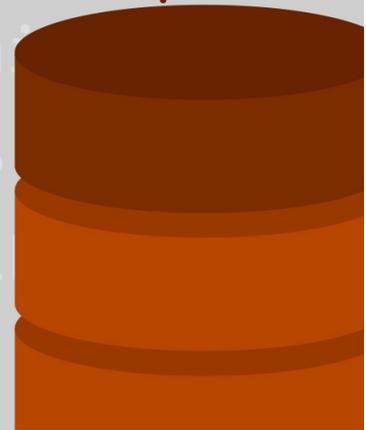
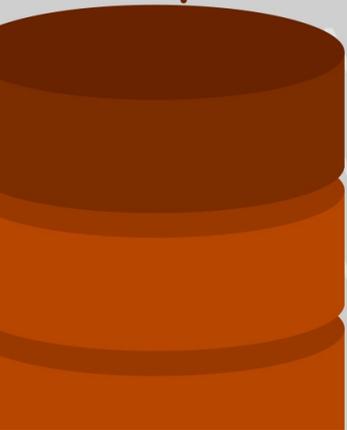


SELECT \* FROM

.SQL



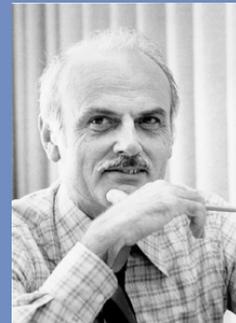
*i*

Les premiers modèles de base de données ne permettent pas de décrire de façon satisfaisante les relations entre deux données à l'aide de pointeurs logiques.

Dans les années 1960 et 1970, Edgar Frank « Ted » Codd (23 août 1923 - 18 avril 2003) un informaticien britannique se met à la recherche d'une solution satisfaisante d'arrangement de données.

Cette recherche aboutit au **modèle relationnel**.

Il recherche à décrire les relations entre les données de façon logique et mathématique, sous le nom d'algèbre relationnelle, dont le théorème de Codd est l'un des théorèmes fondamentaux de la théorie des bases de données.



Source : Wikipedia

## Modèle relationnel

### Définition

### Modèle relationnelle

Le modèle relationnel, c'est :

- Une structure unique, la relation (ou table)
- Des contraintes qui définissent des formes normales, évitant les défauts de conception
- Un langage (SQL) pour effectuer les requêtes

### Définition

### Relation

Dans le modèle relationnel, la structure pour représenter les données est la **relation**. Elle contient des données ayant un lien entre-elles.

## Exemple

On peut par exemple créer une relation contenant le nom des départements et leur code.

Ardèche	↔	07
Gard	↔	30
Isère	↔	38
Manche	↔	50
Paris	↔	75
Mayotte	↔	976

## Définition

### Base de donnée

Une **base de données** est un objet qui contient un ensemble de relation sur lesquelles on peut effectuer des recherches, des ajouts, des mise à jour par le biais d'un langage.

## Représentation et vocabulaire



Comment représente-t-on une relation ?

Une relation est un objet abstrait, on peut la représenter de différentes manières. Une représentation possible est la table, qui s'avère dans beaucoup cas la représentation la plus pratique.

Départements	
nom	code
Ardèche	07
Gard	30
Isère	38
Manche	50
Paris	75
Mayotte	976

On notera que la représentation inclus un nommage de chaque colonne (le **nom** du département, son **code**, dans notre exemple). Ce nommage n'est pas strictement indispensable (on pourrait utiliser la position par exemple), mais s'avère très pratique et sera donc utilisé systématiquement.

## Définition

### Vocabulaire

Dans cette représentation, on dira que :

- le tableau est une **relation** ou une **table**
- une colonne est un **attribut**.
- une ligne (sauf la première ligne) est une **occurrence** ou une **entrée** ou un **n-uplet** ou encore un **enregistrement**
- la première ligne est une **entête** et contient le nom des attributs.

Départements	
nom	code
Ardèche	07
Gard	30
Isère	38
Manche	50
Paris	75
Mayotte	976

Plusieurs termes coexistent pour désigner les mêmes éléments suivants le modèle choisi (relationnel ou table). On choisira pour les cours le vocabulaire du modèle relationnel.

