INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

- **Présentation globale des bases de données**
  - présentation, architecture, définitions, objectifs, historique, références.
- **Modélisations des données**
  - Systèmes hiérarchiques, réseaux, relationnels
- **Le modèle relationnel**
  - concepts, dépendances fonctionnelles, règles d 'intégrité, formes normales
- **Le langage SQL (Structured Query Language)**
  - définition de données, manipulation de données, contrôle des données, les vues.
- **Un aperçu de « NoSQL ».**
  - Définition des « index », manipulation de données.

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# I Présentation globale des bases de données

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# II Présentation globale des bases de données

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# III LE MODELE RELATIONNEL

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

- **Schéma logique : Modèle Relationnel**
- 1970,
  - CODD présente le modèle relationnel
  - Schéma logique représenté par des RELATIONS
- LE SCHÉMA RELATIONNEL
  - C'est l'ensemble des RELATIONS qui modélisent le monde réel
  - Les relations représentent les **entités** du monde réel (des personnes, des objets, etc.) ou les **associations** entre ces entités.
  - Passage d'un schéma conceptuel E-A à un schéma relationnel
    - Une **entité** est représentée par la relation :
      - nom_de_l'entité (liste des attributs de l'entité)
    - une **association** M:N est représentée par la relation :
      - nom_de_l'association (liste des identifiants des entités participantes, liste des attributs de l'association)

    Exemples :

    - CLIENT (IdCli, nom, ville) Entité
    - PRODUIT(IdPro, nom, prix, qstock) Entité
    - VENTE (IdCli, IdPro, date, qte) Association

- Passage d'un schéma E/A (E/R) à un schéma relationnel
  - En définissant les **entités**

Nom (**clé1**,….,**cléN**,attribut1, attribut2, attribut3,……., attributM)

Ou

Nom (clé1,….,cléN,attribut1, attribut2, attribut3,……., attributM)

Ou

NomAssociation **(liste des identifiants,**
liste des attributs de l'association**)**

  - En définissant les **associations** N,M

NomAssociation **(liste des identifiants des entités participantes,**
liste des attributs de l'association**)**

## LES AVANTAGES DU MODELE RELATIONNEL LOGIQUE

- Simplicité de présentation par tables
- Opérations relationnelles
  - algèbre relationnelle ( plus, moins, divise, mulitplie, etc…)
  - langage « assertionnel » ou spécialisé
- Indépendance physique
  - optimisation des accès
  - stratégie d'accès déterminée par le système
- Indépendance logique
  - concept de VUES
- Maintien de l'intégrité
  - contraintes d'intégrité définies au niveau du schéma

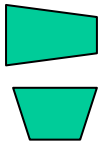
- LES AVANTAGES DU MODELE RELATIONNEL LOGIQUE
- Domaine
  - Ensemble de valeurs atomiques d 'un type sémantique :
    - NOM_SITE = {ATOMIUM, TOUR_EFFEL,STATUE_LIBERTE,...}
    - ensemble des valeurs possibles
- Relation
  - Sous ensemble du produit cartésien de plusieurs domaines
- N-uplets
  - un élément d 'une relation est un n-uplet de valeurs (tuple in english)
- Attributs
  - son nom doit être porteur de sens
  - différent du nom de domaine
  - plusieurs attributs peuvent avoir le même nom de domaine
- Schéma d 'une relation

- Exemple de définition de quatre entités relationnelles
  - Station (**id**, nom, capacité, lieu, région, tarif)
  - Activité (**idStation, libellé**, prix)
  - Client (**id**, nom, prénom, ville, région, solde)
  - Séjour (**id**, *idClient*, *idStation*, début, fin, nbPlaces)

- Entité exprimée sous forme de table
  - exemple de la table Station

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| pa | Passac | 400 | Alpes | Europe | 1000 |

- Les Opérateurs de base de l'algèbre relationnel
  - La restriction ou sélection: σ (unaire)
  - La projection: π (unaire)
  - Le produit cartésien: × (binaire)
  - L'union: ∪ (binaire)
  - L'intersection ∩ (binaire)
  - La division / (binaire)
  - La différence: − (binaire)



- Opérateurs dérivés
  - *Jointure* : ⋈, la composition d'un produit cartésien et d'une sélection.

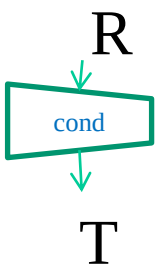
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

- Restriction/sélection
  - But :
    - sélectionner certains N-uplets
    - Réduire la taille verticalement
  - Contraintes
    - Unaire
    - Spécifier une condition
  - Exemple : stations aux Antilles
  - Station_antilles<- $\sigma_{\text{région='Antilles'}}$(Station)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| pa | Passac | 400 | Alpes | Europe | 1000 |

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |

  - Notation textuelle T <- $\sigma_{\text{cond}}(R)$
  - Notation graphique



INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

- Projection (unaire)
  - But :
    - sélectionner certains attributs
    - Réduire la taille horizontalement
    - Supprime les doublons
  - Contraintes
    - Unaire
    - Spécifier une liste d'attributs

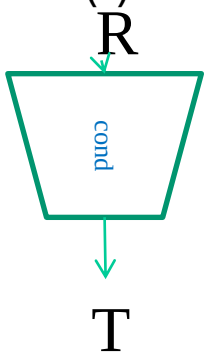
| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| pa | Passac | 400 | Alpes | Europe | 1000 |

## Exemple : région des noms

- Station_Region <- π nom,région (Station)

| nom | région |
|---|---|
| Venusa | Antilles |
| Farniente | Océan Indien |
| Santalba | Antilles |
| Passac | Europe |

- Notation textuelle T <- π condition (R)
- Notation graphique




- exemple π region(Station)

- Union : (Supprime les doublons)

StationAsie (**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| ma | MangaCity | 50 | Tokyo | Asie | 1800 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |

StationEurope (**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| pa | Passac | 400 | Alpes | Europe | 1000 |
| vt | Val Thorens | 350 | Alpes | Europe | 1200 |

