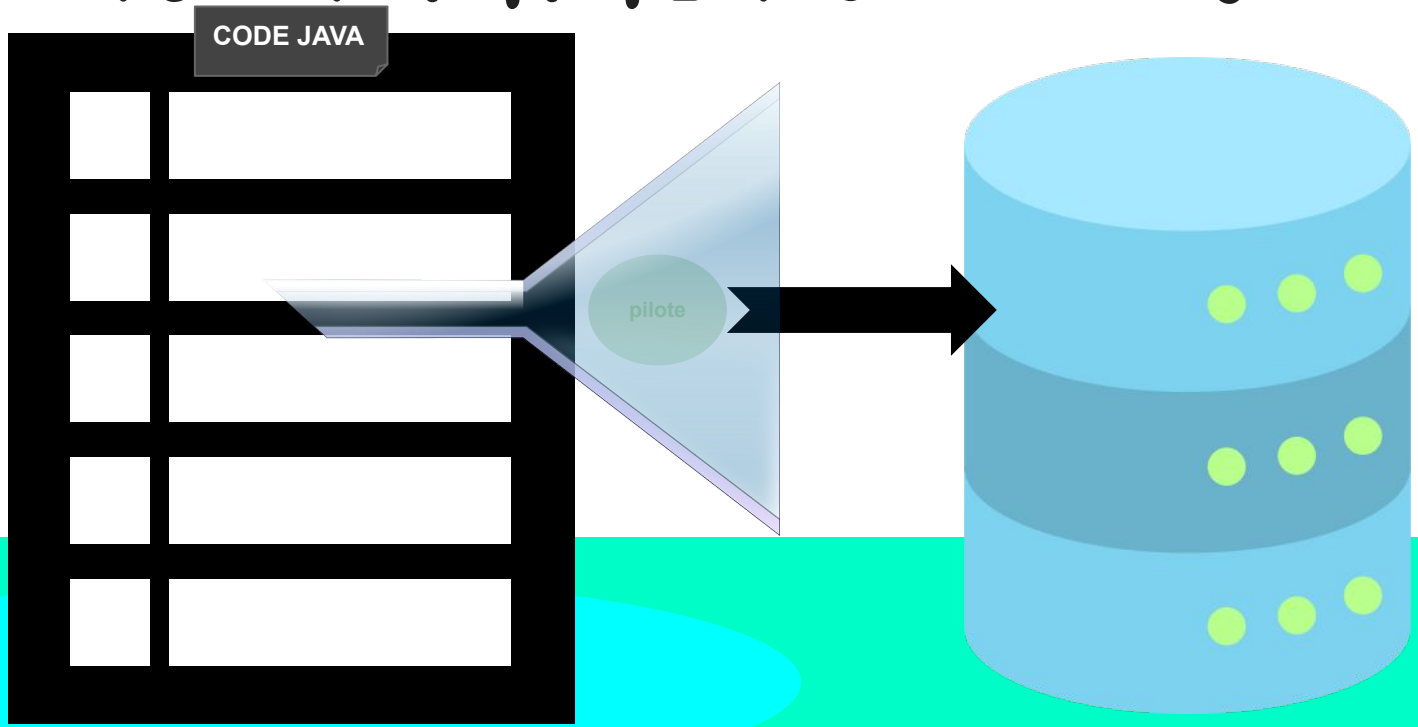
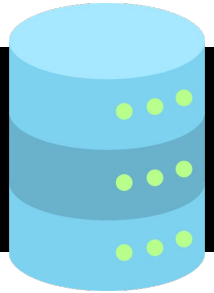
Programmer un DAO



sans les requêtes préparées

LES FONCTIONS de LECTURE



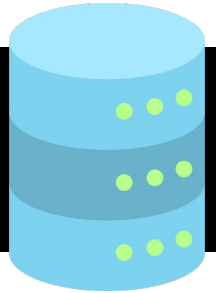
Les fonctions qui vont chercher des données et les rapportent sont celles qui varient le plus dans les DAOs.

On a les classiques :

- List<Truc> listerTrucs()
- Truc detaillerTruc()

Toutes ces fonctions rapportent quelque chose en valeur de retour.

