



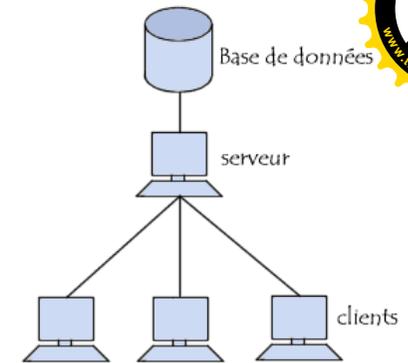
Introduction au langage SQL (Structured Query Language)

Premières requêtes



Les « Bases de Données »

► Plan du cours



1/ Qu'est ce qu'une base de données (relationnelle) ?

Concepts et terminologie

Une introduction par l'exemple

2/ Introduction à SQL : langage standard



Améliorer la lisibilité des requêtes

SQL (Structured Query Language)

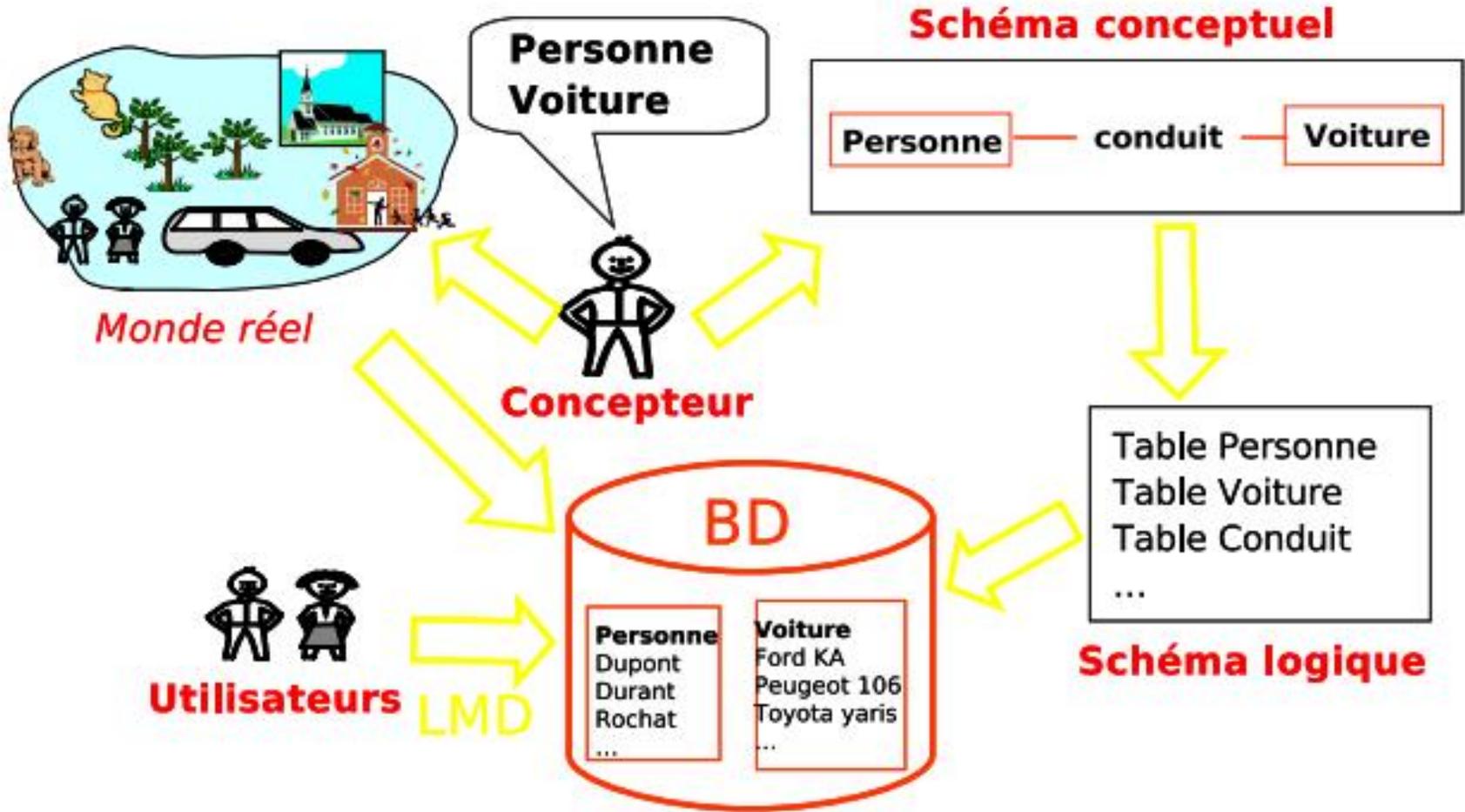
- ▶ Langage de Manipulation de Bases de Données développé par IBM (San José, 1981)
- ▶ Interrogation de bases de données ; Manipulation de bases de données dans des programmes.

Un standard industriel

NB. Il existe des dialectes "constructeurs"



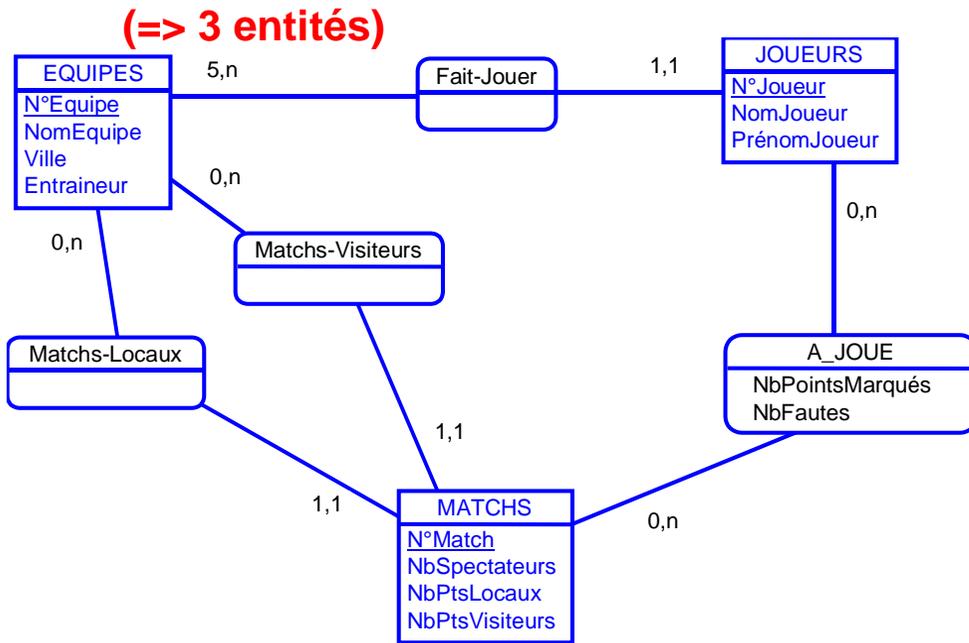
Rappels: cycle de vie de la base de donnée