– Station_AE= StationAsie U StationEurope

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| pa | Passac | 400 | Alpes | Europe | 1000 |
| vt | Val Thorens | 350 | Alpes | Europe | 1200 |
| ma | MangaCity | 50 | Tokyo | Asie | 1800 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |

- Intersection:

Station_Mer (**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |

Station_Antilles(**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| fo | FortAventure | 50 | Martinique | Antilles | 2200 |

StationMer_Antilles = Station_Antilles ∩ Station_Mer

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |

- Produit cartésien:

Station (**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| pa | Passac | 400 | Alpes | Europe | 1000 |

Activité (**libellé**, prix)

| libellé | prix |
|---|---|
| randonnée | 350 |
| Plongée | 150 |

Activité_Station (**id**, nom, capacité, lieu, région, tarif, **libelle**, prix)

= Activité (**libellé**, prix) X Station (**id**, nom, capacité, lieu, région, tarif)

# Produit cartésien: Résultat

Activité_Station (**id**, nom, capacité, lieu, région, tarif, **libelle**, prix) =
Activité (**libellé**, prix) X Station(**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif | libellé | prix |
|---|---|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 | Randonnée | 350 |
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 | plongée | 150 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 | Randonnée | 350 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 | plongée | 150 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 | Randonnée | 350 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 | plongée | 150 |
| pa | Passac | 400 | Alpes | Europe | 1000 | Randonnée | 350 |
| pa | Passac | 400 | Alpes | Europe | 1000 | plongée | 150 |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

- Différence :

Station_Mer (**id**, nom, capacité, lieu, région, tarif)

| id | Nom | Capacité | Lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |
| Sa | Santalba | 150 | Martinique | Antilles | 2000 |

Station_Antilles(**id**, nom, capacité, lieu, région, tarif)

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| va | Venusa | 350 | Guadeloupe | Antilles | 1200 |
| sa | Santalba | 150 | Martinique | Antilles | 2000 |
| fo | FortAventure | 50 | Martinique | Antilles | 2200 |

Station_MerSaufAntilles = Station_Mer - Station_Antilles

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| fa | Farniente | 200 | Seychelles | Océan Indien | 1500 |

Station_MerSaufAntilles = Station_Antilles - Station_Mer

| id | nom | capacité | lieu | région | tarif |
|---|---|---|---|---|---|
| fo | FortAventure | 50 | Martinique | Antilles | 2200 |

- Division : ( Élimine les doublons )

Station_Nom (nom, capacité, lieu)

| Nom | Capacité | Lieu |
|---|---|---|
| Venusa | 350 | Guadeloupe |
| Venusa | 350 | Seychelles |
| Santalba | 150 | Martinique |
| Madinia | 50 | Martinique |

Station_Capacité (capacité, lieu)

| capacité | lieu |
|---|---|
| 350 | Guadeloupe |
| 350 | Seychelles |

Station_Divise = Station_Nom / Station_Capacité

| nom |
|---|
| Venusa |

- Jointure :

Station_Nom (nom, capacité)

| Nom | Capacité |
|---|---|
| Venusa | 350 |
| Farniente | 200 |
| Santalba | 150 |
| Passac | 400 |

Station_Capacité (capacité, lieu)

| Région | lieu |
|---|---|
| Antilles | Guadeloupe |
| Ocean Indien | Seychelles |
| Antilles | Martinique |
| Europe | Alpes |

Station_Jointe = Station_Nom ⋈ Station_Capacité

ou σ capacité<400 et lieu != Martinique (Station_Nom X Station_Capacité)

| Nom | Capacité | Lieu | Région |
|---|---|---|---|
| Venusa | 350 | Guadeloupe | Antilles |
| Farniente | 200 | Seychelles | Océan Indien |

- Normalisation
  - Afin de réduire les doublons.
  - Définition des formes normales ajoutés au schéma relationnel :
    - **1ère forme : 1FN** : une relation est en 1FN si tout attribut est atomique à savoir non décomposable.

ELEVE( **n_eleve**, nom_prenom, liste_notes) devient en 1FN :

- ELEVE( **n_eleve**, nom,prenom)
- NOTE(**n_eleve**,matiere,note)

- Définition des formes normales ajoutés au schéma relationnel :
  - **2ème forme : 2FN : une relation est en 2FN si :**
    - **elle est** en 1FN
    - tout attribut n'appartenant pas à la clé ne dépend pas que d'une partie de la clé ou
    - aucun attribut ne faisant pas partie de la clé primaire ne doit dépendre que d'une partie de la clé primaire.

COMMANDE( **date**, **n_cli**, **n_produit**, qte, prix_unitaireHT )

Le prix_unitaire HT est dépendant de n°produit : partie de la clé !

Manière de le mettre en 2FN

- COMMANDE ( **date**, **n_cli**, **n_produit**,qte)
- PRODUIT(**n_produit**, prix_unitaire_HT)

! Ne pas respecter la 2FN entraîne des redondances. Cela gaspille de l'espace de stockage, et pose aussi le problème de la mise à jour des données.

- Définition des formes normales ajoutés au schéma relationnel :
  - **3ème forme : 3FN** : une relation est en 2FN si :
    - elle est en 2FN
    - si tout attribut n'appartenant pas à la clé primaire ne dépend pas d'un attribut non clé

    Ou

    - aucun attribut ne faisant pas partie de la clé primaire ne doit dépendre d'une partie des autres attributs ne faisant pas non plus partie de la clé primaire.

