



Langage SQL

- Résumé

UNIVERSITÀ DI CORSICA
PASQUALE PAOLI

Requêtes de création de table

```
CREATE TABLE Nom_Table
(
  colonne1      type1,
  colonne2      type2,
  .....
  CONSTRAINT nom_constraint1
  type_constraint1,
  .....
);
```

Trois Types de Contraintes

- PRIMARY KEY
- FOREIGN KEY
- CHECK (condition)

Requêtes de création de table clé primaire mono-attribut

```
CREATE TABLE BATEAUX (
  BATNUM VARCHAR2(5) CONSTRAINT pk_bateaux
  PRIMARY KEY,
  BATNOM VARCHAR2(50),
  BATPORT VARCHAR2(50),
  CAPACITE NUMBER(4));
```

Définition des types syntaxiques

Définition de la clé primaire (contrainte colonne)

Requêtes de création de table clé primaire multi-attribut

```
CREATE TABLE PASSAGERS (
  CROISNUM VARCHAR2(5),
  NOMPASSAGER VARCHAR2(50),
  PRIX NUMBER(6),
  REMISE NUMBER(6),
  CONSTRAINT pk_passagers
  PRIMARY KEY ( CROISNUM, NOMPASSAGER) );
```

Définition d'une contrainte table (obligatoire pour les multi-attributs)

Requêtes de création de table Contrainte de vérification

```
CREATE TABLE SKIPPERS(
  SKNUM VARCHAR2(5) CONSTRAINT pk_skippers
  PRIMARY KEY,
  SKNOM VARCHAR2(50) NOT NULL,
  SKPORT VARCHAR2(50),
  SALAIRE NUMBER(6),
  CONSTRAINT ck_skippers_salaire CHECK(salaire >=
  1200) );
```

Définition d'une contrainte de vérification des valeurs

Requêtes de création de table Définition des clés étrangères

```
CREATE TABLE CROISIERS (
  CROISNUM VARCHAR2(5) CONSTRAINT pk_croisieres
  PRIMARY KEY,
  DEPPORT VARCHAR2(50),
  ARRPORT VARCHAR2(50),
  DEPDATE DATE,
  ARRDATE DATE,
  BATNUM VARCHAR2(5) REFERENCES BATEAUX (BATNUM),
  SKNUM VARCHAR2(5),
  CONSTRAINT fk_croisieres_sknun FOREIGN KEY (sknum)
  REFERENCES SKIPPERS(SKNUM) );
```

Définition d'une clé étrangère (contrainte de colonne)

Définition d'une clé étrangère (contrainte de table)

Résumé contraintes

NULL / NOT NULL : niveau colonne

UNIQUE (colonne1 [, colonne2] ...)

PRIMARY KEY (colonne1 [, colonne2] ...)

FOREIGN KEY (colonne1 [, colonne2] ...)

REFERENCES nomTableLieu (colonne1 [, colonne2] ...)

[ON DELETE {CASCADE | SET NULL}]

CHECK {condition}

Options lors des suppression

Contraintes définissables au niveau colonne (ligne de définition de l'attribut) ou au niveau table.

Requêtes de suppression/modification de table

DROP TABLE Nom-Table ;

Supprimer la table BATEAUX.

DROP TABLE BATEAUX;

Modification de table

Ajout d'un attribut

ALTER TABLE Nom-Table
ADD Attribut type;

Suppression d'un attribut

ALTER TABLE Nom-Table
DROP COLUMN Attribut;

Requêtes d'ajout/suppression de contrainte

Ajout d'une contrainte

ALTER TABLE Nom-Table
ADD CONSTRAINT nom_contrainte
.....définition;

Ajouter une contrainte spécifiant qu'il ne peut y avoir qu'un seul départ de croisière par jour pour chaque port

ALTER TABLE CROISIERES **ADD CONSTRAINT**
UK_DEPART **UNIQUE** (DEPART, DEPDATE);

Suppression d'une contrainte

ALTER TABLE Nom-Table
DROP CONSTRAINT nom_contrainte;

Requêtes de modification de table Définition des clés étrangères

Ajout des clés étrangères après la création de la table

ALTER TABLE croisiere **ADD CONSTRAINT** Cbatnum **FOREIGN KEY**
(batnum) **REFERENCES** bateau (batnum);

ALTER TABLE croisiere **ADD CONSTRAINT** Csknum **FOREIGN KEY**
(sknum) **REFERENCES** skipper (sknum);

Requêtes de modification de table Renommer une table

▪ **ALTER TABLE** ancienNom **RENAME** nouveauNom;

▪ Ou plus simplement

RENAME TABLE t1 TO newt1
[, t2 TO newt2] ...