Modèles représentatifs d'une BD

Modèle Conceptuel de Données



Modèle Logique de données

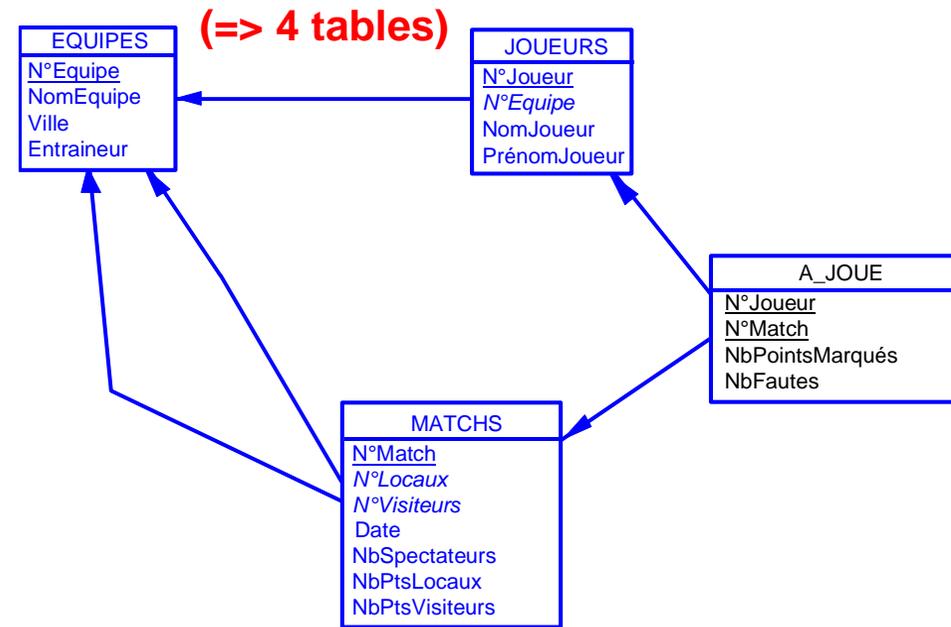


Schéma Relationnel (=> 4 Tables)

EQUIPES (N°Equipe, NomEquipe, Ville, Entraîneur)

JOUEURS (N°Joueur, N°Equipe, NomJoueur, PrénomJoueur)

MATCHES(N°Match, N°Locaux, N°Visiteurs, Date, NbSpectateurs, NbPtsLocaux, NbPtsVisiteurs)

A_JOUE (N°Joueur, N°Match, NbPoints Marqués, NbFautes)



1 – L'interrogation

- ▶ Forme générale :

SELECT liste-de-noms-de-colonnes
FROM liste-de-tables

Ex : Tous les joueurs

SELECT *
FROM JOUEURS ;

N°Joueur	N°Equipe	NomJoueur	PrénomJoueur
1	1	Prof	Jean
2	1	Simplet	Eric
3	1	Grincheux	Guy
4	1	Timide	Jean
5	1	Dormeur	Paul
6	2	Cantona	Eric
7	2	Picasso	Pablo
8	2	Dali	Salvador
9	2	Manet	Edouard
10	2	Matisse	Henri
11	3	Verlaine	Paul
12	3	Rimbaud	Arthur
13	3	Villon	François
14	3	Musset	Alfred de
15	3	Vigny	Alfred de
16	3	Cantona	Joël



1.1 – La projection

- ▶ La forme générale réalise une PROJECTION sans élimination des doubles (sélection de colonnes d'une table) :

Ex : Prénom de tous les joueurs

```
SELECT PrénomJoueur  
FROM JOUEURS ;
```

- ▶ Projection avec élimination des doubles

```
SELECT DISTINCT liste-de-noms-de-colonnes  
FROM liste-de-tables ;
```

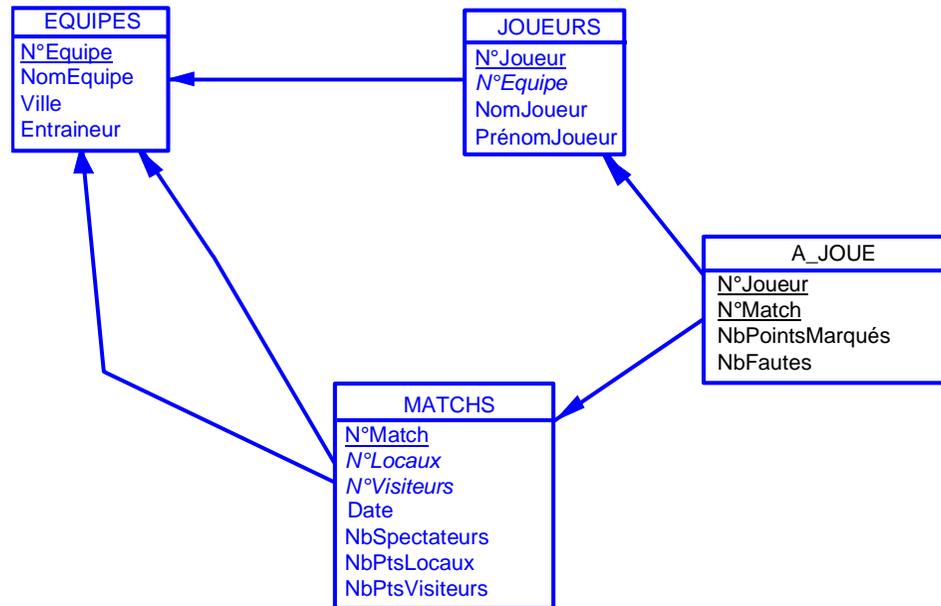
```
SELECT DISTINCT JOUEURS.PrénomJoueur  
FROM JOUEURS ;
```

PrénomJoueur
Jean
Eric
Guy
Jean
Paul
Eric
Pablo
Salvador
Edouard
Henri
Paul
Arthur
Francois
Alfred de
Alfred de
Joël

PrénomJoueur
Alfred de
Arthur
Edouard
Eric
Francois
Guy
Henri
Jean
Joël
Pablo
Paul
Salvador

1.1 – La projection

Q1 : Liste complète des noms d'équipe





1.2- La sélection

```
SELECT liste-de-noms-de-colonnes  
FROM liste-de-tables  
WHERE qualification ;
```

Une sélection est une restriction selon la qualification suivie d'une projection pour ne garder que les colonnes désignées.