Terme du modèle relationnel
Relation
Occurrence ou N-uplet
Attribut
Nom d'attribut
Valeur d'attribut
Domaine

Terme de la représentation par table
Table
Ligne
Colonne
Nom de colonne
Cellule
Type

## Domaine

### Définition

### Domaine

Chaque attribut est défini par son nom et son type. Le type des données est appelé **domaine**. Les domaines les plus courants sont :

- INTEGER ou INT : Nombre entier
- FLOAT : Nombre réel en notation flottante
- CHAR(X) : Chaîne de caractères de taille fixe de **X** caractères
- VARCHAR(X) : Chaîne de caractères de taille variable d'au plus **X** caractères
- DATE (AAAA-MM-JJ) : Date contenant l'année, le mois et le jour
- DATETIME (AAAA-MM-JJ HH :MM :SS) : Date et heure

## Schéma relationnel simple

### Définition

### Schema relationnel

On peut aussi décrire une relation par les éléments qui le constitue.

On note cette représentation **schéma relationnel (simple)**.

On y décrira :

1. Le nom de la relation.
2. Un nom (distinct) pour chaque attribut,
3. Le domaine de valeur (type) de chaque attribut.

## Exemple

En utilisant notre exemple, la relation **Départements** peut ainsi se décrire par **Départements( nom, code )** ou en précisant le domaine de valeur **Departements( nom: VARCHAR(255), code: INT)**

---

## Métriques

---

### Définitions

L' **ordre** est le nombre d'attributs de la relation.

Le **cardinal** est le nombre d'occurrences de la relation.

---

### Exemple

- La relation **Départements** est d'ordre 2.  
En effet y a 2 colonnes : **Nom** et **Code**
  - Le cardinal de la relation **Départements** est 6.  
Il y a 6 occurrences : (Ardèche, 07 ),(Gard, 30 ),(Isère, 38 ),(Manche, 50 ),(Paris, 75 ),(Mayotte, 976 ).
-

---

### Exercice 1 ★

Dans cet exercice, nous disposons d'une relation `Coupe_du_monde` qui contient les années, les vainqueurs et les pays organisateurs des différentes coupe du monde de football féminin.

Id	Année	Champion	Hote
1	1991	Etats-Unis	Chine
2	1995	Norvège	Suède
3	1999	Etats-Unis	Etats-Unis
4	2003	Allemagne	Etats-Unis
5	2007	Allemagne	Chine
6	2011	Japon	Allemagne
7	2011	Etats-Unis	Canada

1. Écrire le schéma relationnel de la table `Coupe_du_monde`.
  2. Donner l'ordre et le cardinal de cette relation.
- 

### Exercice 2 ★

Un grand magasin de meubles propose à ses clients plusieurs meubles. Les informations correspondantes sont rangées dans une base de données comportant la relation `Meubles` dont on donne le contenu ci-dessous :

id	intitule	prix	stock	description
62	'skap'	69.99	2	'Armoire blanche 3 portes'
63	'skap'	69.99	3	'Armoire noire 3 portes'
74	'stol'	39.99	10	'Chaise en bois avec tissu bleu'
98	'hylla'	99.99	0	'Bibliothèque 5 étages blanche'

1. Donner le schéma relationnel de la relation `Meubles`.
  2. Quel est le cardinal de la relation `Meubles` ?
  3. Quel est l'ordre' de la relation `Meubles` ?
  4. Combien de meubles sont disponibles dans le magasin ?
-



## Langage SQL

*i* Le SQL (**Structured Query Language**) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web. SQL permet notamment de lire, insérer, modifier et supprimer des données dans une base.