LES FONCTIONS de LECTURE



Sur le plan technique, ces fonctions peuvent soit

➔ utiliser la fonction executeQuery

```
Statement requete = basededonnees.createStatement();
ResultSet curseur = requete.executeQuery(SQL_LISTER_FILMS);
curseur.next();
```

➔ ou utiliser la fonction prepareStatement

```
PreparedStatement requete = basededonnees.prepareStatement(SQL_LISTER_FILMS);
ResultSet curseur = requete.executeQuery();
curseur.next();
```

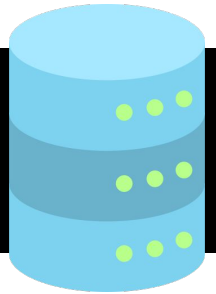
Programmer une fonction lister

La fonction veut rapporter une liste d'objets de données.

Elle doit :

- Déclarer et instancier une liste d'objets
- Chercher les données et les rapporter
  - Se connecter à la base de données
  - Faire exécuter son SELECT
  - Itérer dans le curseur et récupérer les données
  - Instancier les objets du modèle et ajouter à une liste
- Retourner la liste

La fonction DETAILLER

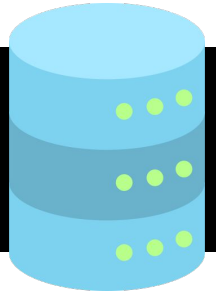


```
String SQL_DETAILLER_FILM = "SELECT * FROM film WHERE id = 5";
Film film = new Film();

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
Statement requete = connexion.createStatement();
ResultSet curseur = requete.executeQuery(SQL_DETAILLER_FILM);
curseur.next())
String titre = curseur.getString("titre");
film.setTitre(titre);

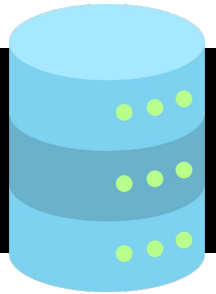
return film;
```

La fonction LISTER



```
String SQL_LISTER_FILMS = "SELECT * FROM film";
List<Film> films = new ArrayList<Film>();
Connection connexion = BaseDeDonnees.getInstance().getConnexion();
Statement requete = connexion.createStatement();
ResultSet curseur = requete.executeQuery(SQL_LISTER_FILMS);
while(curseur.next())
{
        String titre = curseur.getString("titre");
        Film film = new Film(titre);
        films.add(film);
}
return films;
```

LES FONCTIONS d'ECRITURE



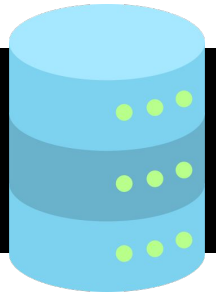
Les fonctions qui modifient les données sont presque toujours les même :

On a les classiques :

- void ajouterTruc(Truc)
- void editerTruc(Truc)
- void effacerTruc(id ou Truc)

Toutes ces fonctions recoivent des données à modifier en paramètre et ne retournent rien.

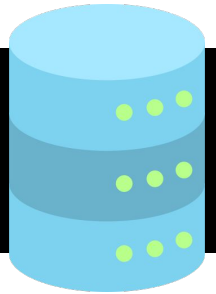
La fonction AJOUTER



```
String SQL_AJOUTER_FILM = "INSERT into FILM(titre, realisateur) VALUES('test','test')";

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
Statement commande = connexion.createStatement();
ResultSet curseur = commande.executeUpdate(SQL_AJOUTER_FILM);
```

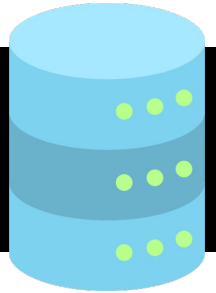
La fonction EDITER



```
String SQL_EDITER_FILM = "UPDATE film SET titre = 'test', realisateur = 'test' WHERE id = " + id;

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
Statement commande = connexion.createStatement();
ResultSet curseur = commande.executeUpdate(SQL_EDITER_FILM);
```

