

Examen final (Projet)

Résumé

Ce document est le projet associé à la note du module **Base de données**. Il reprend des éléments du cours et test vos connaissances en SQL et modélisation d'une base de données. La réponse à une question relative à des commandes SQL doit se faire en une seule requête.

1 Entreprise (4 points)

On considère la base “entreprise” qui contient les relations suivantes :

- EMPLOYE(ID_EMP, NOM, PRENOM, FONCTION, ID_CHEF, DATE_EMBAUCHE, SALAIRE, PRIME, ID_PROJET)
- PROJET (ID_PROJ, NOM, ID_CHEF_PROJET)

À vous !

- a) Écrire les commandes de création de ces deux tables en SQL sachant que :
 - les identificateurs (attributs ID_) sont des entiers de petite taille
 - les noms et prénoms sont tous des chaînes de 20 caractères
 - la fonction d'un employé est une chaîne de 7 caractères.
 - le salaire et prime sont en euro et centime d'euro, aucun salaire ne dépassant 40 000 euros
 - les attributs soulignés sont les clés primaires de la table correspondante
 - tous les attributs sauf les attributs ID_CHEF, PRIME et ID_PROJET doivent être renseignés.
- b) insérer l'ingénieur Pierre Dupont embauché le 3 mars 2010 avec un salaire de 3000
- c) insérer l'assistante Marie Salto embauchée le 5 mai 2010 avec comme directeur Pierre Dupont et un salaire de 1000 euros
- d) insérer le projet ATLAS qui a comme chef de projet Pierre Dupont ;
- e) donner à Marie Salto une prime de 50 euros
- f) augmenter le salaire de tout le monde de 5%
- g) effacer tous les projets
- h) effacer tous les employés qui ont comme directeur Pierre Dupont ;

2 Tennis (6 points)

On considère la base “tennis” d'un schéma le schéma suivantes :

- JOUEUR (NuJOUEUR, PRENOM, AnNaiss, NATIONALITE)
- RENCONTRE (NuGAGNANT, NuPERDANT, LIEUTOURNOI, ANNEE)

— GAIN (NuJOUEUR, LIEUTOURNOI, ANNEE, PRIME, NOMSPONSOR)

Voici le code associé pour la création des tables :

```
create table JOUEUR
( NuJOUEUR      int(8),
  NOM           varchar(12),
  PRENOM       varchar(14),
  AnNaiss      int(8),
  NATIONALITE   varchar(12));
```

```
create table RENCONTRE
( NuGAGNANT     int(8),
  NuPERDANT     int(8),
  LIEUTOURNOI   varchar(15),
  ANNEE        int(8));
```

```
create table GAIN
( NuJOUEUR      int(8),
  LIEUTOURNOI   varchar(15),
  ANNEE        int(8),
  PRIME        float(8),
  NOMSPONSOR    varchar(10));
```

À vous !

- Copier coller les commandes du fichier “base-tennis.sql” (fourni) pour créer vos tables et les peupler.
- Adapter les tables sachant que
 - les clés primaires sont en bleu soulignée
 - les clés étrangères sont en vert
 - on suppose que tous les joueurs, engagés dans un tournoi, touchent une prime.
- Écrire les requêtes en SQL pour observer le tournoi d’engagement (défini par le lieu et l’année) et le numéro des joueurs sponsorisés par Peugeot entre 1990 et 1994 ;
- Écrire les requêtes en SQL pour observer le nom et année de naissance des joueurs ayant participé à Roland Garros en 1994 ;
- Écrire les requêtes en SQL pour observer le nom et nationalité des joueurs ayant participé à la fois au tournoi de Roland Garros et à celui de Wimbledon, en 1992 ;
- Écrire les requêtes en SQL pour observer le nom et nationalité des joueurs ayant été sponsorisés par Peugeot et ayant gagné à Roland Garros au moins un match (avec un sponsor quelconque) ;
- Écrire les requêtes en SQL pour observer le nom des joueurs ayant toutes leurs primes à Roland Garros supérieures à 10.000 euros
- Écrire les requêtes en SQL pour observer les numéros des joueurs qui ont toujours gagné à Roland Garros ;

- i) Écrire les requêtes en SQL pour observer le nom des joueurs ayant participé à au moins deux tournois disputés en 1994 (vous pouvez utiliser la fonction `COUNT()`);
- j) Écrire les requêtes en SQL pour observer les numéros des joueurs ayant eu exactement deux sponsors;
- k) Écrire les requêtes en SQL pour observer les noms des joueurs qui ont gagné un tournoi, ou qui ont touché plus de 1000 euros à un tournoi;
- l) Écrire les requêtes en SQL pour observer la moyenne des primes gagnées par année ((vous pouvez utiliser la fonction `AVG()`);
- m) Écrire les requêtes en SQL pour observer la plus grande prime attribuée lors d'un tournoi en 1992, et noms des joueurs qui l'ont touchée (fonction `MAX()`);
- n) Écrire les requêtes en SQL pour observer la somme gagnée en 1992 par chaque joueur, pour l'ensemble des tournois auxquels il a participé (présentation par ordre de gain décroissant);
- o) Écrire les requêtes en SQL pour observer le nombre de joueurs ayant participé au tournoi de Wimbledon en 1993;

3 Cinéma (6 points)

Soit la base de données "cinéma" présentant les caractéristiques suivantes

- NUMF, NUMP, NOMC, NUMS sont des identifiants uniques (clés primaires) pour respectivement : FILM, PERSONNE, CINEMA, SALLE;
- tout nom de relation utilisé comme attribut est une clé étrangère qui renvoie à l'identifiant (clé primaire) de la relation correspondante (par exemple dans GENERIQUE, le champ NRFILM correspond à NUMF de FILM et est défini sur le même domaine);
- REALISATEUR, dans FILM, correspond à NOM dans PERSONNE;