PERSONNE (**id_personne**, civilité, nom, prenom, sexe ) devient en **3FN** :

- PERSONNE (**id_personne ,** civilité, nom, prenom )
- CIVILITE ( **civilité**, sexe)

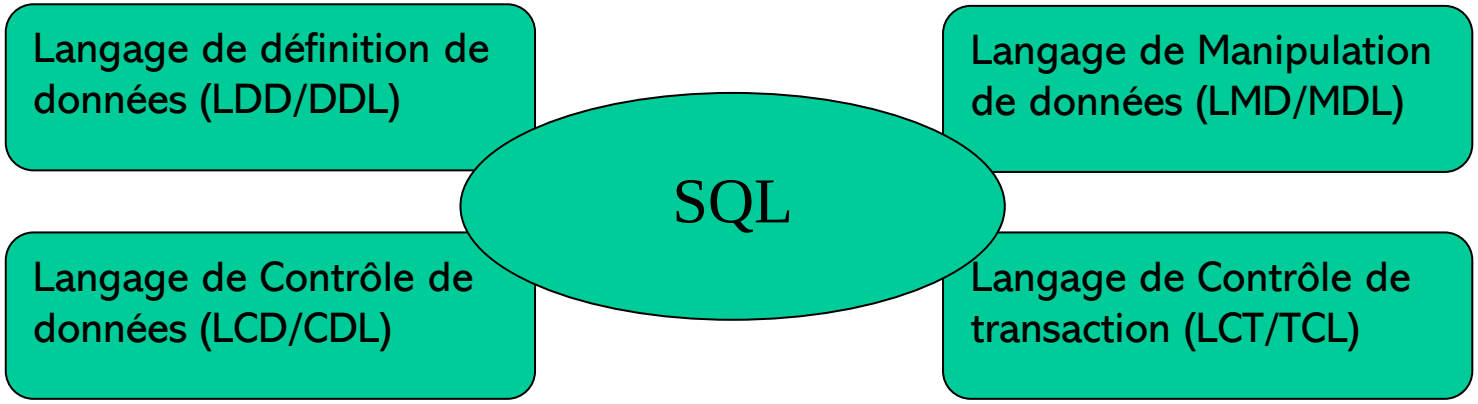
- Intégrité
  - de domaine
    - les valeurs d 'une colonne de relation doivent appartenir au domaine correspondant ( *liste de valeurs possibles*)
      - contrôle des valeurs des attributs
      - contrôle entre valeurs des attributs
  - de clé
    - les valeurs de clés primaires doivent être uniques
      - uniques => unicité de clé
      - non nulles => unicité des n-uplets
  - référentielle
    - les valeurs de clés étrangères sont nulles ou sont des valeurs de la clé primaires auxquelles elles font référence
      - relations indépendantes

      Séjour (**id**, *idClient*, *idStation*, début, fin, nbPlaces)

- Clé d'une relation
  - primaire :
    - attribut (ou groupe d 'attributs) qui détermine tous les autres
      - PRODUIT (**no_produit**, nom, prix_HT)
    - une clé détermine de manière unique une n-uplet
    - une relation peut posséder plusieurs clé **candidates**
      - *si nom de produit est unique, nom peut devenir la clé primaire.*
  - Clé étrangère ou clé secondaire
    - attribut (ou groupe d 'attributs) qui fait référence à la clé primaire d 'une autre relation
    - CATEGORIE(**no_categ**, designation, tva)
    - PRODUIT(**no_produit**, nom, marque, no_categ, prixHT)
      - no_categ est clé étrangère dans PRODUIT, c'est la clé primaire de CATEGORIE

# IV LE LANGAGE SQL

- Structured Query Langage (IBM)
  - Issu du langage QUEL, SEQUEL commercialisé par ORACLE
  - Inclut une grande partie des possibilités de l'algèbre relationnel
  - langage interprété : interface d 'utilisation (MySql, Squirrel, SQLylog, Dbeaver, MSAccess,…)
  - langage intégré dans un programme ( C, JAVA) : interface de programmation
  - procédures exécutés dans le SGBDR Intégrés mais exécutable de l 'extérieur par les interfaces d 'utilisation ou de programmation.



- Structured Query Langage (IBM):
  - Les commandes SQL
    - Définition de données (LDD)

      CREATE

      DROP

      ALTER
    - Manipulation de données (LMD)

      SELECT

      INSERT

      UPDATE

      DELETE
    - Contrôle de transactions (LCD)
      - Accès concurrents
        - COMMIT
        - ROLLBACK
      - Droits d'accès
        - GRANT / REVOKE

```
--LDD --
CREATE TABLE table
(
-- définition des colonnes
Nom  et type de colonne [ NOT NULL [UNIQUE] ] [ DEFAULT valeur ]
       [ PRIMARY KEY ] [ REFERENCES table ] [ CHECK condition ] ,... ,
-- contraintes de table
    [ PRIMARY KEY (liste de colonnes) ],
    [ UNIQUE (liste de colonnes) ] ,... ,
    [ FOREIGN KEY (liste de colonnes) REFERENCES table
    [ ON DELETE {RESTRICT | CASCADE | SET NULL} ]
    [ ON UPDATE {RESTRICT | CASCADE | SET NULL} ] ,... ,
    [ CHECK condition ] ,...
)
```

**-- Exemple de LDD--**

```
CREATE TABLE vente
    (
        IdCli     CHAR(4)   NOT NULL          REFERENCES client        ,
        IdPro     CHAR(6)   NOT NULL,
        date      DATE      NOT NULL,
        qte       SMALLINT                    CHECK (qte BETWEEN 1 AND 10),
      -- contrainte de table
      PRIMARY KEY (IdCli, IdPro, date)                ,
      FOREIGN KEY (IdPro) REFERENCES produit
      --ON DELETE CASCADE  ON UPDATE CASCADE  --SQL2 ONLY
    )
```

- ALTER TABLE permet de modifier la table

  ALTER TABLE client ADD COLUMN telephone CHAR(16)

- DROP TABLE détruit immédiatement l'accès la table

  DROP TABLE client

- CREATE INDEX permet d 'accélérer les recherches et de créer des indexes multi-colonnes

  CREATE [UNIQUE] INDEX index

  ON table (colonne [ASC|DESC], ...)

  l 'option UNIQUE assure l 'unicité de la clé

  l 'option ASC(endant) ou DESC(endant) permet de connaître l 'ordre de création de l 'indexe et donc de la recherche.