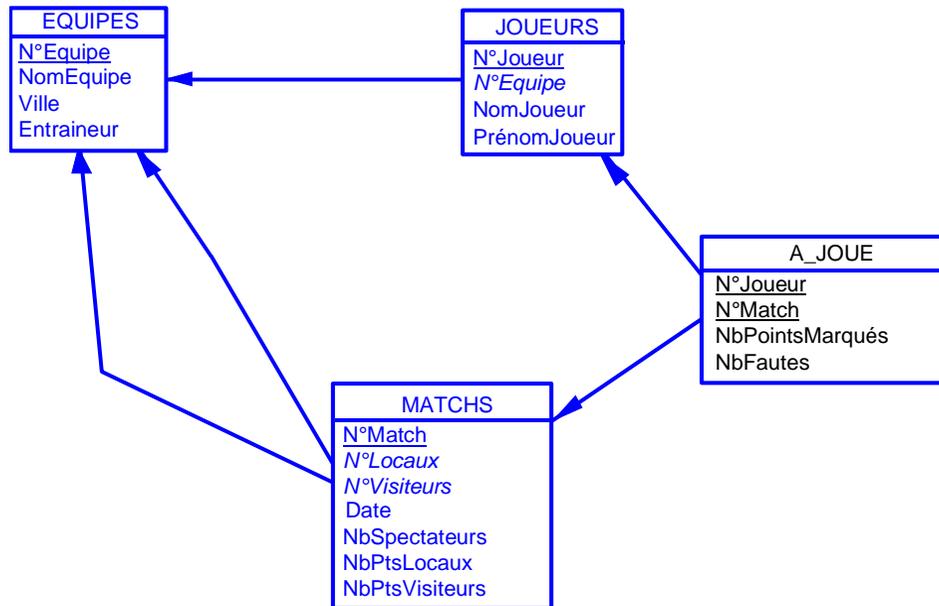
► *Liste des joueurs de l'équipe 1 :*

```
SELECT NomJoueur, N°Equipe  
FROM JOUEURS  
WHERE N°Equipe = 1 ;
```

NomJoueur	N°Equipe
Prof	1
Simplet	1
Grincheux	1
Timide	1
Dormeur	1

1.2- La sélection

Q2: Liste des joueurs dont le prénoms commence par la lettre 'J' ?





1.2.1 – Présentation du résultat

Tri des lignes de la table résultat.

ORDER BY nom-de-colonne [ordre] [, nom-de-colonne [ordre]] ...

ordre : **ASC** ou **DESC** ascendant ou descendant

```
SELECT NomJoueur, N°Equipe  
FROM JOUEURS  
WHERE N°Equipe = 1  
ORDER BY NomJoueur ASC;
```

NomJoueur	N°Equipe
Dormeur	1
Grincheux	1
Prof	1
Simplet	1
Timide	1



1.2.2- Qualification composée

La qualification peut être composée de plusieurs conditions combinées à l'aide des opérateurs booléens **AND** (et), **OR** (ou) et **NOT** (négation).

Les opérateurs de comparaisons sont **=**, **<>** ou **!=**, **<**, **<=**, **>** et enfin **>=**.

- *Liste des matchs ayant eu lieu chez l'équipe 1 avec plus de 6000 spectateurs :*

```
SELECT N°Match, N°Locaux, N°Visiteurs, Date, NbSpectateurs  
FROM MATCHS  
WHERE N°Locaux = 1 AND NbSpectateurs > 6000 ;
```

N°Match	N°Locaux	N°Visiteur	Date	NbSpectateurs
2	1	3	08/02/1996	10000



1.3- Le produit cartésien

Produit cartésien entre EQUIPES et MATCHS avec une projection pour ne pas avoir toutes les colonnes des deux tables :

```
SELECT EQUIPES.N°Equipe, EQUIPES.NomEquipe,  
MATCHS.N°Match , MATCHS.N°Locaux,  
MATCHS.Date, MATCHS.NbSpectateur,  
FROM EQUIPES, MATCHS ;
```

Q3: Si EQUIPES contient 3 tuples (lignes) et 4 attributs et MATCHS 6 tuples et 7 attributs combien de lignes et de colonnes en contient leur produit cartésien?

1. (18, 28)
2. (18,11)
3. (9, 11)



1.3- Le produit cartésien

Q4: Si on projette sur $N^{\circ}Equipe$, $NomEquipe$ et $N^{\circ}Match$, $N^{\circ}Locaux$, $Date$, $NbSpectateurs$, combien reste-il de colonnes et de lignes?

1. (28,6)

2. (18,9)

3. (28,9)

4. (18,6)



1.3- Le produit cartésien

Q4: Si on projette sur N°Equipe, NomEquipe et N°Match, N°Locaux, Date, NbSpectateurs, combien reste-il de colonnes et de lignes?

- 1. (28,6)
- 2. (18,9)
- 3. (28,9)
- 4. (18,6)

N°Equipe	NomEquipe	N°Match	N°Locaux	Date	NbSpectateurs
1	Les Nains	1	1	01/02/1996	5000
1	Les Nains	2	1	08/02/1996	10000
1	Les Nains	3	2	15/02/1996	15000
1	Les Nains	4	2	22/02/1996	9000
1	Les Nains	5	3	29/02/1996	21000
1	Les Nains	6	3	07/03/1996	12000
2	Les Peintres	1	1	01/02/1996	5000
2	Les Peintres	2	1	08/02/1996	10000
2	Les Peintres	3	2	15/02/1996	15000
2	Les Peintres	4	2	22/02/1996	9000
2	Les Peintres	5	3	29/02/1996	21000
2	Les Peintres	6	3	07/03/1996	12000
3	Les Poètes	1	1	01/02/1996	5000
3	Les Poètes	2	1	08/02/1996	10000
3	Les Poètes	3	2	15/02/1996	15000
3	Les Poètes	4	2	22/02/1996	9000
3	Les Poètes	5	3	29/02/1996	21000
3	Les Poètes	6	3	07/03/1996	12000



1.3- Le produit cartésien

Q4: Si on projette sur N°Equipe, NomEquipe et N°Match, N°Locaux, Date, NbSpectateurs, combien reste-il de colonnes et de lignes?

- 1. (28,6)
- 2. (18,9)
- 3. (28,9)
- 4. (18,6)

N°Equipe	NomEquipe	N°Match	N°Locaux	Date	NbSpectateurs
1	Les Nains	1	1	01/02/1996	5000
1	Les Nains	2	1	08/02/1996	10000
1	Les Nains	3	2	15/02/1996	15000
1	Les Nains	4	2	22/02/1996	9000
1	Les Nains	5	3	29/02/1996	21000
1	Les Nains	6	3	07/03/1996	12000
2	Les Peintres	1	1	01/02/1996	5000
2	Les Peintres	2	1	08/02/1996	10000
2	Les Peintres	3	2	15/02/1996	15000
2	Les Peintres	4	2	22/02/1996	9000
2	Les Peintres	5	3	29/02/1996	21000
2	Les Peintres	6	3	07/03/1996	12000
3	Les Poètes	1	1	01/02/1996	5000
3	Les Poètes	2	1	08/02/1996	10000
3	Les Poètes	3	2	15/02/1996	15000
3	Les Poètes	4	2	22/02/1996	9000
3	Les Poètes	5	3	29/02/1996	21000
3	Les Poètes	6	3	07/03/1996	12000



Concrètement

1.4- La jointure (1)

La jointure est le produit cartésien de deux relations avec une sélection (et c'est là que ça donne du sens).

