

Cours de bases de données,  
<http://sql.bdpedia.fr>

Création d'un schéma relationnel

# Création d'un schéma relationnel

La partie *Data Definition Language* (DDL) définit les commandes de création d'un schéma relationnel.

Dans cette session :

- La commande `create table`
- Contrainte de clé primaire `primary key`
- Contrainte de clé étrangère `foreign key`

**Ces diapositives correspondent au support en ligne disponible sur le site <http://sql.bdpedia.fr/>**

# La commande create table

Un premier exemple : la table Internaute.

```
create table Internaute (email varchar (40) not null,  
                           nom varchar (30) not null ,  
                           prénom varchar (30) not null,  
                           région varchar (30),  
                           primary key (email));
```

Quelques choix à effectuer : conventions de nommage, accents, etc.

# La contrainte not null

Les valeurs à `null` sont source de problèmes : on peut les interdire avec `not null` ou donner une valeur par défaut.

```
create table Cinéma (idCinéma integer not null,  
                    nom varchar (30) not null ,  
                    adresse varchar(255) default 'Inconnue',  
                    primary key (id));
```

Le SGBD rejettera alors toute tentative d'insérer un nuplet avec une valeur manquante.

# La clé primaire

Elle est spécifiée avec la clause `primary key`.

```
create table Pays (code varchar(4) not null,  
                  nom  varchar (30) not null,  
                  langue varchar (30) not null,  
                  primary key (code));
```

Il doit **toujours** y avoir une clé primaire dans une table et le système garantit l'unicité de ses valeurs.

# Clé primaire composée

Une clé primaire peut comprendre plusieurs attributs.

```
create table Notation (idFilm integer not null,  
                        email  varchar (40) not null,  
                        note   integer not null,  
                        primary key (idFilm, email));
```

Tous les attributs d'une clé primaire doivent être `not null`

# Clé primaire, clés "secondaires"

On peut définir d'autres clés avec la clause `unique`.

```
create table Artiste (idArtiste integer not null,  
                      nom varchar (30) not null,  
                      prénom varchar (30) not null,  
                      annéeNaiss integer,  
                      primary key (idArtiste),  
                      unique (nom, prénom))
```

Permet d'avoir un identifiant "abstrait", non modifiable, et d'ajouter des contraintes flexibles sur les attributs descriptifs.

# Clé étrangère

Elle est spécifiée avec la clause `foreign key`.

```
create table Film (idFilm integer not null,  
                  titre    varchar (50) not null,  
                  année    integer not null,  
                  idRéalisateur integer,  
                  genre    varchar (20) not null,  
                  résumé    varchar(255),  
                  codePays  varchar (4),  
                  primary key (idFilm),  
                  foreign key (idRéalisateur)  
                      references Artiste(idArtiste),  
                  foreign key (codePays)  
                      references Pays(code))
```

Si la clé étrangère est `not null` : association de type 1-n,  
sinon association de type 0-n.



# Actions liées clé primaire / étrangère

On peut associer les actions appliquées à un nuplet et aux nuplets qui le référencent.

```
create table Salle (idCinéma integer not null,  
no integer not null,  
capacité integer not null,  
primary key (idCinéma, noSalle),  
foreign key (idCinéma)  
references Cinéma(idCinéma)  
on delete cascade,  
on update cascade)
```

Utile en particulier pour les entités dites “faibles”

## À retenir

Après la phase de conception, la création du schéma ne présente aucune difficulté.

- Connaître les principaux types SQL (cf. support de cours)
- Connaître la commande `create table`
- Ajouter les contraintes de clés primaire et étrangères.

La spécification des contraintes n'est pas une lourdeur. Elle garantit que la base est saine, et évité beaucoup d'ennuis.