

Projet de Bases de Données Relationnelles

Executive Master IASD

Enoncé

Le script¹ SQL, situé à la fin du sujet, modélise une base de données permettant de gérer les évaluations de séries télévisées.

Une série télévisée est décrite par un nom et est diffusée par une plateforme. Deux séries ne peuvent pas avoir le même nom. Un exemple d'instance (créée à partir du script situé en fin de sujet) de la relation `serie` est :

sid	nom	plateforme
1	Squid Game	Netflix
2	Validé	Canal +
3	Germinal	Salto
4	Game of Thrones	OCS
5	Profilage	Salto
6	The Crown	Netflix
7	Baron Noir	Canal +

Une personne est caractérisée par un nom, un prénom et un pseudo. Il n'est pas possible d'avoir 2 pseudos identiques. Un exemple d'instance (créée à partir du script situé en fin de sujet) de la relation `personne` est :

pid	nom	prenom	pseudo
1	Gamotte	Albert	AlGam
2	Zarela	Maude	mozza
3	Computing	Claude	cloud
4	Kontakt	Jessy	jess
5	Neymar	Jean	neymarJ

¹ Le script est incomplet, il manque toutes les contraintes !

Une personne peut être abonnée ou non à une plateforme. Un exemple d'instance (créée à partir du script situé en fin de sujet) de la relation `abonnes` est :

aid	pid	plateforme
1	1	Netflix
2	2	Canal +
3	2	Netflix
4	3	OCS
5	4	Slate
6	5	Netflix
7	5	Canal +
8	5	OCS

Une personne peut donner une note (entre 0 et 5) à chaque série, qu'elle soit abonnée ou non à la plateforme qui diffuse la série. Pour chaque série, la base enregistre la note moyenne des notes et le nombre de notes. Un exemple d'instance (créée à partir du script situé en fin de sujet) de la relation `evaluation` est :

eid	pid	sid	note
1	1	1	4
2	1	2	4
3	1	3	3
4	1	4	5
5	2	1	0
6	2	2	3
7	3	1	2
8	3	2	2
9	3	3	2
10	3	4	2
11	3	5	2
12	3	7	2
13	4	1	5
14	4	7	5
15	5	1	1
16	5	7	1
17	5	3	2

Travail à réaliser

Ce projet est à effectuer seul.e

Requêtes d'interrogation en SQL

Vous pouvez insérer, dans la base, des nuplets supplémentaires, pour tester les requêtes.

Pour connaître le temps d'exécution d'une requête vous pouvez utiliser la commande EXPLAIN ANALYSE (requête SELECT). Cette instruction affiche le plan d'exécution de la requête, avec à la fin de l'affichage le temps que le SGBD a mis pour optimiser et le temps pour exécuter la requête. Sous dbfiddle, pour calculer le temps, il faut lancer la requête seule (pour que les résultats des requêtes précédentes ne soient pas en mémoire), de préférence plusieurs fois, pour faire une moyenne des temps.

Ecrivez en SQL les requêtes suivantes² :

- a) Quelles plateformes n'ont pas d'abonnés ?
- b) Quelles personnes (en donnant son pseudo) a évalué une série de Netflix ou une série de Salto ? *Écrire 2 requêtes SQL, une avec UNION et une sans UNION et comparer leur temps d'exécution.*
- c) Quelles personnes (en donnant son pseudo) a évalué une série de Netflix **et** une série de Salto ? *Écrire 2 requêtes SQL, une avec INTERSECT et une sans INTERSECT et comparer leur temps d'exécution.*
- d) Quelles séries (en donnant leur nom) ont été évaluées par au moins 2 personnes ? *Écrire 3 requêtes SQL, une requête sans GROUP BY et sans EXISTS, une requête sans GROUP BY mais avec un EXISTS et une requête avec GROUP BY et sans EXISTS. Comparer leur temps d'exécution.*
- e) Quelles séries (en donnant leur nom) ont été évaluées par toutes les personnes de la base de données ? *Écrire 2 requêtes SQL, une requête sans GROUP BY et une requête avec GROUP BY et comparer leur temps d'exécution.*
- f) Écrire les différentes versions des requêtes a) à e) (sans tenir compte des requêtes écrites avec un GROUP BY) en Algèbre Relationnelle. *Pour simplifier vous pouvez écrire les opérateurs algébriques en français (sélection, projection, jointure, division etc.).*
- g) Créer une vue permettant d'obtenir pour chaque série (en précisant son nom) le nombre de personnes ayant noté la série et la note moyenne.

Les séries non notées doivent apparaître.

² Vous pouvez utiliser des vues pour décomposer vos requêtes en sous-requêtes si besoin.

