

Langage SQL

Interrogation des données (requêtes)

FROM : les relations
Déclaration des variables

SELECT : Ce que l'on cherche
Les attributs qu'on veut à la sortie
DISTINCT : sans les doublons

AS : alias
Relation as R

SELECT ALL | **DISTINCT** attr1 , attr2...
FROM Relation1 **AS** R1 , Relation2 **AS** R2 ...
WHERE conditions

Group By
formation des sous-relations
Having
Restriction sur les sous-relations
Order by
tri du résultat

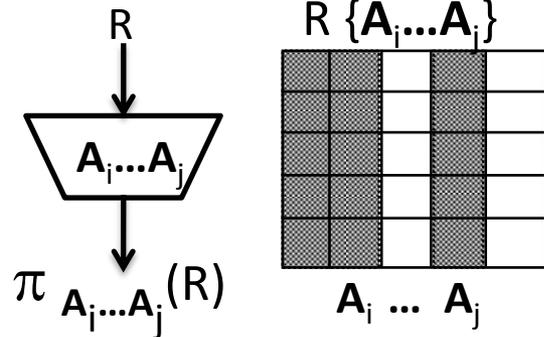
GROUP BY attr1 , ...
HAVING condition
ORDER BY attr1 , ...

WHERE : conditions à satisfaire
Conditions de restriction et de jointure
Opérateurs :
<, <=, =, <>, >=, >, **AND, OR, NOT, NULL, LIKE, BETWEEN, IN, EXISTS...**

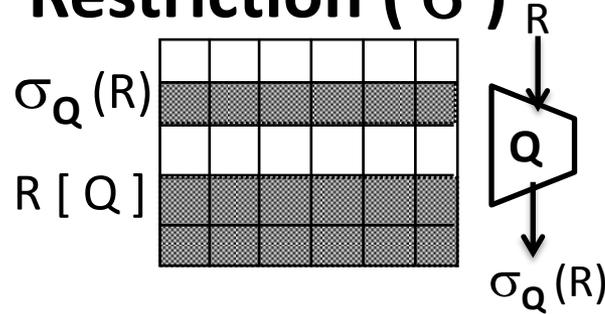
AVG(x), SUM(x), MAX(x), MIN(x), COUNT (x | *)
Calcul sur les valeurs de l'attribut x (Select, Having)

Rappel : algèbre relationnelle

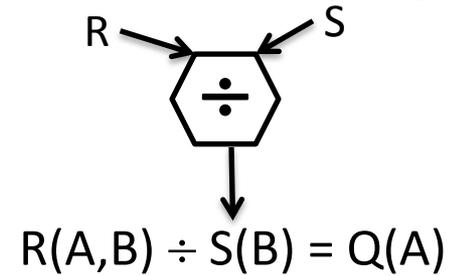
Projection (π)



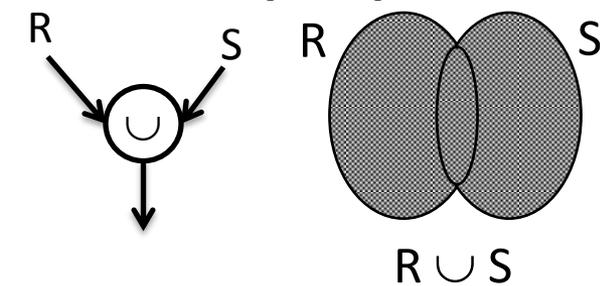
Restriction (σ)



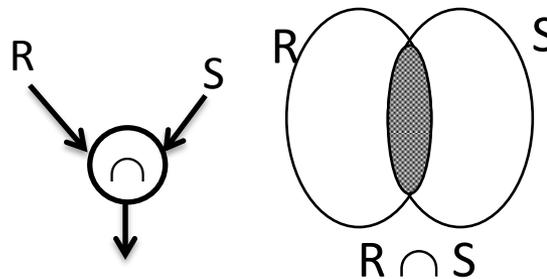
Division (\div)



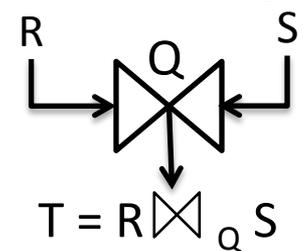
Union (\cup)



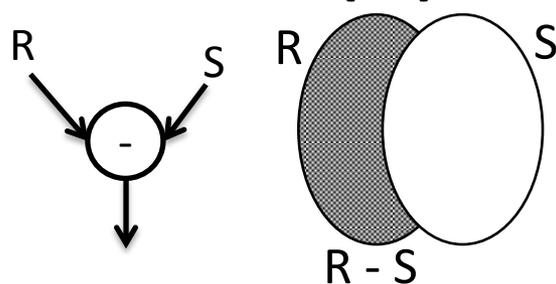
Intersection (\cap)



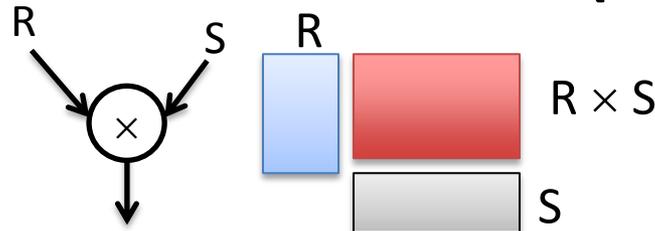
Jointure (\bowtie)



