

QCM SQL et Langage objet

Alexandre LAIRAN

October 31, 2023

Durée : 45 minutes

Nom : _____

Instructions

Avant de commencer, veuillez prendre un moment pour lire les instructions ci-dessous afin de comprendre comment le questionnaire sera évalué.

Ce QCM est conçu pour évaluer vos connaissances sur le sujet en question. Il y a plus de 20 points à obtenir. La notation sera entre 0 et 20.

Répondez aux questions suivantes en cochant la ou les bonnes réponses.

Le QCM durera 45 minutes.

Notation

Pour chaque question à laquelle vous répondez correctement, vous obtiendrez la totalité des points attribués à cette question.

Si une question requiert plusieurs réponses, et que vous en sélectionnez certaines correctement, vous obtiendrez un pourcentage des points basé sur le nombre de bonnes réponses que vous avez fournies.

Toutefois, soyez prudent : pour chaque mauvaise réponse, 0,25 points seront retirés de votre score total. Si vous n'êtes pas sûr de votre réponse, il pourrait être préférable de ne pas répondre afin d'éviter une pénalité.

Conseils

Lisez chaque question attentivement.

Assurez-vous de bien comprendre ce qui est demandé avant de cocher vos réponses.

Si vous n'êtes pas certain d'une réponse, il est peut-être préférable de passer à la question suivante et de revenir plus tard.

Questions

1. Quel est le but principal du langage SQL? /0.5

- ☐ Création de graphiques
- ☐ Manipulation et requête de bases de données
- ☐ Développement d'applications web
- ☐ Création de jeux vidéo

2. Quelle instruction SQL est utilisée pour extraire des données d'une base de données? /0.5

- ☐ INSERT
- ☐ UPDATE
- ☐ SELECT
- ☐ DELETE

3. Quelle commande SQL est utilisée pour supprimer une table? /0.5

- ☐ REMOVE TABLE
- ☐ DELETE TABLE
- ☐ DROP TABLE
- ☐ ERASE TABLE

4. Si vous voulez sélectionner toutes les colonnes d'une table appelée "Étudiants", quelle requête utiliseriez-vous? /0.5

- ☐ SELECT ALL FROM Étudiants;
- ☐ SELECT * FROM Étudiants;
- ☐ SELECT Étudiants FROM ALL;
- ☐ GET * FROM Étudiants;

5. Quelle instruction SQL est utilisée pour insérer de nouvelles données dans une table? /0.5

- ☐ ADD
- ☐ PUSH
- ☐ PUT INTO
- ☐ INSERT INTO

6. Comment pouvez-vous empêcher l'affichage des doublons dans un résultat de requête SELECT? /0.5

- ☐ USE DISTINCT
- ☐ NO DUPLICATES
- ☐ UNIQUE
- ☐ SELECT DISTINCT

7. Laquelle de ces instructions est utilisée pour modifier les données dans une table existante? /0.5

- ☐ MODIFY
- ☐ ALTER
- ☐ CHANGE
- ☐ UPDATE

8. Quelle commande SQL permet de créer une nouvelle table? /0.5

- ☐ CREATE NEW TABLE
- ☐ MAKE TABLE
- ☐ NEW TABLE
- ☐ CREATE TABLE

9. Si vous voulez extraire toutes les colonnes d'une table "Étudiants" où le champ "Nom" est égal à "Dupont", quelle requête SQL utiliseriez-vous? /1

- ☐ SELECT * FROM Étudiants WHERE Nom = "Dupont";
- ☐ GET * FROM Étudiants WHERE Nom = "Dupont";
- ☐ SELECT * FROM Étudiants WHERE Nom == "Dupont";
- ☐ SELECT * FROM Étudiants IF Nom = "Dupont";

10. Quelle est la commande SQL pour renommer une table? /1

- ☐ RENAME
- ☐ RENAME TABLE
- ☐ ALTER TABLE ... RENAME TO
- ☐ MODIFY TABLE

11. Qu'est-ce que JDBC en Java? /0.5

- ☐ Un framework Java pour le développement web.
- ☐ Une API Java pour interagir avec des bases de données via SQL.
- ☐ Une bibliothèque Java pour la création d'interfaces graphiques.
- ☐ Un outil de déploiement Java pour les applications mobiles.

