

Exemple Introductif

Cette série de transparents contient les notions utiles pour réaliser le TP 1

Sommaire

- Base de données : quelques définitions
- Création d'une table avec la commande CREATE TABLE
- Type d'une colonne
- Contraintes
- Contraintes d'unicité (UNIQUE), NOT NULL et de vérification (CHECK)
- Contrainte de clé primaire (PRIMARY KEY)
- Contrainte de clé externe (FOREIGN KEY)
- Insertion, suppression et modification de rangées dans la base de données
- Requête SQL (interrogation de la bases de données)
- Requête SQL portant sur plusieurs tables

Exemple : la base données CAPO

- La bases de données CAPO contient des données concernant des clients, des articles, des agents et des commandes.
- Il s'agit d'un exemple réduit et simplifié conçu dans un but purement pédagogique.
- Vous pouvez implanter cette base de données (Site du cours / Notes de cours / semaine 8).

BD, exemple introductif : CAPO

Products

pid	pname	city	quantity	price
p01	comb	Dallas	111400	0.5
p02	brush	Newark	203000	0.5
p03	razor	Duluth	150600	1
p04	pen	Duluth	125300	1
p05	pencil	Dallas	221400	1
p06	folder	Dallas	123100	2
p07	case	Newark	100500	1

Orders

ordno	month	cid	aid	pid	qty	dollars
1011	jan	c001	a01	p01	1000	450
1012	jan	c001	NULL	p01	1000	450
1019	feb	c001	a02	p02	400	180
1017	feb	c001	a06	p03	600	540
1018	feb	c001	NULL	p04	600	540
1023	mar	c001	a04	p05	500	450
1022	mar	c001	a05	p06	400	720
1025	apr	c001	a05	p07	800	720
1013	jan	c002	a03	p03	1000	880
1026	may	c002	a05	p03	800	704
1015	jan	c003	a03	p05	1200	1104
1014	jan	c003	a03	p05	1200	1104
1021	feb	c004	a06	p01	1000	460
1016	jan	c006	a01	p01	1000	500
1020	feb	c006	a03	p07	600	600
1024	mar	c006	a06	p01	800	400

Agents

aid	aname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6
a05	Otasi	Duluth	5
a06	Smith	Dallas6	5

Customers

cid	cname	city	discent
c001	TipTop	Duluth	10
c002	Basics	Dallas	12
c003	Allied	Dallas	8
c004	ACME	Duluth	8
c006	ACME	Kyoto	0

Base de données : structure

- Une *base de donnée relationnelle* est constituée d'un *ensemble de tables*.
- Une *table* est définie par
 - 1) *sa structure* (relativement fixe),
 - 2) *son contenu* (susceptible de varier en permanence).
- La structure d'une table est définie par un ensemble de *colonnes*. Chaque colonne possède un *type*.
- Une instruction CREATE TABLE permet en SQL de créer une table et de définir sa structure.

Base de données : manipulation

- Le contenu d'une table est constitué d'un ensemble de *lignes* (ou *rangées*). Le contenu peut être modifié par
 - l'ajout de nouvelles rangées (INSERT),
 - la suppression (DELETE) ou
 - la modification (UPDATE) des rangées existantes.
- Une *requête SQL* permet d'extraire des données appartenant à une ou plusieurs tables (SELECT).

Exemple

```
> CREATE TABLE Customers (  
  cid CHAR(4) PRIMARY KEY,  
  cname VARCHAR(20),  
  city VARCHAR(25),  
  discnt NUMBER CHECK (discnt BETWEEN 0.00 AND 15.00)  
);  
  
> INSERT INTO Customers VALUES ('c001', 'Tip Top', 'Duluth', 10.00) ;  
> INSERT INTO Customers VALUES ('c002', 'Basics', 'Dallas', 12.00) ;  
> INSERT INTO Customers VALUES ('c003', 'Allied', 'Dallas', 8.00) ;  
> INSERT INTO Customers VALUES ('c004', 'ACME', 'Duluth', 8.00) ;  
> INSERT INTO Customers VALUES ('c006', 'ACME', 'Kyoto', 0.00) ;  
  
> SELECT * FROM CUSTOMERS WHERE city = 'Dallas' ;
```

Création d'une table

```
CREATE TABLE Customers (  
  cid CHAR(4),  
  cname VARCHAR(20),  
  city VARCHAR(25),  
  discent NUMBER  
);
```

- L'instruction ci-dessus est légale en SQL. Cependant, elle autorise certaines anomalies graves :
 - Il est possible d'introduire deux clients avec le même code (cid).
 - Il est possible d'introduire un taux d'escompte négatif ou supérieur à 100 %.
- Il est possible d'interdire ces anomalies en spécifiant des *contraintes*.