- ▶ *Liste des matchs où les équipes reçoivent :*

SELECT EQUIPES.N°Equipe, EQUIPES.NomEquipe,
 MATCHS.N°Match , MATCHS.N°Locaux,
 MATCHS.Date, MATCHS.NbSpectateur

FROM EQUIPES, MATCHS
WHERE EQUIPES.N°Equipe = MATCHS.N°Locaux;

N°Equipe	NomEquipe	N°Match	N°Locaux	Date	NbSpectateurs
1	Les Nains	1	1	01/02/1996	5000
1	Les Nains	2	1	08/02/1996	10000
2	Les Peintres	3	2	15/02/1996	15000
2	Les Peintres	4	2	22/02/1996	9000
3	Les Poètes	5	3	29/02/1996	21000
3	Les Poètes	6	3	07/03/1996	12000



1.4– La jointure (2):

Lorsqu'on a besoin d'un complément d'information

Q5: Indiquez pour chaque nom de joueur le nom de son équipe correspondante....

Combien de tables?

Quelles projections?

Quelles sélections?



1.4.1 – Jointure de 3 tables

- ▶ *Liste des points marqués par les joueurs à domicile contre l'équipe 3.*

```
SELECT      JOUEURS.NomJoueur, MATCHS.N°Match,
            MATCHS.Date, A_JOUE.NbPointsMarqués,
            A_JOUE.NbFautes, MATCHS.N°Visiteur
FROM        MATCHS ,JOUEURS, A_JOUE
WHERE       JOUEURS.N°Joueur = A_JOUE.N°Joueur
            and MATCHS.N°Match = A_JOUE.N°Match
            and MATCHS.N°Visiteur=3 ;
```

NomJoueur	N°Match	Date	NbPointsMarqué	NbFautes	N°Visiteur
Prof	2	08/02/1996	10	5	3
Simplet	2	08/02/1996	12	4	3
Grincheux	2	08/02/1996	15	3	3

1.4.2– Imbrication de requêtes

- ▶ par utilisation d'opérateurs ensemblistes :

IN : appartenance à un ensemble

NOT IN : non appartenance

- ▶ *Liste des joueurs ayant participé à des matchs contre l'équipe 3.*

```
SELECT N°Joueur  
FROM MATCHS, A_JOUE  
WHERE  
MATCHS.N°Match = A_JOUE.N°Match  
and N°Visiteur=3
```

Ceux qui ont
joué contre
l'équipe 3

Ensemble des JOUEURS

```
SELECT NomJoueur  
FROM JOUEURS
```



1.4.2- Imbrication de requêtes

- ▶ par utilisation d'opérateurs ensemblistes :
 - IN** : appartenance à un ensemble
 - NOT IN** : non appartenance

▶ *Liste des joueurs ayant participé à des matchs à domicile contre l'équipe 3.*

Requête principale

```

SELECT      NomJoueur
FROM        JOUEURS
WHERE       N°Joueur IN

```

Requête imbriquée

```

(SELECT
FROM
WHERE
and N°Visiteur=3 );

```

Produit cartésien

NomJoueur
Prof
Simplet
Grincheux

jointure

sélection



1.4.2– Imbrication de requêtes

ANY : Un ou plusieurs éléments de l'ensemble satisfont la condition.

ALL : Tous les éléments de l'ensemble satisfont la condition.



1.4.2– Imbrication de requêtes

- Liste des matchs où $1/5^{\text{ème}}$ des points marqués par les locaux est inférieur à au moins un nombre de points marqués par un joueur :

```
SELECT    MATCHS.N°Match, MATCHS.N°Locaux,  
            MATCHS.NbPtsLocaux , A_JOUE.N°Joueur,  
            A_JOUE.NbPointsMarqués  
FROM      MATCHS, A_JOUE  
WHERE     MATCHS.N°Match = A_JOUE.N°Match  
            and MATCHS.NbPtsLocaux / 5 < ANY  
            (SELECT A_JOUE.NbPointsMarqués  
             FROM A_JOUE);
```

N°Match	N°Locaux	NbPtsLocaux	N°Joueur	NbPointsMarqué
1	1	89	1	20
1	1	89	2	10
1	1	89	3	14
2	1	56	1	10
2	1	56	2	12
2	1	56	3	15
3	2	78	1	15
3	2	78	2	18
3	2	78	3	16



1.4.2– Imbrication de requêtes

- Liste des matchs où $1/5^{\text{ème}}$ des points marqués par les locaux est inférieur à au moins un nombre de points marqués par un joueur :

```
SELECT    MATCHS.N°Match, MATCHS.N°Locaux,  
            MATCHS.NbPtsLocaux , A_JOUE.N°Joueur,  
            A_JOUE.NbPointsMarqués  
FROM      MATCHS, A_JOUE  
WHERE     MATCHS.N°Match = A_JOUE.N°Match  
            and MATCHS.NbPtsLocaux / 5 < ANY  
            (SELECT A_JOUE.NbPointsMarqués  
             FROM A_JOUE);
```

N°Match	N°Locaux	NbPtsLocaux	N° Joueur	NbPointsMarqué
1	1	89	1	20
1	1	89	2	10
1	1	89	3	14
2	1	56	1	10
2	1	56	2	12
2	1	56	3	15
3	2	78	1	15
3	2	78	2	18
3	2	78	3	16



1.4.2– Imbrication de requêtes

- Liste des matchs où $1/5^{\text{ème}}$ des points marqués par les locaux est inférieur à au moins un nombre de points marqués par un joueur :

```
SELECT    MATCHS.N°Match, MATCHS.N°Locaux,  
            MATCHS.NbPtsLocaux , A_JOUE.N°Joueur,  
            A_JOUE.NbPointsMarqués  
FROM      MATCHS, A_JOUE  
WHERE     MATCHS.N°Match = A_JOUE.N°Match  
            and MATCHS.NbPtsLocaux / 5 < ANY  
            (SELECT A_JOUE.NbPointsMarqués  
             FROM A_JOUE);
```

N°Match	N°Locaux	NbPtsLocaux	N° Joueur	NbPointsMarqué
1	1	89	1	20
1	1	89	2	10
1	1	89	3	14
2	1	56	1	10
2	1	56	2	12
2	1	56	3	15
3	2	78	1	15
3	2	78	2	18
3	2	78	3	16



1.4.2– Imbrication de requêtes

- ▶ *Liste des matchs où 1/5^{ème} des points marqués par les locaux est inférieur à tous les points marqués par un joueur :*

```
SELECT      MATCHS.N°Match, MATCHS.N°Locaux,  
              MATCHS.NbPtsLocaux , A_JOUE.N°Joueur,  
              A_JOUE.NbPointsMarqués  
FROM        MATCHS, A_JOUE  
WHERE       MATCHS.N°Match = A_JOUE.N°Match  
              and MATCHS.NbPtsLocaux / 5 < ALL  
              (SELECT A_JOUE.NbPointsMarqués  
                FROM A_JOUE);
```

Table résultat ??



1.5- Autres possibilités d'interrogation

UNION : sélection de lignes appartenant à l'une ou à l'autre de 2 relations (avec suppression des doublons).

► *Liste des noms des entraîneurs et des joueurs :*

```
SELECT JOUEURS.NomJoueur
FROM JOUEURS
UNION
SELECT EQUIPES.Entraîneur
FROM EQUIPE;
```

► D'autres opérateurs SQL existent ...

INTERSECT : sélection de lignes appartenant aux 2 relations.