12. Que fait un "query builder" en programmation? /1

- ☐ Il génère automatiquement des requêtes SQL à partir de commandes dans un langage de programmation.
- ☐ Il construit des interfaces utilisateur pour les applications web.
- ☐ Il génère des tests unitaires pour les requêtes SQL.
- ☐ Il optimise les requêtes SQL pour une meilleure performance.

13. Dans le contexte d'un ORM, qu'est-ce qu'un "repository"? /1

- ☐ Un lieu où le code source est stocké et versionné.
- ☐ Une classe ou un ensemble de classes permettant d'accéder aux données et d'encapsuler les requêtes vers la base de données.
- ☐ Un outil pour visualiser les schémas de base de données.
- ☐ Un outil pour optimiser les performances d'une application.

14. Quelle est la principale utilisation de la gem 'pg' en Ruby? /0.5

- ☐ Créer des graphiques et des visualisations.
- ☐ Interagir avec des bases de données PostgreSQL.
- ☐ Gérer des processus parallèles et des threads.
- ☐ Fournir des outils de débogage pour Ruby on Rails.

15. Quel est l'avantage principal d'utiliser un "query builder" par rapport à l'écriture directe de requêtes SQL? /2

- ☐ Les requêtes générées sont généralement plus optimisées pour la performance.
- ☐ Il est plus facile de créer des interfaces utilisateur.
- ☐ Les requêtes sont souvent plus sécurisées et évitent les injections SQL.
- ☐ Les bases de données peuvent être facilement changées sans modification du code.
- ☐ Cela permet l'auto-complétion de l'IDE.

16. Comment la gem 'pg' protège-t-elle contre les injections SQL dans Ruby? /1

- ☐ En utilisant des sessions sécurisées.
- ☐ En utilisant des requêtes préparées (prepared statements).
- ☐ En cryptant toutes les données envoyées à la base de données.
- ☐ Elle ne fournit aucune protection; c'est à l'utilisateur de le gérer.