### • Les requêtes de sélection

**SELECT ... FROM ... ;**

Avec le langage SQL, il est possible de faire des requêtes de sélection sur un ou des **noms d'attribut**. Ce type de requêtes est construit à partir des instruction **SELECT ... FROM ...**. On doit préciser le.s attribut.s que l'on souhaite sélectionner ainsi que le nom de la relation (ou table)

#### •• Sélection d'un attribut dans une table

Pour une requête portant sur un seul attribut, on utilisera une requête du type

```
1 SELECT attribut FROM nom_relation;
```

#### Exemple

Pour sélectionner les attributs **code** de la table **departements**, on utiliserait la requête :

```
1 SELECT code FROM departements;
```

#### •• Sélection de plusieurs attributs dans une table

Pour sélectionner plusieurs attributs, il faut les séparer par une virgule dans la requête.

```
1 SELECT attribut1, attribut2 FROM nom_relation;
```

### Exemple

Pour sélectionner les valeur de l'attribut **code** ainsi que celles de l'attribut **nom** de la table **departements**, on utiliserait la requête :

```
1 SELECT code, nom FROM departements ;
```

- Sélection de tous les attributs dans une table

```
1 SELECT * FROM table
```

### Exemple

Pour sélectionner les valeur de tous les attributs de la table **departements**, on utiliserait la requête :

```
1 SELECT * FROM departements
```

- Sélection avec filtre

```
SELECT ... FROM ... WHERE ... ;
```

- Sélection avec un filtre

```
1 SELECT * FROM nom_relation WHERE attribut1 = valeur1;
```

### Exemple

Pour sélectionner les valeurs de l'attribut **nom** de la table **departements** qui ont un **code** supérieur à 40.

```
1 SELECT nom FROM departements WHERE code > 40
```

- Sélection avec plusieurs filtres

Pour pouvoir utiliser plusieurs filtre, on utilise les connecteurs **AND** ou **OR**.

```
1 SELECT * FROM nom_relation WHERE attribut1 = valeur_1 AND attribut2 = valeur_2;
```

### Exemple

Pour sélectionner les valeurs de l'attribut **nom** de la table **departements** qui ont un **code** supérieur à 40 et inférieur à 50.

```
1 SELECT nom FROM departements WHERE code > 40 AND code < 50;
```

---

### • Sélection avec un classement

---

**SELECT ... FROM ... ORDER BY ... ;**

- Sélection avec un classement par ordre croissant

```
1 SELECT * FROM nom_relation ORDER BY attribut1 ;
```

- Sélection avec un classement par ordre décroissant

```
1 SELECT * FROM nom_relation ORDER BY attribut1 DESC;
```

### • Compter

---

**SELECT COUNT(...) FROM ... ;**

- Compter toutes les entrées

```
1 SELECT COUNT(*) FROM nom_relation ;
```

- Compter les entrées distinctes

```
1 SELECT COUNT(DISTINCT attribut) FROM nom_relation
```

### • Insertion d'une occurrence dans une table

---

**INSERT INTO ... VALUES ... ;**

```
1 INSERT INTO nom_relation(attribut1, attribut2, attribut3) VALUES (valeur1, valeur2,
↳ valeur3);
```

#### Exemple

Pour insérer le département Mayotte avec le code 976.

```
1 INSERT INTO departement(Nom, Code) VALUES ("Mayotte",976);
```

---

### • Mise à jour de la valeur d'une occurrence dans une table

---

**UPDATE ... SET ... ;**

```
1 UPDATE nom_relation SET attribut1 = nv_valeur;
```



Cette requête va modifier **toutes** les occurrences de la table.

Souvent on va coupler cette requête avec un filtre. Ainsi on si l'on veut changer la valeur de l'**attribut1** de l'occurrence qui a pour **attribut2** la valeur2.

```
1 UPDATE nom_relation SET attribut1 = nv_valeur WHERE attribut2 = valeur2;
```

### • Suppression d'une occurrence dans une relation

**DELETE FROM ... WHERE ... ;**

```
1 DELETE FROM nom_relation WHERE attribut = valeur;
```

### Exemple

La requête pour supprimer l'occurrence (**Ardèche, 07**) dans la relation **departements** est donc

```
1 DELETE FROM departement WHERE code = 07;
```

### • Opérateurs de comparaisons

Il existe plusieurs opérateurs de comparaisons. La liste ci-jointe présente quelques uns des opérateurs les plus couramment utilisés.

