. Liste des voitures de 1996.
3. Voitures qui appartiennent au proprio Id01.
4. Liste des voitures entre 10000 et 20000 euros.
5. Différentes marques de voitures.

Renommage...1

- Pour des raisons de commodité ou de clarté, il est possible de renommer les colonnes de la table résultat.
- Le mot clé **AS**

```
SELECT codePC AS Code_Conditionné FROM Commande;
```

Commande			
numCom	codePC	quantite	DateCom
'CD01'	'C012'	25	12/05/2002
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD05'	'C693'	71	09/01/2003

⇒

Code_Conditionné
'C012'
'C693'
'C012'
'C012'
'C693'

Renommage...2

- Pour des raisons de commodité ou de clarté, il est possible de renommer les tables dans une requête.
- On utilise aussi le mot clé **AS**

SELECT codePC AS Code_Conditionné FROM Commande AS CMD where CMD.quantite>20;

Commande			
numCom	codePC	quantite	DateCom
'CD01'	'C012'	25	12/05/2002
'CD02'	'C693'	14	09/11/2002
'CD03'	'C012'	4	03/07/2002
'CD04'	'C012'	11	02/06/2002
'CD05'	'C693'	71	09/01/2003

$\Rightarrow$

Code_Conditionné
'C012'
'C693'

Attributs Calculés
Création de nouvelles colonnes

- On peut créer de nouvelles colonnes qui sont "construites" à partir de colonnes existantes.

SELECT Poids*0.9 AS PoidsNet FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

PoidsNet
2.07
0.9
1.8
1.8

Exemple

SELECT codePC,Poids,Volume,Volume/Poids AS Densité
FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

CodePC	Poids	Volume	Densité
'C012'	2.3	0.69	0.3
'C253'	1	0.25	0.25
'C258'	2	0.4	0.2
'C693'	2	0.45	0.22

Attributs calculés
Fonctions d'agrégation

- Obtention de valeurs agrégées sur une colonne
  - Somme `SUM`, moyenne `AVG`
  - Minimum `MIN`, Maximum `MAX`
  - Nombre de lignes `COUNT`

Exemple

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

- MIN(codePC) = 'C012'
- SUM(Poids) = 7.3
- Max(Volume) = 0.69
- Count(codePC) = 4

Syntaxe

- **SELECT SUM(c1),AVG(c2) ... FROM ... WHERE ...;**
- Ceci nous donne, pour un nombre quelconque de lignes, **une seule et unique valeur.**
- Nous obtenons un résultat agrégé à partir de l'ensemble des valeurs d'une colonne.

Exemple

SELECT SUM(Poids) FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

Sum(Poids)
7.3

Exemple

SELECT COUNT(*) AS nbP, AVG(Volume) AS volMoyen, MIN(Volume) AS volMin FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

nbP	VolMoyen	VolMin
4	0.45	0.25

Attention

- Le résultat d'un opérateur d'agrégation fournit une unique valeur.
- Une table à une seule ligne.

SELECT codePC, SUM(Poids) FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$ **IMPOSSIBLE**

Attributs calculés et Sélection

- Lorsqu'une requête combine des opérations de sélection et de calcul, la sélection est toujours faite **AVANT** le calcul.

SELECT SUM(Poids) FROM ProduitCond WHERE Volume $<$ 0.5;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01' ✓
'C258'	2	0.4	'P01' ✓
'C693'	2	0.45	'P02' ✓

$\Rightarrow$

Sum(Poids)
5

Exemple

SELECT SUM(Poids) AS pBrut, SUM(Poids)*0.9 AS pNet FROM ProduitCond WHERE Volume*2 < 1;

ProduitCond			
codePC	Poids	Volume	
'C012'	2.3	0.69	
'C253'	1	0.25	✓
'C258'	2	0.4	✓
'C693'	2	0.45	✓