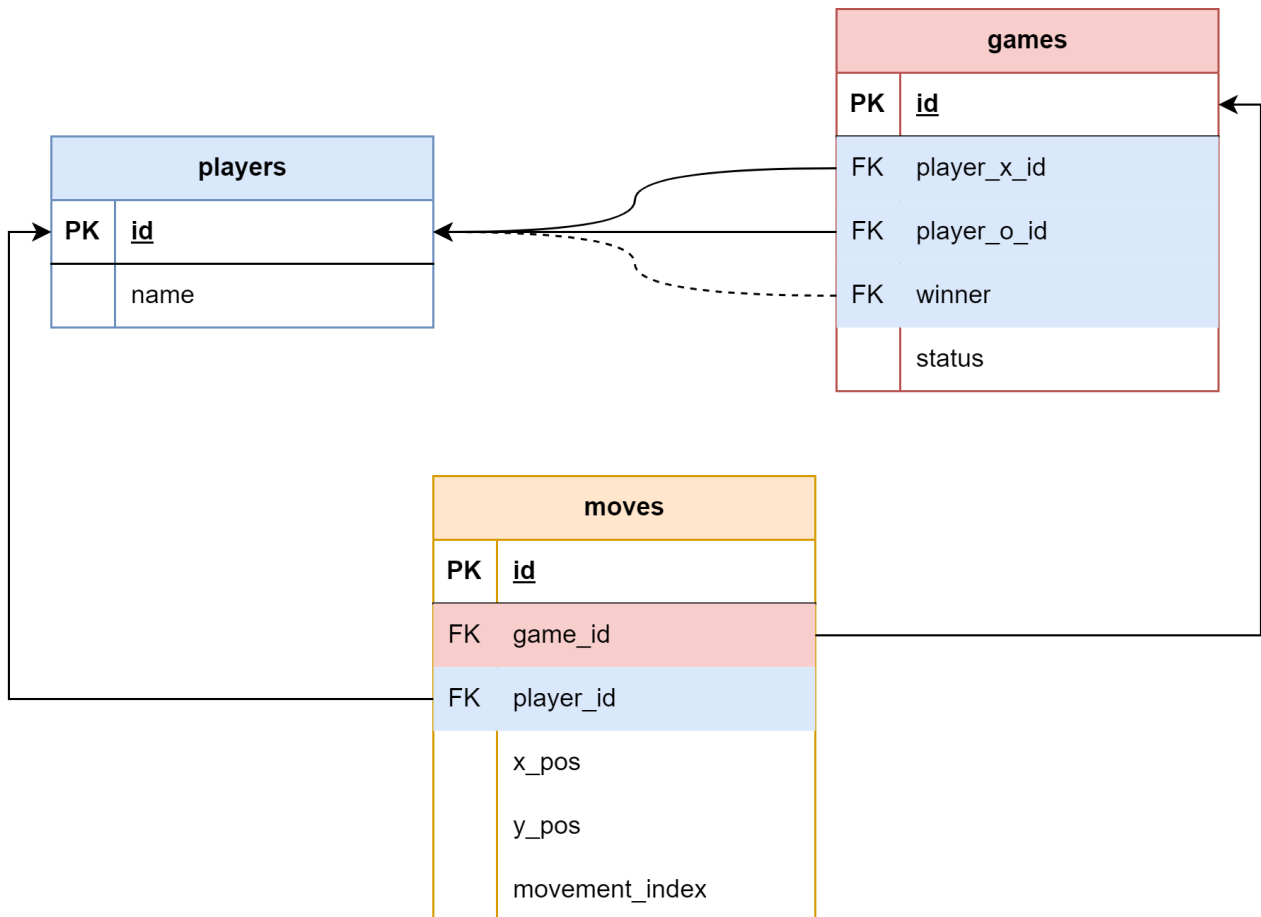


Figure 1: Modèle Conceptuel de Données - Morpion

Pour toutes les questions ci-dessous, referez-vous au MCD du morpion, ainsi qu'à l'annexe.

17. Quelles commandes SQL pour créer la table game? /2

☐

```
CREATE TABLE games (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  player_x_id UUID NOT NULL REFERENCES players(id),
  player_o_id UUID NOT NULL REFERENCES players(id),
  winner UUID REFERENCES players(id) NULL,
  status TEXT CHECK (status IN ('ongoing', 'finished')) NOT NULL
);
```

☐

```
CREATE TABLE games (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  player_x_id UUID NOT NULL,  
  player_o_id UUID NOT NULL,  
  winner UUID NULL,  
  status TEXT CHECK (status IN ('ongoing', 'finished')) NOT NULL  
);  
ALTER TABLE games  
ADD CONSTRAINT games_player_x_id_fkey FOREIGN KEY (player_x_id) REFERENCES  
  players(id);  
  
ALTER TABLE games  
ADD CONSTRAINT games_player_o_id_fkey FOREIGN KEY (player_o_id) REFERENCES  
  players(id);  
  
ALTER TABLE games  
ADD CONSTRAINT games_winner_fkey FOREIGN KEY (winner) REFERENCES players(id);
```

☐

```
CREATE TABLE games (  
  id UUID PRIMARY KEY,  
  player_x_id UUID NOT NULL,  
  player_o_id UUID NOT NULL,  
  winner UUID NULL,  
  status TEXT CHECK (status IN ('ongoing', 'finished')) NOT NULL  
);
```

☐

```
CREATE TABLE games (  
  id STRING PRIMARY KEY,  
  player_x_id STRING NOT NULL,  
  player_o_id STRING NOT NULL,  
  winner STRING NULL,  
  status TEXT CHECK (status IN ('ongoing', 'finished')) NOT NULL  
);
```

☐

```
CREATE TABLE games (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  player_x_id UUID NOT NULL REFERENCES players(id),  
  player_o_id UUID NOT NULL REFERENCES players(id),  
  winner UUID REFERENCES players(id) NULL  
);
```