- Types de données
  - chaînes de caractère
    - **CHAR**(n) chaîne constante ou n est la longueur maximale
    - **VARCHAR**(n) chaîne variable ou n est la longueur maximale
  - entiers
    - **SMALLINT** sur 2 octets -32.768 à +32.767
    - **INTEGER** sur 4 octets -2.147.483.648 à +2.147.483.647
  - décimaux
    - **NUMERIC** => (n,m) le nombre de décimal doit être exactement m
    - **DECIMAL**(n,m)

- Types de données
  - Numériques à virgule flottante
    - **REAL** (SIMPLE PRECISION) au moins 7 chiffres significatifs
    - **FLOAT** (DOUBLE PRECISION) au moins 15 chiffres significatifs
  - Types temporels
    - DATE : jour : 2 chiffres, mois : 2 chiffres, année : 4 chiffres
    - TIME : heures, minutes, secondes : ex : 23:52:46,123
    - TIMESTAMP : heures, minutes, secondes : ex : 23:52:46,123456
    - INTERVAL : intervalle de temps
  - Valeur NULL
    - colonne non renseignée et donc vide d'information. La valeur n'est pas zéro, c'est une absence de valeur

- Contraintes d'intégrité
  - NOT NULL valeur null impossible
  - UNIQUE unicité d'un attribut
  - PRIMARY KEY clé primaire
  - FOREIGN KEY clé étrangère
  - CHECK plage ou liste de valeurs
- Toute opération violant une des contraintes sera rejetée
- Le système garantit l'intégrité des données
- Les contraintes d'intégrité peuvent être ajoutées ou supprimées :
  - sur une table
  - sur une colonne

Exemple

```
ALTER TABLE SALARIE
DROP CONSTRAINT NOM_UNIQUE
ADD CONSTRAINT SAL_MIN CHECK(SAL > 1000)
RENAME CONSTRAINT NOM1 TO NOM2
```

## --Manipulation de données (LMD) -- 4 commandes : SELECT INSERT UPDATE DELETE

- SELECT : selection de lignes ou colonnes

```
SELECT P.prix
FROM produit P
WHERE P.idPro = 'p1'
```

- INSERT ajout de lignes,

```
INSERT INTO client
(idPro, nom, ville)
VALUES ('c42','Duchemin','Bourges')
```

- UPDATE mise à jour de lignes,

```
UPDATE TABLE produit
SET P.prix= P.prix*1,21
WHERE  P.idPro='p2'
```

- DELETE suppression de lignes.

```
DELETE FROM produit
WHERE P.idPro='p2'
```

## --Manipulation de données (LMD) – SELECT

| | |
|---|---|
| SELECT | [DISTINCT] liste d'attributs, expressions, agrégat |
| FROM | liste de tables ou vues |
| WHERE | clause de recherche : qualifications / prédicats |
| GROUP BY | liste d'attributs de partion. |
| HAVING | qualification de groupe (agrégat) |
| ORDER BY | liste de colonnes [ ASC \| DESC ] |

- Contrairement à l'algèbre relationnel, SQL n'élimine pas les doublons : pour cela, il faut spécifier DISTINCT !
- les opérations arithmétiques sont disponibles ( +, -, *, / )
- les opérateurs d 'agrégats COUNT, MIN, MAX, SUM, AVG
- l 'étoile permet de lister tous les attributs
- Peut utiliser un tri du résultat ORDER
- Peut découper le résultat selon certains critères GROUP BY
- Peut restreinte le résultat selon certains critères HAVING

- La clause WHERE : condition de recherche
  - une condition de recherche est spécifiée par un prédicat.
  - Les qualifications ou prédicats simples :
    - ( =, <>, <, >, <=, >= )
    - **LIKE** contient ‘ %toto% ’ commence par ‘ toto% ’ , finit par ‘ %toto ’
    - **BETWEEN** : colonne entre deux valeurs : (BETWEEN 5000 AND 12000)
    - **IS NULL** : dont la valeur est inconnue
    - **IN** : colonne dans une liste :
      - P.marque IN (‘ LENOVO ’, ‘ APPLE ’, ‘ DELL ’)
    - **EXIST** : au moins une valeur définie
      - SELECT * FROM R
      - WHERE EXISTS (SELECT * FROM P WHERE R.attribut=P.attribut)
    - **ALL** : prédicat vrai pour tous
    - **ANY** : prédicat vrai pour au moins un

- La clause WHERE : condition de recherche
  - Les prédicats composés
    - composé de plusieurs prédicats simples articulés par :
      - **AND**
      - **OR**
      - **NOT**

- Les opérateurs d'agrégats
  - Pas d'équivalent en algèbre relationnelle,
  - Permettent d'effectuer des calculs sur des groupes de n-uplets
  - Principaux opérateurs :
    - **COUNT**: nombre de valeurs ou n-uplets,
    - **MIN** : minimum des valeurs,
    - **MAX** : maximum des valeurs,
    - **SUM** : somme des valeurs,
    - **AVG** : moyenne des valeurs.
- Exemple :

Nombre de client : SELECT COUNT(*) FROM client

Nombre de produit : SELECT COUNT(DISTINCT no_pro) FROM produit

- Le tri du résultat d ’un SELECT
  - **ORDER BY** : spécifie les colonnes qui vont définir le critère de tri dans l’ordre des colonnes
  - **ASC** (croissant) **, DESC** (décroissant): l ’ordre de tri ORDER BY peut être précisé par ASC ou DESC. Par défaut ASC est utilisé.

- **HAVING** Spécifie une condition de restriction de groupe nécessite la présence de l ’opérateur **GROUP BY**

```
SELECT      P.marque, AVG ( P.prix )
FROM        produit P
GROUP BY    P.marque
HAVING      AVG ( P.prix ) < 5000
```

- LES VUES
  - Une vue est une vision partielle ou particulière des données d'une ou plusieurs tables de la base.
  - La définition d'une vue est donnée par un SELECT qui indique les données de la base qui seront vues.
  - Les données des tables peuvent être modifiées à travers la vue
  - la commande de création de vue

    **CREATE VIEW** V_R (col1, col2...) AS SELECT … FROM R….
  - La spécification des noms des colonnes de la vue est facultative : par défaut, les colonnes de la vue ont pour nom les noms des colonnes résultat du SELECT.
  - Le SELECT d ’une vue ne peut pas utiliser la clause ORDER BY.
  - la commande de suppression de vue

    **DROP VIEW** V_R

- LCD : Langage de contrôle des données
  - Accès concurrents
    - COMMIT : valider la transaction
    - ROLLBACK : la transaction est annulée
  - Droits d'accès
    - GRANT : accorder des privilèges à d 'autres utilisateurs ou à tous
    - REVOKE : supprimer des privilèges à d autres utilisateurs
      - pour les droits de SELECT, INSERT, UPDATE, DELETE
  - Exemple : On accorde à Morgan le droit de sélectionner et d'insérer des lignes dans la relation Etudiant. Un utilisateur ayant reçu ce privilège peut le transmettre à son tour.