Création d'une table

```
CREATE TABLE Customers (  
  cid CHAR(4) PRIMARY KEY,  
  cname VARCHAR(20),  
  city VARCHAR(25),  
  discnt NUMBER CHECK (discnt BETWEEN 0.00 AND 15.00)  
);
```

- La commande CREATE TABLE ci-dessus crée dans la base de données la table Customers avec :
 - Les colonnes cid, cname, city et discnt possédant les types indiqués ;
 - Une contrainte de clé primaire sur la colonne cid ;
 - Une contrainte de vérification (CHECK ...) sur la colonne discnt ;

Création d'une table

```
CREATE TABLE Agents (  
  aid CHAR(3),  
  aname VARCHAR(20),  
  city VARCHAR(25),  
  percent NUMBER CHECK (percent BETWEEN 0.00 AND 10.00),  
  PRIMARY KEY (aid)  
);
```

```
CREATE TABLE Products (  
  pid CHAR(3),  
  pname VARCHAR(20) NOT NULL,  
  city VARCHAR(25),  
  quantity NUMBER(7) CHECK (quantity >= 0),  
  price NUMBER CONSTRAINT check_price CHECK (price >= 0.00),  
  PRIMARY KEY (pid)  
);
```

Création d'une table

```
CREATE TABLE Orders (  
  ordno NUMBER(6) PRIMARY KEY,  
  month CHAR(3) CHECK (month IN ('jan', 'feb', 'mar', 'apr',  
    'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec')),  
  cid CHAR(4) NOT NULL REFERENCES Customers,  
  aid CHAR(3) REFERENCES Agents,  
  pid CHAR(3) NOT NULL REFERENCES Products,  
  qty NUMBER(5) CHECK (qty >= 0),  
  dollars NUMBER CHECK (dollars >= 0.00)  
);
```

- Par «**aid CHAR(3) REFERENCES Agents**», on crée une *contrainte de clé externe*
 - qui porte sur la colonne aid (de la table Orders) et
 - qui fait référence à la clé primaire (aid) de la table Agents.
- Des contraintes de clé externe sont aussi créées sur les colonnes cid et pid.

Type d'une colonne

- SQL possède notamment les types prédéfinis suivants :
 - CHAR(n) : chaîne de caractère de longueur n
 - VARCHAR(n) : chaîne de caractère de longueur variable (au plus n)
 - INTEGER : entier
 - DECIMAL (n, p) : décimal
 - FLOAT : flottant
 - DATE : date
- Dans toute rangée, la valeur attribuée à une colonne doit, bien sûr, appartenir au type de la colonne.
- Dans une rangée, la valeur attribuée à une colonne peut également prendre une valeur appelée NULL. La valeur NULL a différents usages. Elle peut servir notamment à indiquer une information manquante ou inconnue.

Notion de contrainte

- Une **contrainte** exprime une propriété que les rangées contenues dans la base de données doivent toujours vérifier.
- Par exemple, «CHECK (dollars \geq 0.00)» garantit que la colonne dollars d'une rangée quelconque de la table ORDERS ne prendra jamais de valeur strictement négative.
- Après chaque instruction de mise à jour, le système vérifie que toutes les contraintes sont respectées. Dans le cas contraire,
 - la mise à jour demandée est rejetée et
 - un message d'erreur est émis.
- Par exemple, il est impossible d'insérer dans la table ORDERS avec une valeur égale à -100 dans la colonne dollars.