Différence ($-$)



Produit Cartésien (\times)



Équi-jointure : $A_i=B_i$

Teta-Jointure :

$<, \leq, >, \geq, \neq$

Jointure Naturelle

Même valeurs pour attributs homonymes

Collection de DVD

Film (*nfilm* , *titreOrig*, *genre* , *dir* (npers))

Personne (*npers* , *nom*)

DVD (*ndvd* , *type* , *dispo*, *nfilm*)

Acteur (*npers* , *nfilm*)

Langage SQL – Exemples

- *Quels films (titre original) de Steven Spielberg avons-nous en DVD (type = "dvd") ?*

```
SELECT F.titreOrig
FROM Film F, Personne P, DVD D
WHERE F.dir = P.npers AND F.nfilm = D.nfilm AND
      D.type = 'dvd' AND P.nom = "Steven Spielberg"
```

- *Quels sont les films pour lesquels on dispose d'un support dvd (type="dvd"), mais pas d'un support blue-ray (type="blueray") ?*

```
SELECT F.*
FROM Film AS F, DVD AS D1
WHERE F.nfilm = D1.nfilm
      AND F.nfilm NOT IN ( SELECT D2.nfilm FROM D2.type = "blueray" )
```

Langage SQL

Langage Manipulation Données (LMD)

- **INSERT INTO** : ajout de nouveaux n-uplets

```
INSERT INTO nom_table ( attr_1 , attr_2 , ... )  
VALUES (valeur_1 , valeur_2 , ... )
```

- **DELETE FROM** : suppression de n-uplets

```
DELETE FROM nom_table WHERE condition
```

- **UPDATE** : mises à jour des n-uplets

```
UPDATE nom_table SET attr_1 = val_1 , attr_2 = val_2 , ...  
WHERE condition
```

Langage SQL

Langage Définition des Données (LDD)

```
CREATE TABLE nom_table  
  ( attribut_1 domaine_attribut_1 contrainte_attrb_1,  
    attribut_2 domaine_attribut_2 contrainte_attrb_2 , ...  
    contrainte_relations );
```

Contrainte sur attribut

Domaine de valeur : *types des données*

INTEGER, VARCHAR, NUMERIC...

Domaine obligatoire : **NOT NULL**

Valeur par défaut :

DEFAULT *valeur*

Clé primaire : **PRIMARY KEY**

Unicité de l'attribut : **UNIQUE**

Contrainte sur relation

Clé primaire (composée) :

PRIMARY KEY (*attr1* , *attr2* , ...)

Contrainte référentielle :

FOREIGN KEY (*attr1* , *attr2* , ...)

REFERENCES table

(*attr1* , *attr2* , ...)

Langage SQL

Langage Définition des Données (LDD)

- **Type des données**

- *Numérique* :

- **INTEGER** : nb entier jusqu'à 32 bits ($-2^{31} \sim 2^{31}$)
 - **SMALLINT** : nb entier jusqu'à 16 bits ($-2^{15} \sim 2^{15}$)
 - **DECIMAL (n, p)** : nb réel de *n* chiffres, dont *p* après la virgule
 - **NUMERIC (n, p)** : idem DECIMAL (**max p** après la virgule)
 - **FLOAT** : nb réel

- *Chaîne de caractères*

- **CHAR (n)** : chaîne de caractères de **longueur n**
 - **VARCHAR (n)** : chaîne de caractères de taille variable (**max n**)
 - **TEXT (n)** : *idem VARCHAR (accepté par certains SGBD)*

- *Temporel*

- **DATE** : date (2009-11-09)
 - **TIME** : heure (00:11:22.333)

- *Logique*

- **Boolean** : valeur booléenne, vrai ou faux ('True' ou 'False')

Langage SQL

Langage Définition des Données (LDD)

- **ALTER TABLE** : modification d'une relation par l'ajout ou suppression d'une colonne

```
ALTER TABLE nom_table
```

```
    ADD attribut_1 domaine_1 contrainte_attrb
```

```
    DROP attribut_1
```

- **DROP TABLE** : suppression d'une relation

```
DROP TABLE nom_table ;
```

Langage SQL - Exemples

Langage Manipulation Données (LMD)

```
CREATE TABLE Projets
( NP      INTEGER    PRIMARY KEY ,
  ND      INTEGER    FOREIGN KEY REFERENCES Départements ,
  BudP    FLOAT      NOT NULL ,
  CHECK (BudP >= 0)
);
```

Contrainte

ND est la
clé étrangère

```
CREATE TABLE Hist_Sal
( MAT      INTEGER ,
  DatSal   DATE ,
  Sal      DECIMAL(2) NOT NULL ,
  PRIMARY KEY (MAT, DatSal) ,
  FOREIGN KEY (MAT) REFERENCES Employé (MAT)
);
```

MAT et DatSal forment
la clé primaire

MAT est la
clé étrangère

Langage SQL - Exemples

- **Langage Manipulation Données (LMD)**

INSERT INTO Départements (ND, Bud, NomD)
VALUES (100, 1500.00, "Financier")

UPDATE Départements **SET** Bud = Bud-150.00
WHERE ND=100;

DELETE FROM Départements **WHERE** ND=100

- **Langage Définition des Données (LDD)**

ALTER TABLE client **ADD** (*tel* CHAR (10) , *dateNasc* DATE);

DROP TABLE client ;