$\Rightarrow$

pBrut	pNet
5	4.5

Opérateurs d'agrégation

Opérateur	Numérique	Chaîne	Date/Heure
Count	Nombre de valeurs connues		
MAX	Le plus grand	Le plus grand	Le plus tard
MIN	Le plus petit	Le plus petit	Le plus tôt
SUM	La somme	#####	#####
AVG	La moyenne	#####	#####

Remarque : COUNT

SELECT COUNT(*) as Nb FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

Nb
4

Remarque : COUNT

SELECT COUNT(Poids) as Nb FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

Nb
4

SELECT COUNT(DISTINCT Poids) as Nb FROM ProduitCond;

ProduitCond			
codePC	Poids	Volume	codePV
'C012'	2.3	0.69	'P01'
'C253'	1	0.25	'P01'
'C258'	2	0.4	'P01'
'C693'	2	0.45	'P02'

$\Rightarrow$

Nb
3

La valeur NULL

- Un attribut donné sous certaines conditions peut avoir la valeur NULL.
- Une valeur NULL correspond à l'absence d'une valeur.
- La valeur NULL n'est pas la chaîne de caractères vide ' ' ou la valeur 0 !
- Pour tester si la valeur d'un attribut A est NULL on ne doit pas effectuer le test A=NULL (cela retournera systématiquement faux, comme tout autre prédicat non spécifique à cette valeur).

La valeur NULL

- Pour tester si la valeur d'un attribut A est NULL on doit effectuer le test A IS NULL.
- Pour tester si la valeur d'un attribut A n'est pas NULL on peut effectuer le test A IS NOT NULL.
- Lors de l'utilisation de ORDER BY (ASC) les valeurs NULL sont présentées en premier (en dernier avec DESC).

La valeur NULL

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	NULL

SELECT * FROM ProduitVrac WHERE designation=NULL;
FAUX

SELECT * FROM ProduitVrac WHERE designation is NULL;

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	NULL

$\Rightarrow$

codePV	designation
'P03'	NULL

La valeur NULL

SELECT * FROM ProduitVrac WHERE designation is NOT NULL;

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	NULL

$\Rightarrow$

codePV	designation
'P01'	'sucre'
'P02'	'poivre'

La valeur NULL

Les fonctions d'agrégat COUNT( ), MIN( ), AVG( ) ne prennent pas en compte la valeur NULL.

```
SELECT COUNT(designation) FROM ProduitVrac;
```

ProduitVrac	
codePV	designation
'P01'	'sucre'
'P02'	'poivre'
'P03'	NULL

Résultat : 2

Un petit exercice (1)

Voiture	Immatriculation	Marque	Annee	Prix	IdProprio
	'1111AA01'	'Toyota'	1997	16 000	'Id01'
	'2222BB02'	'Peugeot'	2000	31 200	'Id01'
	'3333CC03'	'Fiat'	1997	2 000	'Id03'
	'4444DD13'	'Fiat'	1995	30 300	'Id02'
	'5555EE62'	'Renault'	1997	21 000	'Id02'
	'6666FF59'	'Opel'	1999	2 900	'Id01'
	'7777ZZ75'	'Ford'	1998	22 222	'Id03'

Personnes	IdProprio	Nom	Prenom	Naissance
	'Id01'	'Martin'	'Paul'	01/02/1967
	'Id02'	'Duval'	'Jean'	03/09/1980
	'Id03'	'Dupond'	'Laurence'	01/01/1945
	'Id04'	'Durand'	'Julie'	03/03/1985

Exercice (2)

1. Afficher le nombre total de voitures.
2. Afficher le nombre total de FIAT.
3. Afficher le prix moyen et le prix total de l'ensemble des voitures en francs et en euros.
4. Idem, mais uniquement pour les voitures de moins de 1996.
5. Afficher le nombre de voitures qui coûtent moins de 20 000 euros.
6. Afficher le nombre de voitures que possède Laurence Dupond.