**GRANT SELECT ON Etudiant TO Morgan With Grant Option**

ou à tous : GRANT SELECT ON Etudiant TO **PUBLIC** With Grant Option

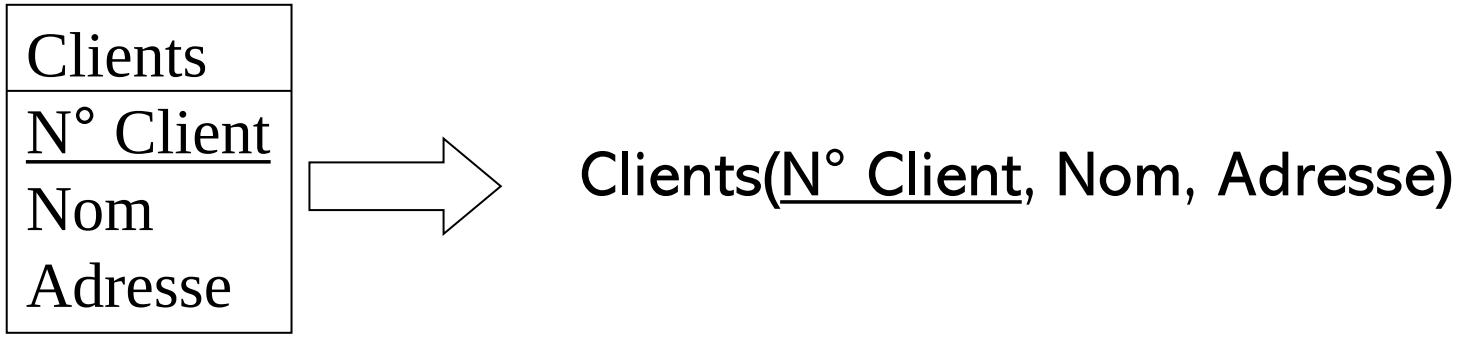
et la suppression :

**REVOKE privilège ON Etudiant FROM Tarik**

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

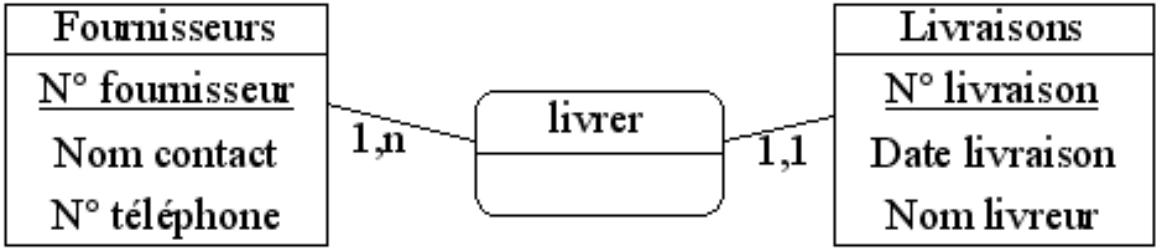
- Pour traduire un Modèle Conceptuel de Données en Modèle Relationnel de Données il suffit d'appliquer **5 règles**.
- Notations : on dit qu'une association binaire (entre deux entités ou réflexive) est de type :
  - 1 : 1 (un à un) si aucune des deux cardinalités maximales n'est de n,
  - 1 : n (un à plusieurs) si une des deux cardinalités maximales est de n,
  - n : m (plusieurs à plusieurs) si les deux cardinalités maximales sont de n.
- Un schéma relationnel ne peut pas faire la différence entre 0,n et 1,n. Par contre, il peut la faire entre 0,1 et 1,1.
- **Règle n°1 :** toute entité devient une table dans laquelle les attributs deviennent les colonnes. L'identifiant de l'entité constitue la clé primaire de la table.



## TRADUCTION du MCD vers un LMD Relationnel..

**Règle n°2 :** une association binaire de type 1 : n disparaît au profit d'une clé étrangère dans la table côté 0, 1 ou 1,1 qui fait référence à la clé primaire de l'autre table. Cette clé étrangère ne peut pas recevoir la valeur vide ou NULL, si la cardinalité est **1, 1**.

Exemple :




L'association livrer de la figure précédente se traduit par :

Fournisseurs(N° fournisseur, nom contact, n° téléphone),

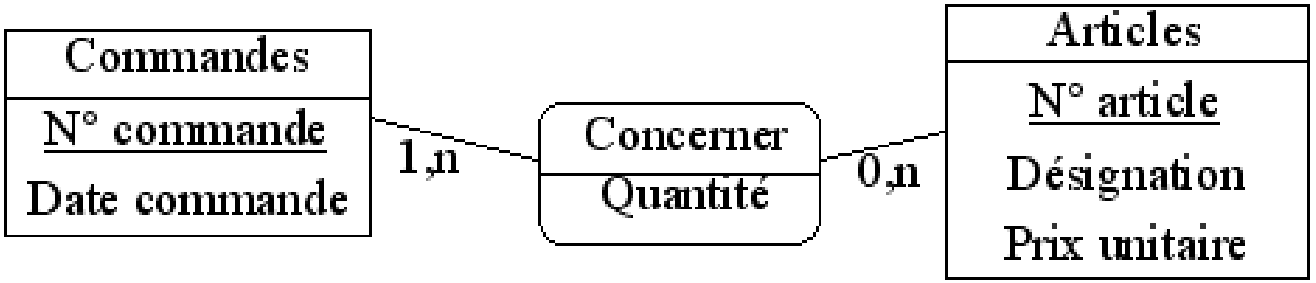
Livraisons(N° livraison, date livraison, nom livreur, #N° fournisseur (non vide)).