Opérateur	Description
=	Égale
!=	Pas égale
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égale à
<=	Inférieur ou égale à
IN	Liste de plusieurs valeurs possibles
BETWEEN	Valeur comprise dans un intervalle donnée (utile pour les nombres ou dates)
LIKE	Recherche en spécifiant le début, milieu ou fin d'un mot. (On utilise la wildcard % )
IS NULL	Valeur est nulle
IS NOT NULL	Valeur n'est pas nulle

## ● Exercices

### Exercice 3 ★

*i* L'énoncé de cet exercice peut utiliser les mots du langage SQL suivants :  
SELECT ... FROM, SELECT ... FROM ... WHERE, INSERT INTO ... VALUES

La relation **Films** est décrite par le tableau ci-dessous.

Films				
id	titre	realisateur	annee_sortie	note_sur_10
1	Alien, le huitième passager	Scott	1979	10
2	Dune	Lynch	1985	5
3	2001 : l'odyssée de l'espace	Kubrick	1968	9
4	Blade Runner	Scott	1982	9

1. Déterminer l'ordre et la cardinal de la relation **Films**
2. Écrire le schéma relationnel de la table **Films**
3. Quelle requête permet de sélectionner tout les titres de la relation **Films**
4. Quel sera le résultat de la requête suivante **SELECT realisateur FROM Films WHERE annee\_sortie < 1980**
5. Quelle requête permet d'insérer dans la table **Films**, le film **JJK-0** de **Sung-ho Park** qui a une note de 8.

### Exercice 4 ★

*i* L'énoncé de cet exercice peut utiliser les mots du langage SQL suivants :  
SELECT, FROM, WHERE, COUNT, DISTINCT, AND, OR, AS, ORDER BY, ASC, DESC

Un site web recueille des données de navigation dans une base de données afin d'étudier les profils de ses visiteurs.

Chaque requête d'interrogation d'une page de ce site est enregistrée dans une première table dénommée **Visites** sous la forme d'un 5-uplet : (identifiant, adresse IP, date et heure de visite, nom de la page, navigateur).

Le chargement de la page `index.html` par `192.168.1.91` le 12 juillet 1998 à 22h48 aura par exemple été enregistré de la façon suivante :

`(1534, "192.168.1.91", "1998-07-12 22:48:00", "index.html", "Internet explorer 4.1")`.

1. Donner une commande d'interrogation en langage SQL permettant d'obtenir l'ensemble des 2-uplets (adresse IP, nom de la page) de cette table.
2. Donner une commande en langage SQL permettant d'obtenir l'ensemble des adresses IP ayant interrogé le site, **sans doublon**.
3. Donner une commande en langage SQL permettant d'obtenir la liste des noms des pages visitées par l'adresse IP `192.168.1.91`.
4. Donner une commande en langage SQL permettant d'obtenir la liste des noms des pages visitées et les ip des page visitée apres le 1 janvier 2020.

---

**Exercice 5** D'après le sujet Métropole 2024

---

Amélie souhaite organiser sa collection de CD. Elle se dit qu'elle doit pouvoir utiliser une base de données relationnelle pour organiser sa collection.

Voici un extrait de la table **Chanson**.

Chanson			
id	titre	album	groupe
1	Sunburn	Showbiz	Muse
2	Muscle Museum	Showbiz	Muse
3	Showbiz	Showbiz	Muse
4	New Born	Origin of Symmetry	Muse
5	Sing for Absolution	Absolution	Muse
6	Hysteria	Absolution	Muse
7	Welcome too the Jungle	Appetite for Destruction	Guns N' Roses
8	Muscle Museum	Hullabaloo	Muse
9	Showbiz	Hullabaloo	Muse

1. Donner l'ordre de la relation **Chanson**
2. Donner le cardinal de la relation **Chanson**
3. Donner le résultat de la requête suivante :

```
1 SELECT titre, album FROM Chanson WHERE groupe = 'Guns N'Roses';
```

4. Écrire une requête SQL permettant d'obtenir tous les titres des chansons de l'album **Showbiz** dans l'ordre croissant.
5. Écrire une requête SQL permettant d'ajouter la chanson dont le titre est **Megalomania** de l'album **Hullabaloo** du groupe **Muse**.

Amélie a remarqué une faute de frappe dans la chanson **Welcome too the Jungle** qui s'écrit normalement **Welcome to the Jungle**.

6. Écrire une requête SQL permettant de corriger cette erreur.
-