Par exemple, sur les nuplets insérés par le script donné en fin de sujet, votre requête doit renvoyer :

nom	nbnotes	notemoyenne
Baron Noir	3	2.666666666666667
Game of Thrones	2	3.500000000000000
Germinal	3	2.333333333333333
Profilage	1	2.000000000000000
Squid Game	5	2.400000000000000
The Crown	0	null
Validé	3	3.000000000000000

- h) Quelles séries (en donnant leur nom) sont les moins bien notées ? (*Vous ne devez tenir compte que des séries ayant au moins une note dans la relation evaluation – vous pouvez utiliser la vue précédente*)
- i) Quel est le nombre de notes par série en ne tenant compte que des notes données par les abonnés de la plateforme diffusant la série ? *Par exemple, sur les nuplets insérés par le script donné en fin de sujet, votre requête doit renvoyer :*

plateforme	nom	nbnotesabonnes
OCS	Game of Thrones	1
Netflix	Squid Game	3
Canal +	Validé	1
Canal +	Baron Noir	1

- j) Quel est le nombre de notes par série en séparant les notes données par les abonnés de la plateforme diffusant la série et les notes des personnes non abonnées à la plateforme ? *Par exemple, sur les nuplets insérés par le script donné en fin de sujet, votre requête doit renvoyer :*

plateforme	nom	nbnotesabonnes	nbnotesnonabonnes
Netflix	Squid Game	3	2
Canal +	Validé	1	2
Sato	Germinal	0	3
OCS	Game of Thrones	1	1
Sato	Profilage	0	1
Netflix	The Crown	0	0
Canal +	Baron Noir	1	2

Place du Maréchal de Lattre
de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

Rendu du TP

Ce TP donnera lieu à un **script SQL** contenant :

- Les requêtes d'interrogation en SQL. *Merci d'indiquer svp en commentaire l'intitulé de chaque requête.*
- Les requêtes a) à e) en Algèbre Relationnelle.

Le **script SQL** devra être enregistré dans un **fichier**, dont le nom devra obligatoirement être de la forme **NomPrenom.sql** (i.e. en précisant les noms et prénoms des 2 membres du binôme), vos requêtes en Algèbre devront être sauvegardées de préférence dans un fichier pdf ou dans un fichier de type image dont le nom devra également comporté votre nom et prénom. Les fichiers **devront être déposés sur Moodle**.

Script SQL (incomplet) de création des relations de la base de données :

```
CREATE TABLE serie (  
    SID serial PRIMARY KEY,  
    nom varchar(25) NOT NULL UNIQUE,  
    plateforme varchar(25) NOT NULL  
);  
  
CREATE TABLE personne (  
    PID serial PRIMARY KEY,  
    nom varchar(25) NOT NULL,  
    prenom varchar(25) NOT NULL,  
    pseudo varchar(25) NOT NULL UNIQUE  
);  
  
CREATE TABLE evaluation (  
    EID serial PRIMARY KEY,  
    PID integer NOT NULL,  
    SID integer NOT NULL,  
    note integer NOT NULL DEFAULT 5,  
    CONSTRAINT FK1_Evaluation FOREIGN KEY(PID) REFERENCES personne(PID) ON  
DELETE CASCADE,  
    CONSTRAINT FK2_Evaluation FOREIGN KEY(SID) REFERENCES serie(SID),  
    CONSTRAINT UN_Evaluation UNIQUE(PID,SID),  
    CONSTRAINT CK_Evaluation CHECK (note >=0 AND note <=5)  
);  
  
CREATE TABLE abonnées (  
    AID serial PRIMARY KEY,  
    PID integer NOT NULL,  
    plateforme varchar(25) NOT NULL,  
    CONSTRAINT FK1_Abonnées FOREIGN KEY(PID) REFERENCES personne(PID) ON  
DELETE CASCADE,  
    CONSTRAINT UN_A UNIQUE(PID,plateforme)  
);
```

Place du Maréchal de Lattre
de Tassigny
75775 Paris Cedex 16
dauphine.psl.eu

```
INSERT INTO Serie(nom,plateforme) VALUES ('Squid Game','Netflix'),  
('Validé','Canal +'),('Germinal','Salto'),('Game of Thrones','OCS'),  
('Profilage','Salto'),('The Crown','Netflix'),('Baron Noir','Canal +');
```

```
INSERT INTO Personne(nom,prenom,pseudo) VALUES  
('Gamotte','Albert','AlGam'),('Zarela','Maude','mozza'),('Computing','Claude',  
'cloud'),('Kontakt','Jessy','jess'),('Neymar','Jean','neymarJ');
```

```
INSERT INTO evaluation (PID,SID,note) VALUES  
(1,1,4),(1,2,4),(1,3,3),(1,4,5),(2,1,0),(2,2,3),(3,1,2),(3,2,2),(3,3,2),(3,  
4,2),(3,5,2),(3,7,2),(4,1,5),(4,7,5),(5,1,1),(5,7,1),(5,3,2);
```

```
INSERT INTO abonnes(PID,plateforme) VALUES (1,'Netflix'), (2,'Canal  
+'), (2,'Netflix'), (3,'OCS'), (4,'Slato'), (5,'Netflix'), (5,'Canal  
+'), (5,'OSC');
```