MINUS : sélection de lignes d'une relation n'appartenant pas à l'autre relation.

NomJoueur
BlancheNeige
Cantona
Dali
Dormeur
Grincheux
Hugo
Kandinski
Manet
Matisse
Musset
Picasso
Prof
Rimbaud
Simplet
Timide
Verlaine
Vigny
Villon



1.6- Opérateurs d'agrégats

AVG : Moyenne d'un ensemble de valeurs

SUM : Somme d'un ensemble de valeurs

COUNT : Nombre de valeurs d'un ensemble

MAX : Valeur Maximum d'un ensemble de valeurs

MIN : Valeur Minimum d'un ensemble de valeurs

► *Moyenne de spectateurs par match :*

SELECT **AVG**(MATCHS.NbSpectateurs)
FROM MATCHS;

Expr1000
12000

ou

SELECT **SUM**(MATCHS.NbSpectateurs) / **COUNT**(*)
FROM MATCHS;



1.6- Opérateurs d'agrégats

- ▶ *Maximum de spectateurs dans un match :*

```
SELECT MAX(MATCHS.NbSpectateurs)  
FROM MATCHS;
```

Expr1000
21000

- ▶ *Nombre total de joueurs :*

```
SELECT COUNT(*)  
FROM JOUEURS;
```

Expr1000
16



1.7- Partition de relations

GROUP BY : application des opérateurs d'agrégat sur des sous-relations obtenues.

- ▶ *Liste des moyennes de points marqués par les joueurs et leur moyenne de fautes :*

```
SELECT      A_JOUE.N°Joueur,  
            AVG(A_JOUE.NbPointsMarqués),  
            AVG(A_JOUE.NbFautes)  
FROM        A_JOUE  
GROUP BY   A_JOUE.N°Joueur;
```

N°Joueur	Expr1001	Expr1002
1	18.75	3.25
2	15	2.5
3	15.5	2.75



1.7- Partition de relations

HAVING : possibilité d'appliquer des conditions sur les sous-relations.

- ▶ *Liste des moyennes de points marqués par les joueurs et leur moyenne de fautes pour ceux dont la moyenne est > à 2.5 :*

```
SELECT      A_JOUE.N°Joueur,  
              AVG(A_JOUE.NbPointsMarqués),  
              AVG(A_JOUE.NbFautes)  
FROM        A_JOUE  
GROUP BY    A_JOUE.N°Joueur  
HAVING      AVG(A_JOUE.NbFautes) > 2.5 ;
```

N° Joueur	Expr1001	Expr1002
1	18.75	3.25
3	15.5	2.75



1.7- Partition de relations

WHERE et **GROUP BY** :

- ▶ *Liste des moyennes de points marqués par les joueurs et leur moyenne de fautes, pour les matchs 2 et 3 :*

```
SELECT      A_JOUE.N°Joueur, AVG(A_JOUE.NbPointsMarqués),  
              AVG(A_JOUE.NbFautes)  
FROM        A_JOUE  
WHERE        A_JOUE.N°Match=2 OR A_JOUE.N°Match=3  
GROUP BY    A_JOUE.N°Joueur;
```

N° Joueur	Expr1001	Expr1002
1	12,5	4
2	15	3,5
3	15,5	3,5



1.7- Partition de relations

WHERE, **GROUP BY** et **HAVING** :

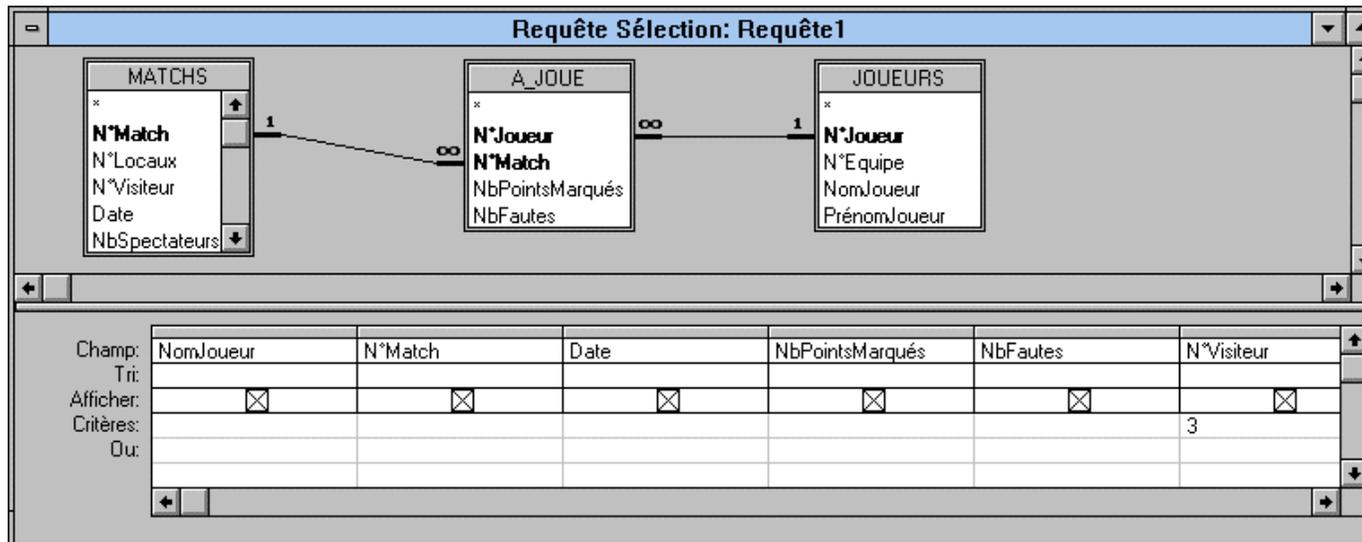
Sélection selon la condition du **WHERE**, puis réalisation du **GROUP BY** et enfin sélection selon la condition du **HAVING**.

- *Pour les matchs 2 et 3, donner la liste des moyennes de points marqués par les joueurs et leur moyenne de fautes pour ceux dont la moyenne est > à 2.5 :*

```
SELECT      A_JOUE.N°Joueur, AVG(A_JOUE.NbPointsMarqués),  
              AVG(A_JOUE.NbFautes)  
FROM        A_JOUE  
WHERE       A_JOUE.N°Match=2 OR A_JOUE.N°Match=3  
GROUP BY   A_JOUE.N°Joueur  
HAVING     AVG(A_JOUE.NbFautes) > 2.5 ;
```

N° Joueur	Expr1001	Expr1002
1	12,5	4
2	15	3,5
3	15,5	3,5

Suppression des doublons



Remarques :

DISTINCT/DISTINCTROW

DISTINCTROW

élimination des doublons avant projection.

DISTINCT

élimination des doublons après projection.