Création d'un objet SEQUENCE

CREATE SEQUENCE nom

START WITH n

valeur initiale de la numérotation

INCREMENT BY p

pas d'incréméntation (positif ou négatif)

[**MINVALUE** min (ou **NOMINVALUE**)]

[**MAXVALUE** max (ou **NOMAXVALUE**)]

[**CYCLE** ou **NOCYCLE**]

[**CACHE** c ou **NOCACHE**];

anticipation des c valeurs suivantes en mémoire

Manipulation d'un objet SEQUENCE

- nomseq.**CURRVAL**
 - valeur courante de la séquence
- nomseq.**NEXTVAL**
 - incrémente la séquence et retourne la nouvelle valeur

```
CREATE SEQUENCE C_NOCLI START WITH 1000
MAXVALUE 9999 NOCYCLE;
```

```
INSERT INTO CLIENTS(NOCLI, NOMCLI)
VALUES (C_NOCLI.NEXTVAL, 'PAOLI');
```

Requêtes d'ajout de tuples



```
INSERT INTO Nom-Table (liste attributs)
VALUES
( expression1, expression2, ....);
```

```
INSERT INTO Nom-Table (liste attributs)
SELECT liste attributs
FROM Table
WHERE Condition ;
```

Requêtes de suppression et modification de tuples

```
DELETE FROM Nom-Table
WHERE Condition ;
```

```
UPDATE Nom-Table
SET Attribut1 = expression1 ,
    Attribut2 = expression2, ....
WHERE Condition ;
```

Requêtes de sélection/Projection

```
SELECT A1, A2, ...
FROM T
WHERE condition ,
```

Liste des attributs projetés
* Pour tous les attributs
distinct pour éliminer les répétitions

Table (Relation) concernée

Condition de sélection

- opérateurs de comparaison = < > >= <=
- connecteurs logiques AND, OR, NOT
- IN (ex: Ville IN ("AJACCIO", "BASTIA"))
- BETWEEN (ex: Salaire BETWEEN 1000 and 4000)
- LIKE (comparaison de chaînes)

> le symbole **_** remplace un caractère quelconque
> le symbole **%** remplace 0 ou plusieurs

Requêtes de tri

```
SELECT liste attributs
FROM liste tables
WHERE condition(s)
ORDER BY A1 DESC, A2 ASC, ... ;
```

Tri du résultat par ordre
Décroissant sur l'attribut A1

Tri du résultat par ordre
Croissant sur l'attribut A2

- Quels sont les caractéristiques des croisières classées par port de départ et pour un même port de départ par ordre décroissant de date de départ?

```
SELECT *
FROM CROISIERES
ORDER BY DEPPORT, DEPDATE DESC ;
```

Requêtes avec Calculs et Fonctions statistiques

```
SELECT attribut(s) ← Calculs, Fonctions statistiques
FROM table(s)
WHERE condition(s) sur les tuples;
```

Calculs

Expression sur des attributs: **SELECT** SALAIRE * 0,1

Fonctions Statistiques

- **SUM**(Attribut) : somme des valeurs de la colonne
- **AVG**(Attribut) : moyenne des valeurs de la colonne
- **MAX**(Attribut) : valeur maximale de la colonne
- **MIN**(Attribut) : valeur minimale de la colonne
- **COUNT**(Attribut) ou **COUNT**(*)
: nombre de lignes de la colonne
(ou de la table)

Requêtes de jointure

```
SELECT liste attributs
FROM T1, T2, ← Tables cibles
WHERE T1.A1 op. T2.A2 .....;
```

```
SELECT liste attributs
FROM T1 INNER JOIN T2 ON T1.A1=T2.A2
Ou
SELECT liste attributs
FROM T1 INNER JOIN T2 USING (A1)
```

INNER JOIN peut être remplacé par JOIN

```
SELECT liste attributs
FROM T1 NATURAL JOIN T2
```

Jointures externes: Left et Right Join

Permet de préciser que l'on garde tous les tuples d'une des deux tables (même ceux qui ne vérifient pas la condition)

LEFT JOIN : on garde tous les tuples de la table de gauche
RIGHT JOIN : on garde tous les tuples de la table de droite

```
SELECT liste attributs
FROM T1 LEFT JOIN T2 ON T1.A1=T2.A2
ou
FROM T1 RIGHT JOIN T2 ON T1.A1=T2.A2
```

Requêtes SQL imbriquées

- Sous-requêtes avec résultat unique

```
SELECT liste attributs
FROM liste tables
WHERE Attribut = Bloc SFW avec résultat valeur unique;
```

Opérateur de comparaison
= < > <= >=

- Sous-requêtes avec résultat ensemble

```
SELECT liste attributs
FROM liste tables
WHERE Attribut Co Bloc SFW avec résultat ensemble;
```

Connecteurs
IN - EXISTS / NOT EXISTS - ANY/ ALL

Requêtes de partitionnement

```
SELECT attribut de partitionnement, liste attributs
FROM liste tables cibles
WHERE condition(s) sur les tuples
GROUP BY attribut de partitionnement
HAVING condition sur les groupes;
```

fonctions statistiques

Quels sont les salaires maximaux des skippers pour chaque port de résidence?

```
SELECT skport, MAX(salaire) AS SalaireMax
FROM skipper
GROUP BY skport;
```

attribut de partitionnement

Requêtes avec clauses ensemblistes

Résultats « union compatibles »

```
Bloc SFW
UNION
Bloc SFW;
```

INTERSECT
MINUS

Donner la liste des ports dans lesquels sont domiciliés des skippers et/ou des bateaux?

```
(SELECT BATPORT AS Ports
FROM BATEAUX )
```

UNION

```
(SELECT SKPORT
FROM SKIPPERS ) ;
```

PORTS
AJACCIO
ANTIBES
CALVI

VUE

VUE = Table virtuelle définie à partir d'autres tables grâce à une requête

Définition d'une vue (ORACLE)

```
CREATE VIEW Nom-Vue AS Requete SFW;
```