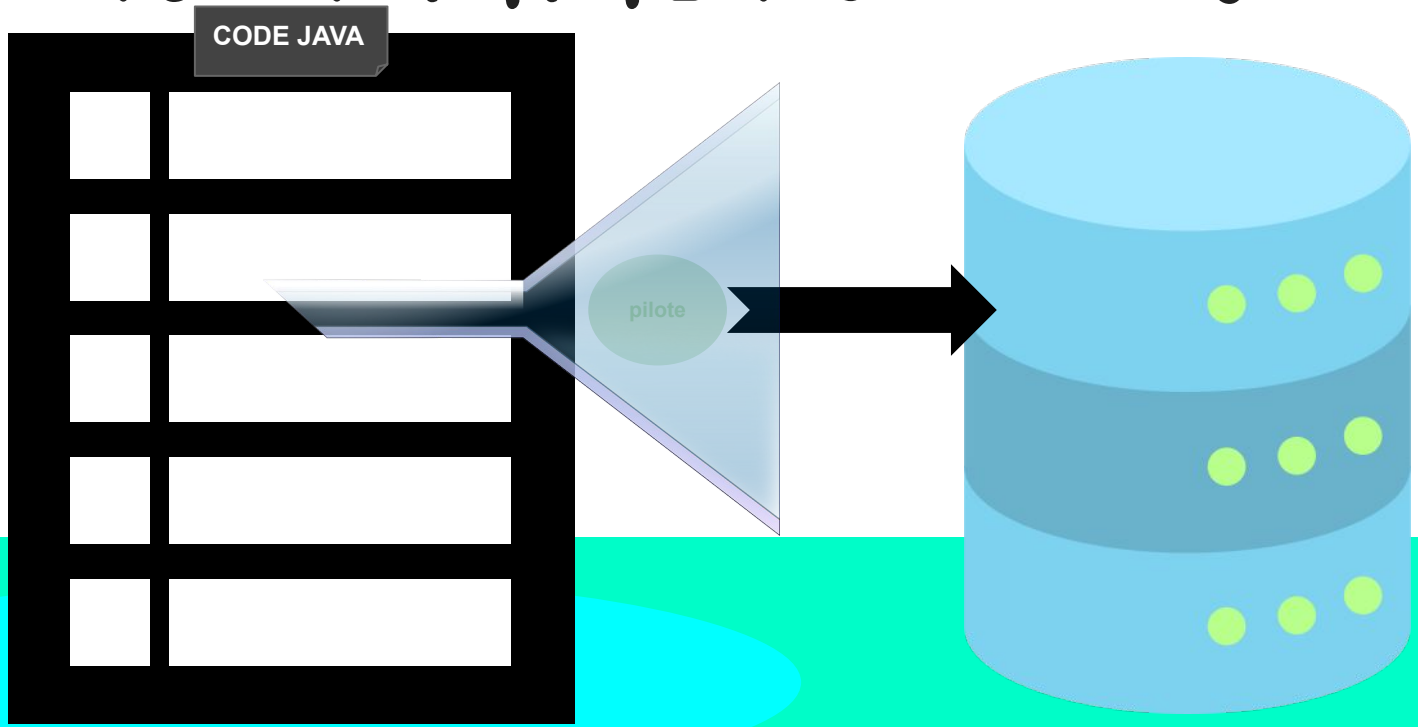
La fonction EFFACER



```
String SQL_EFFACER_FILM = "DELETE film WHERE id = " + id;

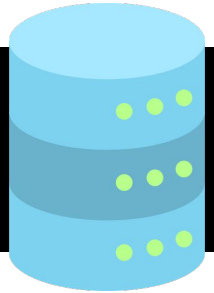
Connection connexion = BaseDeDonnees.getInstance().getConnexion();
Statement commande = connexion.createStatement();
ResultSet curseur = commande.executeUpdate(SQL_EFFACER_FILM);
```

Programmer un DAO



avec les requêtes préparées

LES FONCTIONS de LECTURE



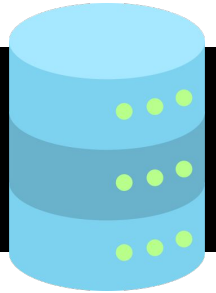
Les fonctions qui vont chercher des données et les rapportent sont celles qui varient le plus dans les DAOs.

On a les classiques :

- List<Truc> listerTrucs()
- Truc detaillerTruc()

Toutes ces fonctions rapportent quelque chose en valeur de retour.