Suppression des doublons

- *Requête avec DISTINCTROW :*

```
SELECT DISTINCTROW JOUEURS.[N°Equipe], JOUEURS.PrénomJoueur  
FROM JOUEURS, A_JOUE  
WHERE JOUEURS.[N°Joueur] = A_JOUE.[N°Joueur];
```

N°Equipe	PrénomJoueur
1	Jean
1	Eric
1	Jean

- *Requête avec DISTINCT :*

```
SELECT DISTINCT JOUEURS.[N°Equipe], JOUEURS.PrénomJoueur  
FROM JOUEURS, A_JOUE  
WHERE JOUEURS.[N°Joueur] = A_JOUE.[N°Joueur];
```

N°Equipe	PrénomJoueur
1	Eric
1	Jean



2- Mise à jour d'une base

- *Insertion d'une ligne :*

INSERT INTO nom-relation [(Champ1, Champ2, ...)

VALUES (Valeur1, Valeur2, ...) ;

INSERT INTO JOUEURS (N°Joueur, N°Equipe, NomJoueur,
PrénomJoueur)

VALUES (17, 1, "Joyeux", "Michel");

- *Modification des valeurs de champs d'une ligne :*

UPDATE nom-relation

SET Champ1 = Valeur1, Champ2 = Valeur2, ...

WHERE condition ;

UPDATE JOUEURS

SET PrénomJoueur = "Alain"

WHERE [N°Joueur] = 17;



2- Mise à jour d'une base

- *Suppression d'une ligne :*

DELETE FROM nom-relation

WHERE condition ;

DELETE FROM JOUEURS

WHERE [N°Joueur] = 17;



3 – Gestion des tables

- *Création de table.*

CREATE TABLE nom-table

(nom-d'attribut description, ...) ;

Créer une table de nom Nouvelle Table avec deux champs de type Texte, un champ de type Entier et un champ Date/Heure. Le champ NSS est contraint à être la clé primaire.

CREATE TABLE NouvelleTable

(Nom **TEXT**, Prénom **TEXT**,

NuméroSS **INTEGER**

CONSTRAINT IndexPrim **PRIMARY KEY** ,

DateNaissance **DATETIME**);



3 – Gestion des tables

- *Suppression de table.*

DROP nom-table

- *Création d'index.*

CREATE [UNIQUE] INDEX nom-index

ON nom-relation (nom-d'attribut, ...);

Créer un index de nom IndNomJoueur :

CREATE INDEX IndNomJoueur

ON JOUEURS (NomJoueur);

- *Suppression d'index.*

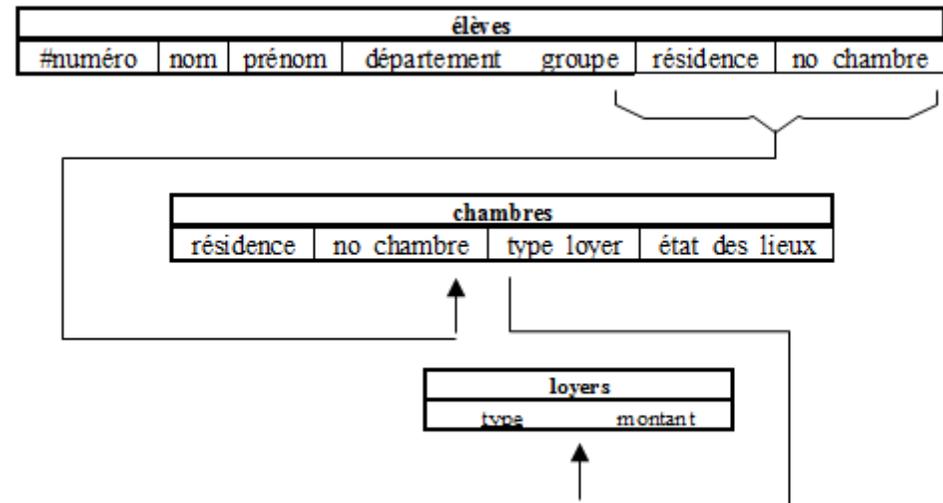
DROP INDEX nom-index



Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des élèves (Numéro, Nom et prénom) logeant dans la résidence A.



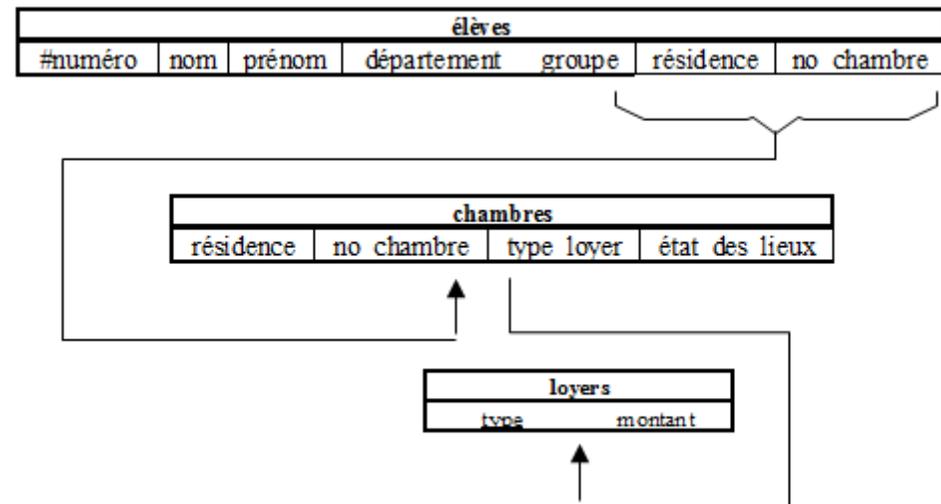


Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des élèves (Numéro, Nom et prénom) logeant dans la résidence A.

```
Select numéro, nom, prénom  
From Elèves  
Where résidence = 'A'
```

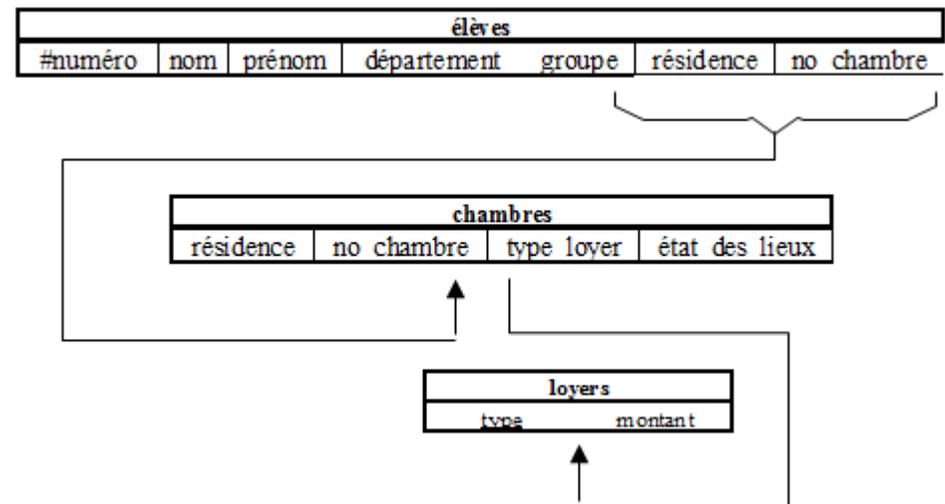




Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des élèves (les deux noms) en binôme associés à leur chambre (résidence et numéro).





