Bases de données

Souhila KACI

Partie 5

Un peu de rappels

- Les requêtes que vous écrivez ne doivent **jamais** dépendre du contenu des tables mais uniquement de leur structure.
- Les produits conditionnés à base de sucre :

~~SELECT * FROM ProduitCond Where codePV="P01";~~

- Où parle t-on de P01 dans la requête ? Et si on change la clé primaire du sucre, il faut réécrire la requête...
- Il faut utiliser une jointure.

```
SELECT ProduitCond.* From ProduitCond,ProduitVrac WHERE
(ProduitVrac.codePV = ProduitCond.codePV AND
Designation="sucre");
```

Rappels : Les jointures

- **N'oubliez pas les conditions de jointure.**
- Autrement les résultats sont totalement incohérents.
- Le plus simple est de garder dans un coin de votre écran le schéma de relation.

Rappels : couple....

- Les requêtes du style *Les couples ...* nécessitent deux fois la même table.
- Pensez à **renommer** celles-ci.
- Les couples de commandes du même produit.

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

⇒

numCom	numCom
'CD03'	'CD01'
'CD04'	'CD01'
'CD04'	'CD03'
'CD05'	'CD02'

```
SELECT c1.numCom,c2.numCom FROM Commande AS
c1,Commande AS c2 WHERE c1.codePC=c2.codePC AND
c1.numCom>c2.numCom;
```

Rappels : Les fonctions d'agrégation

- Les fonctions d'agrégation ne peuvent prendre qu'un champ (ou un champ calculé) comme paramètre.
- Interdit : SELECT MAX(SELECT ...)
- Les requêtes comportant des fonctions d'agrégation ne retournent qu'une seule ligne.
- **On ne peut donc pas afficher d'autres données que des fonctions d'agrégation.**

Somme des poids, le plus lourd...

```
SELECT SUM(poids),Max(poids) FROM ProduitCond;
```

Rappels : Les groupements

- Requête du style : *pour chaque produit ...*
- On veut utiliser une fonction d'agrégation en même temps que des données de la table.
- Il faut utiliser GROUP BY.

Afficher pour chaque codePV, le nombre de références de produits conditionnés qui lui correspondent.

Fonctionnement : 1ere étape

- Les lignes sont regroupées par codePV identiques.

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'

Fonctionnement : 2eme étape

- Chaque ensemble de lignes est regroupé en une seule.
- **Les attributs codePC, Poids, Volume ne peuvent pas être conservés.**

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'

Fonctionnement : 3eme étape

- On ajoute la colonne COUNT(*) qui nous indique, pour chaque codePV, le nombre de lignes qui ont été regroupées.

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	NbP
'P01'	3
'P02'	1

```
SELECT codePV, count(*) AS nbProduit FROM ProduitCond
GROUP BY codePV;
```

Groupement et Condition

- Une condition avec WHERE est opérée avant le groupement :
  - Utilisée pour les jointures.
  - Utilisée si on veut supprimer des lignes avant le groupement : Le nombre de fois où le sucre est conditionné.
- Si vous voulez supprimer des lignes calculées après le groupement il faut utiliser le mot clé HAVING : Les produits vrac conditionnés plus de 2 fois.

Les sous-requêtes

- Utile pour simuler l'opérateur MINUS :
  - Les produits vrac qui ne sont jamais conditionnés.
- Utile lorsque l'on veut connaitre des lignes vérifiant une fonction d'agrégation donnée :
  - Le produit conditionné le plus lourd.
  - Le produit vrac le plus souvent conditionné.

Derniers rappels : Les différentes étapes

- De quelles tables ai-je besoin pour réaliser ma requête ?
  - en déduire les conditions de jointures
- Quelles sont les conditions ?
- Fonction d'agrégation ?
- Groupement ?
  - Si oui, y a t-il des conditions avant/après le groupement ?
- Opérateur MINUS ?
- Besoin de sous-requêtes ?
- La sous-requête peut-elle renvoyer plusieurs valeurs ?
- Que dois-je afficher ?
- Dans quel ordre ?

Exercice : Les derniers

- Les voitures par ordre de prix croissant.
- Les couples de propriétaires nés la même année.
- La valeur totale des véhicules de chaque propriétaire (avec leur nom).
- La valeur totale des véhicules de plus de 3000 euros de chaque propriétaire.
- La valeur totale des véhicules de chaque propriétaire ayant plus de 3 voitures (avec leur nom).
- Les personnes dont toutes les voitures sont des FIAT.
- Les personnes n'ayant pas de voiture.
- La personne qui possède la voiture la plus chère.
- La personne qui possède le plus de voitures.

- **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
  - DECIMAL(p,s), REAL
    p : chiffres avant la virgule, s : chiffres après la virgule
- 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(...);
```

- 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 manquants (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'