Voici les commandes suivantes décrivant les tables (elles sont présentes dans le fichier "cinescope.sql" (fourni)

```
CREATE TABLE PERSONNE (NUMP INTEGER(5) AUTO_INCREMENT ,
                        PRENOM VARCHAR (20) NOT NULL ,
                        NOM VARCHAR (15),
                        DATENAIS DECIMAL (4,0) ,
                        PRIMARY KEY (NUMP));
```

```
CREATE TABLE FILM (NUMF DECIMAL(10,0) NOT NULL ,
                    TITRE VARCHAR (30),
                    ANNEE INTEGER(4),
                    REALISATEUR VARCHAR (15),
                    PRIMARY KEY (NUMF));
```

```
CREATE TABLE GENERIQUE( NRFILM    DECIMAL (10,0) NOT NULL ,
                           ACTEUR   VARCHAR (20) NOT NULL ,
                           ROLE     VARCHAR(20) NOT NULL ,
                           SALAIRE  INTEGER(10) ,
                           PRIMARY KEY (NRFILM , ACTEUR , ROLE)
                           );

CREATE TABLE CINEMA (NOMC   VARCHAR (20) NOT NULL ,
                      ARRONDISSEMENT DECIMAL (2,0) ,
                      ADRESSE VARCHAR (30) ,
                      COMPAGNIE VARCHAR (30) ,
                      PRIMARY KEY (NOMC));

CREATE TABLE PASSE (NUMF VARCHAR(10) NOT NULL ,
                     NOMCINE VARCHAR (20) NOT NULL ,
                     SALLE DECIMAL(3,0) NOT NULL ,
                     PRIMARY KEY (NUMF , NOMCINE , SALLE) ,
                     FOREIGN KEY (NOMCINE) REFERENCES CINEMA(NOMC));

CREATE TABLE SALLE (NUMS DECIMAL(3,0) NOT NULL ,
                     TAILLEECRAN INTEGER(10) NOT NULL ,
                     PRIMARY KEY (NUMS));
```

À vous !

- a) Proposer un schéma de modélisation (entité-association) présentant la base de données cinéma.
- b) Copier coller les commandes SQL du fichier “cinescope.sql” (fourni).
- c) Donner via une requête SQL les titres des films réalisés par Quentin Tarantino.
- d) Donner via une requête SQL le nombre de films tournés par Hitchcock.
- e) Donner via une requête SQL les noms et les prénoms d’acteurs qui ont joué dans des films réalisés par Tim Burton.
- f) Donner via une requête SQL les films qui ne passent dans aucun cinéma de la compagnie FOX.
- g) Trouver via une requête SQL le salaire total des acteurs du film ‘Pulp Fiction’.
- h) Pour chaque film de Tarantino, trouver via une requête SQL le nom et le prénom de l’acteur qui a eu le plus gros salaire pour un rôle dans ce film.
- i) Donner via une requête SQL les noms de cinémas dont la taille moyenne des écrans est supérieure à 40 mètres carré.

- j) Trouver via une requête SQL le nom et le prénom des acteurs qui ont eu un salaire plus important que la moyenne des salaires des autres acteurs (on suppose que chaque acteur a joué qu'un seul rôle).
- k) Donner le prénom, le nom et le numéro des acteurs qui ont joué dans tous les films de Burton via une requête SQL.

4 Monde (bonus, 3 points)

On se donne le schéma suivant d'une base de données MONDE :

- Pays (NumPays :entier, Nom :chaîne, RefPresident :entier, RefContinent :entier)
- Presidents (NumPresident :entier, Nom :chaîne, Prenom :chaîne, Age :entier, Periode :entier)
- Continents (NumContinent :entier, Nom :chaîne)
- Superficie (RefPays :entier, Valeur :réel)
- Population (RefPays :entier, Valeur :entier)

Cette base de données simpliste permet de modéliser quelques informations sur les pays du monde.

La table Presidents contient des informations sur les présidents en exercice.

La table Continents contient les noms des continents (ex. Europe, Afrique, Asie, etc.).

Les tables Superficie et Population donnent des informations supplémentaires sur les pays.

L'attribut Periode représente la durée en nombre d'années du mandat présidentiel.

À vous !

- a) Proposer un schéma entité-relation représentant la base de données MONDE
- b) Dans quel ordre doit on créer les tables dans la base de données ?
- c) Donner les commandes SQL pour la création des tables (+les clés primaires et étrangères). Vos tables doivent intégrer les commandes présentent dans le fichier "monde.sql".
- d) copier coller les commandes permettant de remplir les tables.
- e) Sur quelles colonnes est-il intéressant de créer des index ? Justifier.
- f) Donner les commandes SQL pour connaître les noms des pays qui ont eu le même président plus d'une fois.
- g) Donner les commandes SQL pour connaître les présidents qui ont moins de 40 ans.
- h) Donner les commandes SQL pour connaître le nom du pays d'Europe le plus peuplé.
- i) Donner les commandes SQL pour connaître la superficie totale de l'Europe.

- j) Donner les commandes SQL pour connaître la somme totale du nombre de personnes habitant dans un continent d'une superficie < 15000000 km².
- k) Donner les commandes SQL pour compter le nombre de pays d'Afrique dont la population est > 50000000 d'habitants.
- l) Donner les commandes SQL pour connaître la liste de tous les pays dont la densité (= population / superficie) est inférieure à 20.