Types de contraintes

- Il existe 5 types de contraintes :
 - Contrainte de clé primaire (Primary key, PK)
 - Contrainte d'unicité (Unique, UQ)
 - Différent de NULL (Not Null, NN)
 - Contrainte de vérification (Check, CK)
 - Contrainte de clé externe (Foreign Key, FK), ou de référence

Contraintes

- Une *contrainte d'unicité* (UNIQUE) porte sur une ou plusieurs colonnes d'une même table. Elle indique que deux rangées de la table ne peuvent pas posséder les mêmes valeurs pour toutes les colonnes concernées.
- Une *contrainte not null* (NOT NULL) porte sur une colonne. Elle indique qu'une rangée de la table ne peut pas avoir la valeur NULL pour la colonne concernée.
- Une *contrainte de vérification* (CHECK) porte sur une ou plusieurs colonnes d'une ou plusieurs tables et permet d'imposer une propriété portant sur les valeurs de ces colonnes.

Contrainte de clé primaire

- Pour une table, il existe au plus une clé primaire portant sur une ou plusieurs colonnes de la table.
- La contrainte de clé primaire impose
 - 1) qu'aucune des colonnes concernées ne peut prendre la valeur NULL ;
 - 2) que deux rangées de la table ne peuvent pas posséder les mêmes valeurs pour toutes les colonnes concernées (unicité).

Contrainte de clé primaire : Exemple

```
CREATE TABLE Orders (  
ordno NUMBER(6) PRIMARY KEY,  
...)
```

- Dans la table Orders, il existe une *contrainte de clé primaire* sur la colonne ordno (numéro de commande). La contrainte impose que :
 - 1) Pour toute rangée, la valeur de la colonne «ordno» est différente de NULL.
 - 2) Que deux rangées de la table ne peuvent pas avoir la même valeur pour la colonne ordno.
- **Remarque**
 - Grâce à ces propriétés, il est donc toujours possible d'identifier une rangée de la table Orders de manière unique à l'aide de la colonne «ordno».

Orders

ordno	month	cid	aid	pid	qty	dollars
1011	jan	c001	a01	p01	1000	450
1012	jan	c001	NULL	p01	1000	450
1019	feb	c001	a02	p02	400	180
1017	feb	c001	a06	p03	600	540
1018	feb	c001	NULL	p04	600	540
1023	mar	c001	a04	p05	500	450
1022	mar	c001	a05	p06	400	720
1025	apr	c001	a05	p07	800	720
1013	jan	c002	a03	p03	1000	880
1026	may	c002	a05	p03	800	704
1015	jan	c003	a03	p05	1200	1104
1014	jan	c003	a03	p05	1200	1104
1021	feb	c004	a06	p01	1000	460
1016	jan	c006	a01	p01	1000	500
1020	feb	c006	a03	p07	600	600
1024	mar	c006	a06	p01	800	400

Contrainte de clé externe

- Une *contrainte de clé externe*
 - *porte sur* une colonne A (ou une liste de colonnes) d'une table T et
 - *fait référence* à une colonne (ou une liste de colonnes), appelée colonne référencée, d'une autre table T' (la table *référéncée*).
- La contrainte impose, pour toute rangée de la table T, que
 - Soit la valeur de la colonne A vaut NULL,
 - soit la valeur de A existe parmi les valeurs de la colonne référencée.

Contrainte de clé externe : exemple

CREATE TABLE Orders (...
aid CHAR(3) REFERENCES Agents, ...)

- Dans Orders, il y a une *contrainte de clé externe* qui porte sur la colonne aid. La colonne référencée est la clé primaire de la table Agents (donc aid).
- Cette contrainte indique que, pour toute rangée de la table Orders :
 - Soit la valeur prise par la colonne aid vaut NULL,
 - soit cette valeur est présente parmi les valeurs de la colonne aid de la table Agents.
- Autrement dit : Le code agent mentionné dans la commande est soit valide (correspond à un agent existant), soit à NULL (commande sans agent), mais il ne peut pas être invalide.

Agents

aid	aname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6
a05	Otasi	Duluth	5
a06	Smith	Dallas	5

Orders

ordno	month	cid	aid	pid	qty	dollars
1011	jan	c001	a01	p01	1000	450
1012	jan	c001	NULL	p01	1000	450
1019	feb	c001	a02	p02	400	180
1017	feb	c001	a06	p03	600	540
1018	feb	c001	NULL	p04	600	540
1023	mar	c001	a04	p05	500	450
1022	mar	c001	a05	p06	400	720
1025	apr	c001	a05	p07	800	720
1013	jan	c002	a03	p03	1000	880
1026	may	c002	a05	p03	800	704
...						