Retour sur la gestion des résidences

► Quelques requêtes

Liste des élèves (les deux noms) en binôme associés à leur chambre (résidence et numéro).

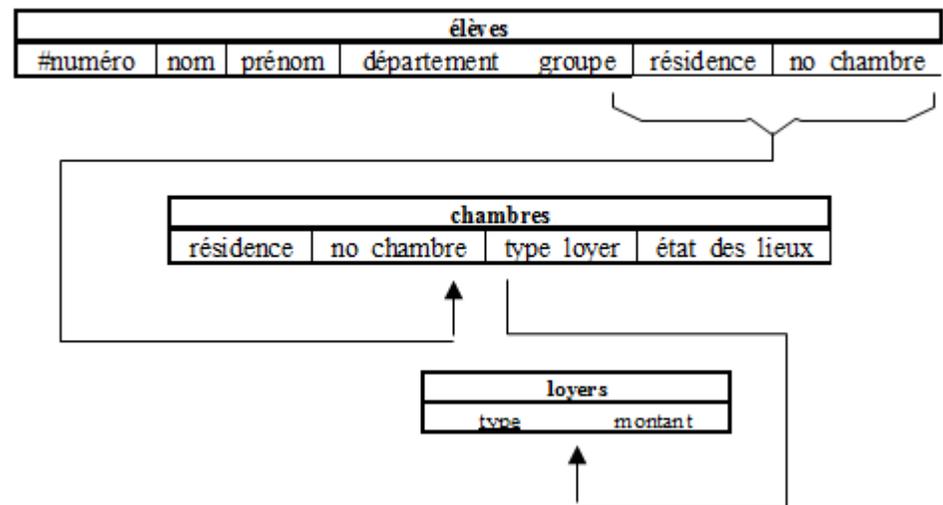
```
Select  E1.nom, E2.nom, E1.résidence, E1.No_chambre
From    Elèves as E1, Elèves as E2
Where   E1.numéro <> E2.numéro
And     E1.résidence = E2.résidence
And     E1.No_chambre = E2.No_chambre
And     E1.résidence is not null
And     E1.No_chambre not null
```



Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des chambres d'état des lieux « passable ».



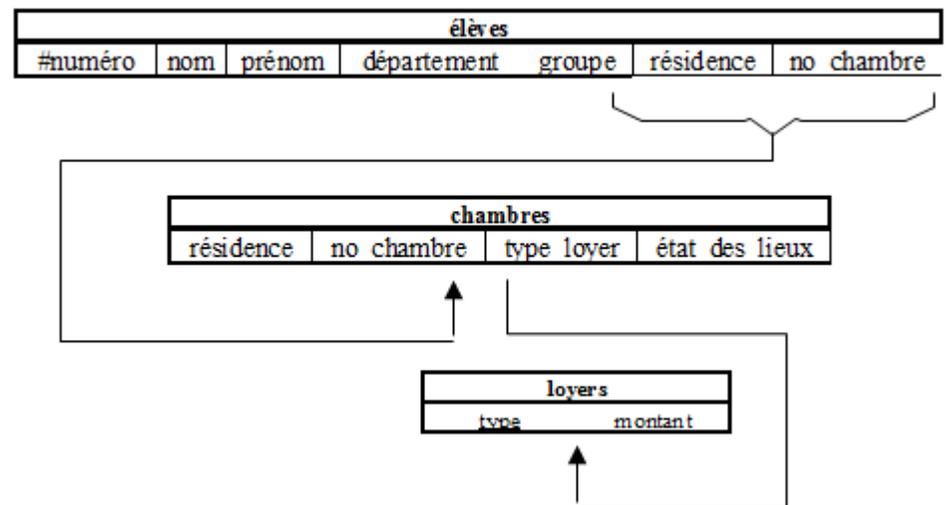


Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des chambres d'état des lieux « passable ».

```
Select résidence, No_chambre  
From chambres  
Where état_des_lieux = 'passable'
```

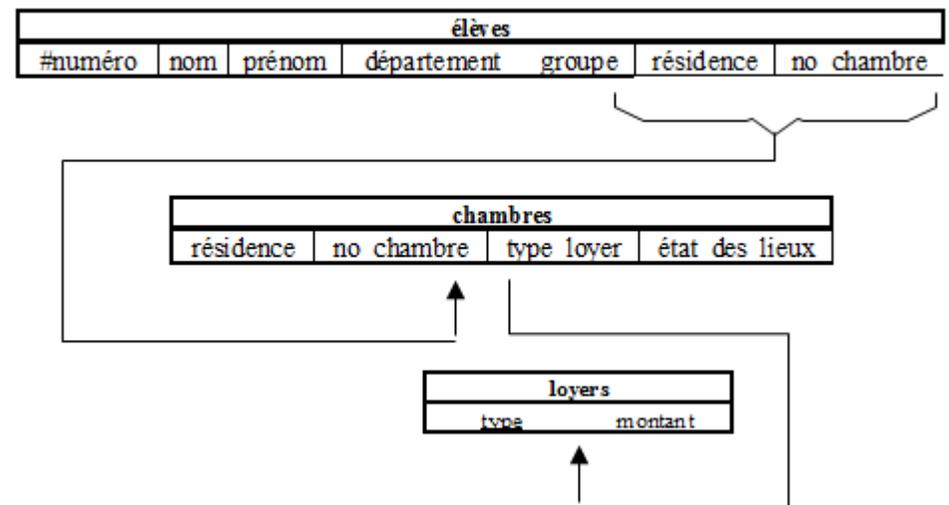




Retour sur la gestion des résidences

- ▶ Quelques requêtes

Liste des étudiants logés dans des chambres à 250 € de loyer



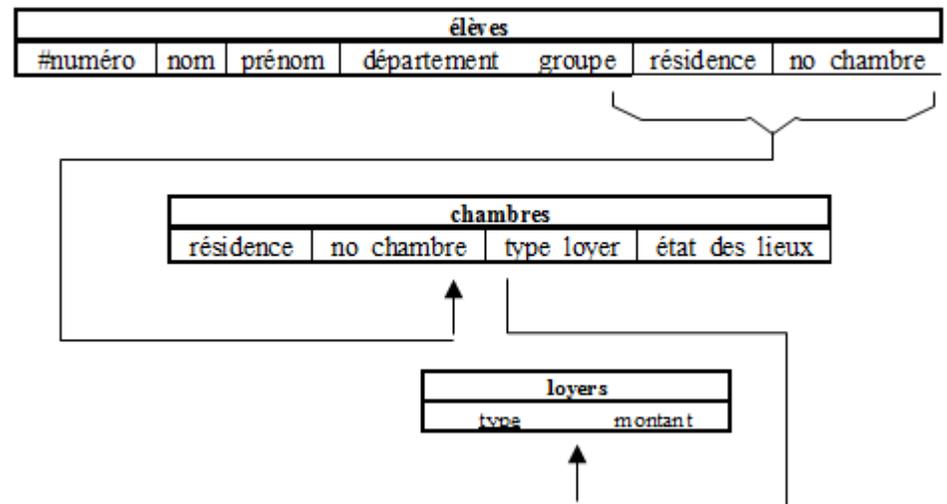


Retour sur la gestion des résidences

► Quelques requêtes

Liste des étudiants logés dans des chambres à 250 € de loyer

```
Select nom, prénom, montant  
From élèves, chambres, loyers  
Where élèves.résidence =chambres.résidence  
And chambre.type_loyer=loyer.type_loyer  
And montant=250
```