LES FONCTIONS de LECTURE



Sur le plan technique, ces fonctions peuvent soit

➔ utiliser la fonction executeQuery

```
Statement requete = basededonnees.createStatement();
ResultSet curseur = requete.executeQuery(SQL_LISTER_FILMS);
curseur.next();
```

➔ ou utiliser la fonction prepareStatement

```
PreparedStatement requete = basededonnees.prepareStatement(SQL_LISTER_FILMS);
ResultSet curseur = requete.executeQuery();
curseur.next();
```

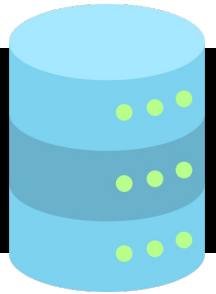
Programmer une fonction lister

La fonction veut rapporter une liste d'objets de données.

Elle doit :

- Déclarer et instancier une liste d'objets
- Chercher les données et les rapporter
  - Se connecter à la base de données
  - Faire exécuter son SELECT
  - Itérer dans le curseur et récupérer les données
  - Instancier les objets du modèle et ajouter à une liste
- Retourner la liste

La fonction DETAILLER

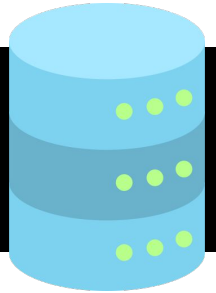


```
String SQL_DETAILLER_FILM = "SELECT * FROM film WHERE id = ?";
Film film = new Film();

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
PreparedStatement requete = connexion.prepareStatement(SQL_DETAILLER_FILM);
requete.setInt(1, 5);
ResultSet curseur = requete.executeQuery();
curseur.next();
String titre = curseur.getString("titre");
film.setTitre(titre);

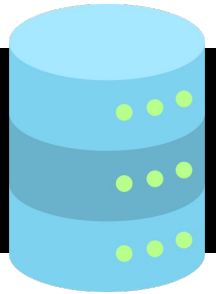
return film;
```

La fonction LISTER



```
String SQL_LISTER_FILMS = "SELECT * FROM film";
List<Film> films = new ArrayList<Film>();
Connection connexion = BaseDeDonnees.getInstance().getConnexion();
PreparedStatement requete = connexion.prepareStatement(SQL_LISTER_FILMS);
ResultSet curseur = requete.executeQuery();
while(curseur.next())
{
    String titre = curseur.getString("titre");
    Film film = new Film(titre);
    films.add(film);
}
return films;
```

LES FONCTIONS d'ECRITURE



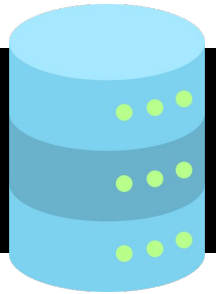
Les fonctions qui modifient les données sont presque toujours les même :

On a les classiques :

- void ajouterTruc(Truc)
- void editerTruc(Truc)
- void effacerTruc(id ou Truc)

Toutes ces fonctions recoivent des données à modifier en paramètre et ne retournent rien.

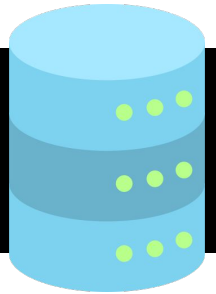
La fonction AJOUTER



```
String SQL_AJOUTER_FILM = "INSERT into FILM(titre, realisateur) VALUES(?,?)";

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
PreparedStatement commande = connexion.prepareStatement(SQL_AJOUTER_FILM);
commande.setString(1, "test");
commande.setString(2, "test");
int resultat = commande.executeUpdate();
```

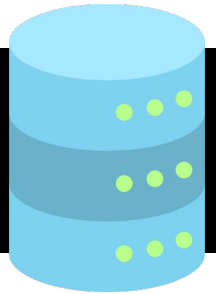
La fonction EDITER



```
String SQL_EDITER_FILM = "UPDATE film SET titre = ?, realisateur = ? WHERE id = ?";

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
PreparedStatement commande = connexion.prepareStatement(SQL_EDITER_FILM);
commande.setString(1, "test");
commande.setString(2, "test");
commande.setInt(3, id);
int resultat = commande.executeUpdate();
```

La fonction EFFACER



```
String SQL_EFFACER_FILM = "DELETE FROM film WHERE id = ?";

Connection connexion = BaseDeDonnees.getInstance().getConnexion();
PreparedStatement commande = connexion.prepareStatement(SQL_EFFACER_FILM);
commande.setInt(1, id);
int resultat = commande.executeUpdate();
```