

Bases de Données

– de la modélisation au SQL –
Requêtes SQL, révisions !

Notes de correction

Cette version contient la correction des exercices !

Objectifs de ce TP

Les objectifs de cette séance d'exercices sont multiples :

- proposer une révision approfondie de la commande `SELECT` ;
- découvrir ou redécouvrir une grande palette de ce qu'il est possible de faire avec chacune des clauses de cette commande ;
- s'écarter des requêtes stéréotypées qui consiste à conserver certaines colonnes de la classique jointure entre deux ou plusieurs tables avec comme condition de jointure l'égalité entre une clé étrangère et une clé primaire.

Il ne s'agit pas d'une séance de TP d'apprentissage de la commande `SELECT`. Cette séance est plutôt calibrée pour une deuxième approche en tant que révision ou approfondissement.

La première partie se focalise individuellement sur chacune des clauses de la commande `SELECT` : `SELECT`, `ORDER BY`, `WHERE`, `FROM`, `GROUP BY` et `HAVING`. Les exercices de cette première partie sont à réaliser sur une première base de données.

Ensuite, la seconde partie utilise une seconde base de données et a pour objectif l'écriture de requêtes plus générales.

style à chasse fixe pour les commandes, mots clés...

Première base de données

Soit le schéma relationnel suivant :

- `PERSONNE(idPersonne, nom, prénom)`
- `FILM(idFilm, idRéalisateur, titre, genre, année)` où `idRéalisateur` est une clé étrangère qui fait référence au schéma de relation `PERSONNE`
- `JOUER(idActeur, idFilm, rôle)` où `idActeur` et `idFilm` sont des clés étrangères qui font respectivement référence aux schémas de relation `PERSONNE` et `FILM`
- `CINÉMA(idCinéma, nom, adresse)`
- `PROJECTION(idCinéma, idFilm, jour)` où `idCinéma` et `idFilm` sont des clés étrangères qui font respectivement référence aux schémas de relation `CINÉMA` et `FILM`

Dans les exercices qui suivent, l'objectif est de trouver la requête SQL qui retourne ce qui est demandé.

x Il faut un lien pour télécharger la BD. Aura-t-on un espace où déposer des supports à télécharger avec une URL publique ?

La clause `SELECT`

1. Afficher l'ensemble de la table film.

Output de terminal en police à chasse fixe.

Extrait du résultat				
id_film	id_realisateur	titre	genre	annee_production
1	15	Crash	Drame	1996
2	15	Faux-Semblants	Epouvante	1988
...				

Ce cadre doit apparaître à la demande de l'apprenant pour l'aider

Aucun piège, la commande est triviale. L'utilisation de `*` dans la clause `SELECT` permet d'afficher toutes les colonnes.

idem. c'est une aide supplémentaire optionnelle à faire apparaître.

Cadre à faire apparaître pour voir la solution



Notes de correction

Style spécifique pour le code SQL avec coloration syntaxique.

```
SELECT * FROM film ;
```

2. Afficher les 3 premières lignes de la table film.

Extrait du résultat

id_film	id_realisateur	titre	genre	annee_production
1	15	Crash	Drame	1996
2	15	Faux-Semblants	Epouvante	1988
3	14	Pulp Fiction	Policier	1994

Aide supplémentaire

Il faut utiliser la clause LIMIT.

Notes de correction

```
SELECT * FROM film LIMIT 3 ;
```

3. Afficher les colonnes titre et genre de la table film.

Extrait du résultat

titre	genre
Crash	Drame
Faux-Semblants	Epouvante
...	

Aide supplémentaire

La projection (*i.e.* le choix) des colonnes se fait dans la clause SELECT.

Notes de correction

```
SELECT titre, genre FROM film ;
```

4. Afficher les genres des films sans doublon.

Extrait du résultat

genre
Epouvante
Western
...

Aide supplémentaire

On supprime les doublons en utilisant la combinaison SELECT DISTINCT.

Notes de correction

```
SELECT DISTINCT genre FROM film ;
```