Il ne devrait pas y avoir d'attribut dans une association de type 1 : n, mais s'il en reste ils glissent vers la table côté 1:1.

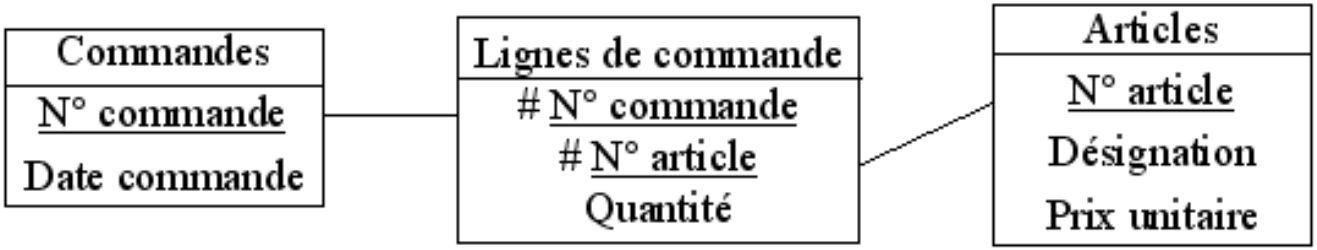
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

- **Règle n°3 :** une association binaire de type n : m devient une table supplémentaire (parfois appelée table de jonction, table de jointure ou table d'association) dont la clé primaire est composée de deux clés étrangères (qui référencent les deux clés primaires des deux tables en association). Les attributs de l'association deviennent des colonnes de cette nouvelle table
- Exemple



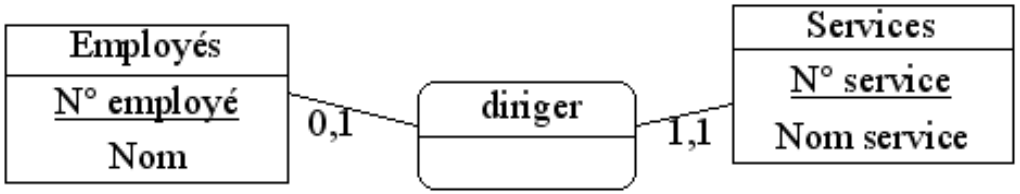

- Traduction



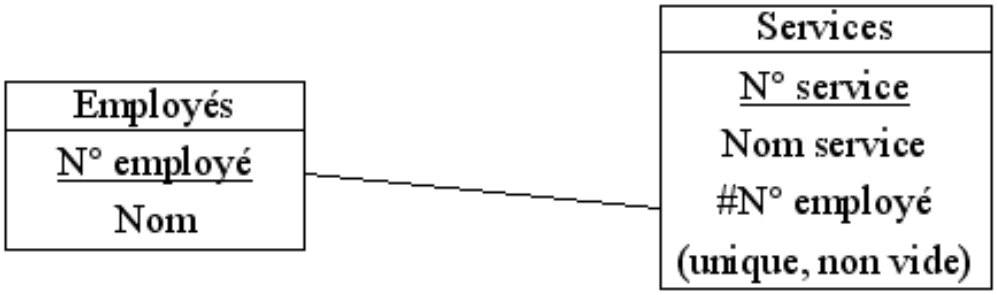
# TRADUCTION du MCD vers un LMD Relationnel.

**Règle n°4 :** une association binaire de type 1 : 1 est traduite comme une association binaire de type 1 : n sauf que la clé étrangère se voit imposer une contrainte d'unicité en plus d'une éventuelle contrainte de non vacuité (cette contrainte d'unicité impose à la colonne correspondante de ne prendre que des valeurs distinctes).

Il devrait y avoir au moins un côté de cardinalité 0,1. C'est alors dans la table du côté opposé que doit aller la clé étrangère. Exemple



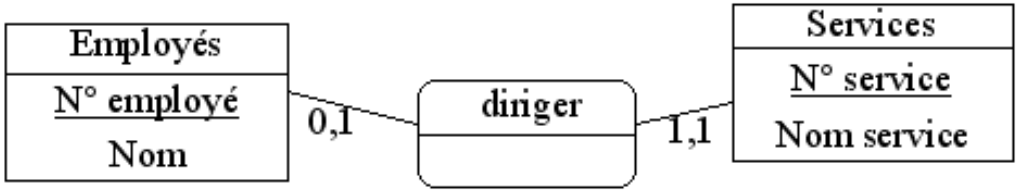

Traduction



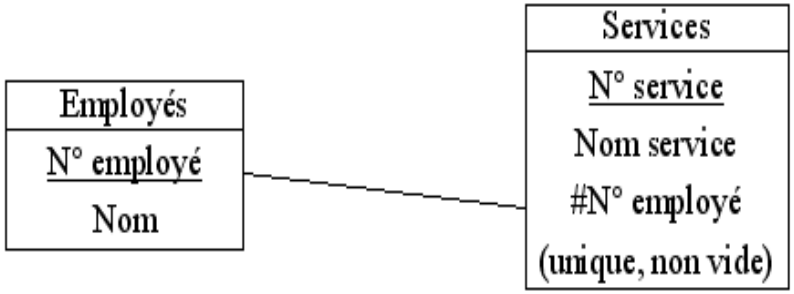
# TRADUCTION du MCD vers un LMD Relationnel.