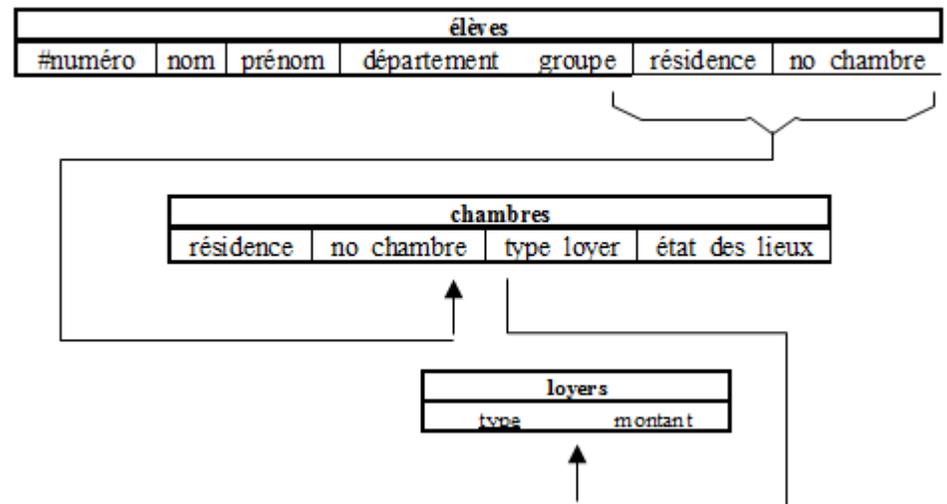




Retour sur la gestion des résidences

- ▶ Quelques requêtes

Dans quelle chambre dort Pierre Dumont?



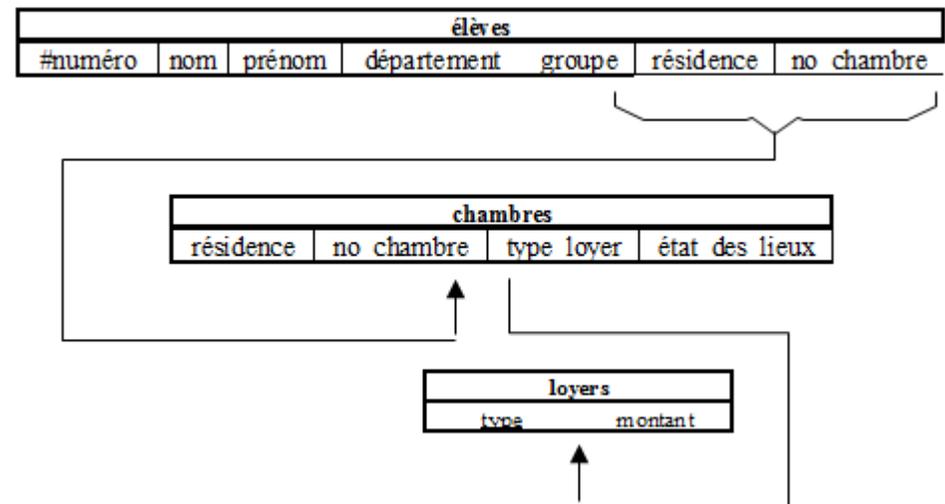


Retour sur la gestion des résidences

- ▶ Quelques requêtes

Dans quelle chambre dort Pierre Dumont?

```
Select no_chambre, résidence  
from Elèves  
Where nom='Dumont' and prénom='Pierre'
```

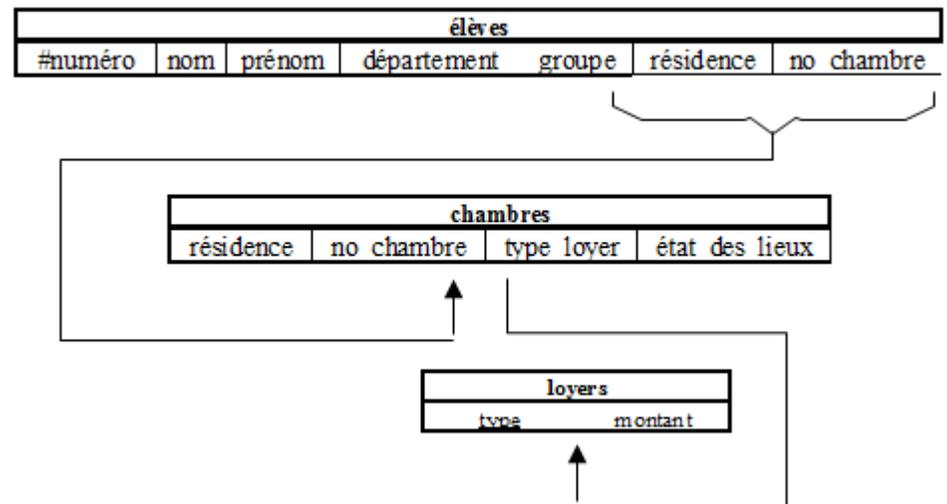




Retour sur la gestion des résidences

- ▶ Quelques requêtes

Quels sont les loyers des chambres inoccupées?





Retour sur la gestion des résidences

► Quelques requêtes

Quels sont les loyers des chambres inoccupées?

Ensemble de toutes les chambres (occupées ou non) =

$$A = \Pi_{C.Résidence, C.No_Chambre} (C)$$

Ensemble des chambres occupées =

$$B = \Pi_{E.Résidence, E.No_chambre} (\sigma_{(E.Résidence \text{ is NOT NULL and } E.No_Chambre \text{ is NOT NULL})}(E))$$

On veut faire le lien avec le loyer!! Il faut donc rajouter le champ «type_loyer» dans A et dans B



Retour sur la gestion des résidences

► Quelques requêtes

Quels sont les loyers des chambres inoccupées?

Ensemble de toutes les chambres (occupées ou non) =

$$A' = \Pi_{C.Résidence, C.No_Chambre, C.Type_loyer} (C)$$

Ensemble des chambres occupées =

$$B' = \Pi_{E.Résidence, E.No_chambre, C.Type_loyer} (\sigma_{(E.Résidence \text{ is NOT NULL and } E.No_Chambre \text{ is NOT NULL AND } E.Résidence=C.Résidence \text{ and } E.No_Chambre=C.No_chambre)}(E \otimes C))$$



Retour sur la gestion des résidences

► Quelques requêtes

Quels sont les loyers des chambres inoccupées?

Ensemble de toutes les chambres inoccupées =

$$C' = A' - B'$$

Enfinement les loyers correspondant...

$$\prod_{L.Montant} (\sigma_{L.Type=C'.Type_loyer})(C' \otimes L)$$