Une contrainte de clé externe porte sur la colonne aid dans **Orders**.

La colonne référencée est la clé primaire (aid) de **Agents**.

Contrainte de clé externe : options

- La contrainte ON DELETE comporte différentes options. L'option choisie est définie lors de la création de la table. Le comportement du SGBD dépend de l'option choisie. En cas de suppression d'une rangée «*r*» dans la table mère, si la rangée «*r*» possède des rangées «filles» :
 - ON DELETE NO ACTION : la modification est rejetée
 - ON DELETE CASCADE : les rangées filles sont supprimées
 - ON DELETE SET NULL : pour chaque rangée fille, la clé externe prend la valeur NULL
 - ON DELETE SET DEFAULT (n'existe pas dans Oracle) : pour chaque rangée fille, la clé externe prend la valeur définie par défaut
- L'option choisie définit un comportement analogue dans le cas d'une mise à jour (UPDATE) ou d'une suppression (DELETE).

Insertion ou suppression de rangées

- On peut *insérer une rangée* dans une table avec la commande **INSERT**.
 - La commande insère une rangée ayant les valeurs indiquées.
- > INSERT INTO Customers VALUES ('c001', 'Tip Top', 'Duluth', 10.00) ;
- On peut *supprimer une ou plusieurs rangées* d'une table avec **DELETE**.
 - Une commande DELETE permet de spécifier une condition.
 - La commande supprime la (les) rangée(s) qui vérifie(nt) la condition indiquée.
- > DELETE FROM Customers
WHERE cid='c012' ;
- > DELETE FROM Orders
WHERE dollars <= 300 ;

Modification de rangées

- On peut *modifier une colonne dans une ou plusieurs rangées* d'une table avec la commande **UPDATE**.
- Une commande UPDATE permet de spécifier une condition.
- La commande effectue la modification demandée (SET ...) pour la (les) rangée(s) qui vérifie(nt) la condition indiquée.

> UPDATE Customers
SET discnt = 12.00
WHERE city='Toronto';

Requête SQL

- On peut *afficher les données d'une ou plusieurs tables* avec la commande SELECT (requête). La syntaxe de base est :

> SELECT <liste de colonnes affichées>
FROM <liste des tables concernées>
WHERE <condition>

- **Exemples**

> SELECT * FROM Customers ;

Cette commande affiche toutes les colonnes (*) pour toutes les rangées de la table Customers.

> SELECT cid, cname FROM Customers
WHERE city = 'Dallas';

Cette commande affiche les colonnes indiquées (cid et cname) pour les rangées de la table Customers qui vérifient la condition spécifiées – celles dont la colonne city vaut ‘Dallas’.

Requête SQL portant sur plusieurs tables

```
> SELECT ordno, dollars, O.cid, cname  
FROM Customers C, Orders O  
WHERE C.cid=O.cid  
AND dollars=>1000 ;
```

- La requête ci-dessus affiche le numéro de commande (ordno), le montant (dollars), le code (cid) et le nom du client (cname), pour toutes les commandes dont le montant atteint ou dépasse 1000 dollars.
- Noter que C et O sont utilisés pour remplacer le nom des tables Customers et Orders (alias).
- On dit que la requête effectue la *jointure* des tables Customers et Orders. «C.cid=O.cid» constitue la *condition de jointure*.
- Une rangée du résultat est produite pour chaque couple de rangées (C, O) qui vérifie la condition de jointure (C.cid=O.cid) et les autres conditions spécifiées (dollars=>1000).

TP 1

Lors du TP1, les étudiants seront amenés à expérimenter plusieurs concepts de base du langage SQL :

- Le comportement du système n'est pas toujours trivial, notamment en présence de contraintes de clés externes.
- Il faut bien comprendre quelles mises à jour risquent de violer des contraintes.
- Dans le cas des contraintes de clé externe toujours, il est possible de spécifier plusieurs comportements différents grâce à l'**option ON DELETE**. Ces différents comportements seront expérimentés.
- La présence de clés externes interdit la suppression de la table référencée, ce qui peut rendre problématique la suppression de certaines tables.

Interface du logiciel SQL*Plus Worksheet