**Règle n°4 – suite :** Si les deux côtés sont de cardinalité 0,1 alors la clé étrangère peut être placée indifféremment dans l'une des deux tables

Exemple




Traduction

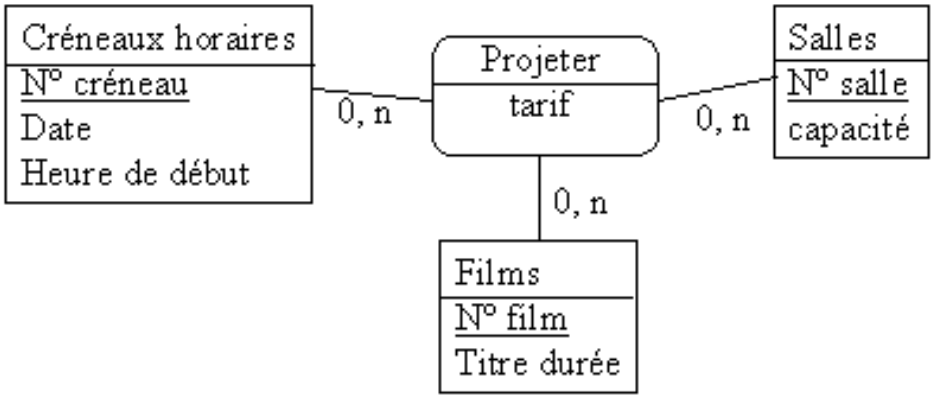


INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

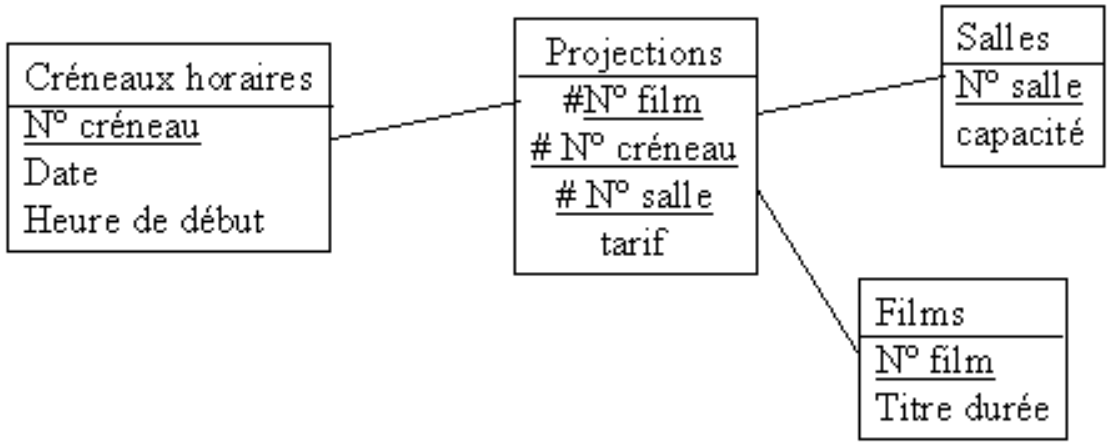
# TRADUCTION du MCD vers un LMD Relationnel.

**Règle n°5 :** une association non binaire est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entités en association. Les attributs de l'association deviennent des colonnes de cette nouvelle table.

Exemple :




Traduction

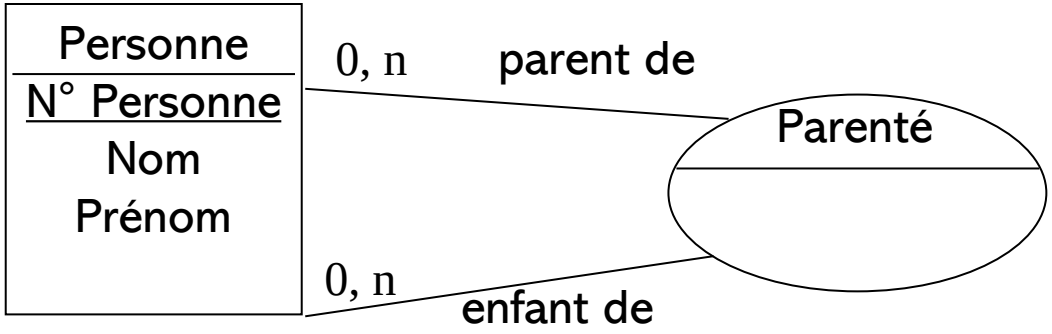


INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

**cas de la relation réflexive :** Dans certains cas, elle peut se traduire par une relation porteuse de deux attributs, duplications de l'identifiant de l'objet, et toutes deux renommées.

Exemple




Traduction

- Personne(N° Personne, Nom, Prénom),
- Parenté(N° Parent, N°Enfant).

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# TRADUCTION du MCD vers un LMD Relationnel.

- **Selection**
  - SELECT * FROM *R* WHERE *Condition ( dateEpreuve >= 01/10/04)*

| NoEt | matiere | dateEpreuve | note |
|---|---|---|---|
| 12 | Maths | 04/10/04 | 12 |
| 17 | Géographie | 07/10/04 | 9 |
| 12 | Géographie | 07/10/04 | 15 |
| 19 | Maths | 03/09/04 | 13 |
| 23 | Droit | 08/09/04 | 8 |
| 17 | Droit | 08/09/04 | 14 |

→

| NoEt | matiere | dateEpreuve | note |
|---|---|---|---|
| 12 | Maths | 04/10/04 | 12 |
| 17 | Géographie | 07/10/04 | 9 |
| 12 | Géographie | 07/10/04 | 15 |

- **Projection** :
  - SELECT DISTINCT(attribut) FROM LesRésultats WHERE Condition (NoEt>16)

| NoEt | matiere | dateEpreuve | note |
|---|---|---|---|
| 12 | Maths | 04/10/04 | 12 |
| 17 | Géographie | 07/10/04 | 9 |
| 12 | Géographie | 07/10/04 | 15 |
| 19 | Maths | 03/09/04 | 13 |
| 23 | Droit | 08/09/04 | 8 |
| 17 | Droit | 08/09/04 | 14 |

→

| NoEt |
|---|
| 17 |
| 23 |

# TRADUCTION du MCD vers un LMD Relationnel.

- **Renommage d 'une relation – exemple sur le produit cartésien**
  - un alias peut désigner une relation, il est valide pour la requête courante
  - il permet de simplifier la lecture d 'une expression
  - ex :

SELECT DISTINCT ( NoEt, nomEnseignant)

FROM LesInscription R1,LesEnseignants R2

WHERE R1.matiere = R2.matiere

| NoEt | matiere |
|---|---|
| 12 | Maths |
| 17 | Géographie |
| 12 | Géographie |
| 19 | Maths |
| 23 | Droit |
| 17 | Droit |
| 12 | Droit |
| 17 | Maths |

| matiere | nomEnseignant |
|---|---|
| Maths | Dupond |
| Géographie | Durand |
| Droit | Leblanc |

→

| NoEt | nomEnseignant |
|---|---|
| 12 | Dupond |
| 17 | Durand |
| 12 | Durand |
| 19 | Dupond |
| 23 | Leblanc |
| 17 | Leblanc |
| 12 | Leblanc |
| 17 | Dupond |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# TRADUCTION du MCD vers un LMD Relationnel.

- **Traduction du produit cartésien**
  - SELECT * FROM R,S
  - ex :

SELECT DISTINCT ( R1.NoEt, R2.NoEt)

FROM LesResultats R1, LesResultats R2

| NoEt | matiere | dateEpreuve | note |
|---|---|---|---|
| 12 | Maths | 04/10/04 | 12 |
| 17 | Géographie | 07/10/04 | 9 |
| 12 | Géographie | 07/10/04 | 15 |
| 19 | Maths | 03/09/04 | 13 |
| 23 | Droit | 08/09/04 | 8 |
| 17 | Droit | 08/09/04 | 14 |

→

| NoEt | NoEt |
|---|---|
| 12 | 12 |
| 12 | 17 |
| 12 | 19 |
| 12 | 23 |
| 17 | 12 |
| 17 | 17 |
| 17 | 19 |
| 17 | 23 |
| 19 | 12 |
| 19 | 17 |
| 19 | 19 |
| 19 | 23 |
| 23 | 12 |
| 23 | 17 |
| 23 | 19 |
| 23 | 23 |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# TRADUCTION du MCD vers un LMD Relationnel.

- **Traduction de la jointure**
  - SELECT * FROM R,S WHERE Condtion
  - ex :

SELECT DISTINCT (NoEt, nomEnseignant)

FROM LesInscriptions, LesEnseignants

WHERE LesInscriptions.matiere = LesEnseignants. matière

| NoEt | matiere |
|---|---|
| 12 | Maths |
| 17 | Géographie |
| 12 | Géographie |
| 19 | Maths |
| 23 | Droit |
| 17 | Droit |
| 12 | Droit |
| 17 | Maths |

| matiere | nomEnseignant |
|---|---|
| Maths | Dupond |
| Géographie | Durand |
| Droit | Leblanc |

| NoEt | nomEnseignant |
|---|---|
| 12 | Dupond |
| 17 | Durand |
| 12 | Durand |
| 19 | Dupond |
| 23 | Leblanc |
| 17 | Leblanc |
| 12 | Leblanc |
| 17 | Dupond |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

– Traduction de l'appartenance ∈

```
SELECT * FROM R
WHERE R.attribut IN ( Select attribut FROM S WHERE C )
```

– Traduction de l'exclusion ∉

```
SELECT * FROM R
WHERE R.attribut NOT IN ( Select attribut FROM S WHERE C )
```

# TRADUCTION du MCD vers un LMD Relationnel.

- Traduction de l 'Union

SELECT * FROM R WHERE Condition

UNION

SELECT * FROM S WHERE Condition

- Exemple :

SELECT NoEt FROM LesInscriptions WHERE matiere='Maths '

**UNION**

SELECT NoEt FROM LesInscriptions WHERE matiere = 'Géographie'

| NoEt | matiere |
|---|---|
| 12 | Maths |
| 17 | Géographie |
| 12 | Géographie |
| 19 | Maths |
| 23 | Droit |
| 17 | Droit |
| 12 | Droit |
| 17 | Maths |

| NoEt |
|---|
| 12 |
| 17 |
| 19 |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

- Traduction de l intersection

SELECT * FROM R WHERE Condition

INTERSECT

SELECT * FROM S WHERE Condition

- Exemple :

SELECT NoEt FROM LesInscriptions WHERE matiere='Maths ’

**INTERSECT**

SELECT NoEt FROM LesInscriptions WHERE matiere = 'Géographie'

| NoEt | matiere |
|---|---|
| 12 | Maths |
| 17 | Géographie |
| 12 | Géographie |
| 19 | Maths |
| 23 | Droit |
| 17 | Droit |
| 12 | Droit |
| 17 | Maths |

| NoEt |
|---|
| 12 |
| 17 |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# TRADUCTION du MCD vers un LMD Relationnel.

– Traduction de la différence : opérateur -

SELECT * FROM R

MINUS

SELECT * FROM S

Exemple :

SELECT NoEt FROM LesInscriptions WHERE matiere=Droit'

**MINUS**

SELECT NoEt FROM LesResultats WHERE matiere = 'Droit'

| NoEt | matiere |
|---|---|
| 12 | Maths |
| 17 | Géographie |
| 12 | Géographie |
| 19 | Maths |
| 23 | Droit |
| 17 | Droit |
| 12 | Droit |
| 17 | Maths |

| NoEt | matiere | dateEpreuve | note |
|---|---|---|---|
| 12 | Maths | 04/10/04 | 12 |
| 17 | Géographie | 07/10/04 | 9 |
| 12 | Géographie | 07/10/04 | 15 |
| 19 | Maths | 03/09/04 | 13 |
| 23 | Droit | 08/09/04 | 8 |
| 17 | Droit | 08/09/04 | 14 |

| NoEt |
|---|
| 12 |

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

## TRADUCTION du MCD vers un LMD Relationnel.

– Traduction de la division : Pas d 'équivalent en SQL !

R/S devient

```
SELECT A FROM R
MINUS
( SELECT * FROM R WHERE A,B IN (
            ( SELECT R.A, S.B FROM R,S
                        MINUS
             SELECT A,B FROM R )
)
```

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

# V UN APERCU DE « NOSQL »