18. Quelles tables ne contient pas directement d'information sur les positions (x et y) d'un mouvement dans un jeu ? /1

☐ players

☐ games

☐ moves

19. Quelles contraintes garantit qu'un mouvement est associé à un jeu spécifique ? /1

- ☐ game_id UUID NOT NULL
- ☐ UNIQUE (game_id, movement_index)
- ☐ id UUID PRIMARY KEY DEFAULT uuid_generate_v4()
- ☐ x_pos INTEGER NOT NULL

20. Est-il possible pour un jeu d'avoir le même joueur comme 'player_x' et 'player_o' ? /1

- ☐ Oui, car il n'y a aucune contrainte empêchant cela.
- ☐ Non, car il y a une contrainte 'CHECK' empêchant cela.
- ☐ Oui, mais seulement si le champ 'winner' est NULL.
- ☐ Non, car un joueur ne peut jouer qu'une seule fois.

21. Dans la table 'games', quelle est la contrainte imposée sur le champ 'status' ? /1

- ☐ Le champ 'status' doit être soit 'ongoing', soit 'finished'.
- ☐ Le champ 'status' doit être soit 'player_x', soit 'player_o'.
- ☐ Le champ 'status' doit être un UUID valide.
- ☐ Il n'y a pas de contrainte imposée sur le champ 'status'.

22. Quelle est la fonction de 'uuid_generate_v4()' dans la définition de la table ? /1

- ☐ Elle génère un identifiant unique basé sur l'heure actuelle.
- ☐ Elle génère un nombre aléatoire entre 1 et 4.
- ☐ Elle génère un identifiant unique basé sur l'adresse IP du serveur.
- ☐ Elle génère un identifiant unique sous forme d'UUID version 4.

23. Quelle requête SQL permet d'obtenir le nom du joueur 'X' et du joueur 'O' pour chaque jeu enregistré dans la table 'games' ? /1

☐

```
|| SELECT p1.name, p2.name
|| FROM games
|| JOIN players p1 ON games.player_x_id = p1.id
|| JOIN players p2 ON games.player_o_id = p2.id;
```

☐

```
|| SELECT player_x_id, player_o_id FROM games;
```

☐

```
|| SELECT name FROM players WHERE id IN (SELECT player_x_id, player_o_id FROM games);
```

☐

```
|| SELECT name FROM players
|| JOIN games ON players.id = games.player_x_id
|| OR players.id = games.player_o_id;
```


24. Si on souhaite créer une vue 'v_active_games' qui liste tous les jeux en cours ('status' égal à 'ongoing') avec les noms des joueurs, quelle serait la requête SQL appropriée ? /2

☐

```
CREATE VIEW v_active_games AS
SELECT p1.name AS Player_X, p2.name AS Player_O
FROM games
JOIN players p1 ON games.player_x_id = p1.id
JOIN players p2 ON games.player_o_id = p2.id
WHERE status = 'ongoing';
```

☐

```
SELECT p1.name, p2.name
FROM games
JOIN players p1 ON games.player_x_id = p1.id
JOIN players p2 ON games.player_o_id = p2.id
WHERE status = 'ongoing';
```

☐

```
CREATE VIEW v_active_games AS
SELECT player_x_id, player_o_id
FROM games
WHERE status = 'finished';
```

☐

```
CREATE TABLE v_active_games AS
SELECT p1.name, p2.name
FROM games
JOIN players p1 ON games.player_x_id = p1.id
JOIN players p2 ON games.player_o_id = p2.id
WHERE status = 'ongoing';
```

☐

```
SELECT * FROM v_active_games
WHERE status = 'ongoing';
```

25. Qu'est-ce qu'une window function en SQL? /1

☐ Une fonction qui ouvre une nouvelle fenêtre dans l'interface utilisateur de la base de données.

☐ Une fonction qui permet d'effectuer des calculs sur un ensemble de lignes de table qui sont d'une certaine manière liées à la ligne courante, sans changer la cardinalité du résultat.

☐ Une fonction qui permet d'appliquer une transformation graphique à l'affichage des données, telle qu'une fenêtre pop-up.

☐ Une fonction qui limite le nombre de résultats retournés par une requête à une fenêtre de temps spécifique.

☐ Une fonction qui active le mode "fenêtre" de la base de données, offrant une interface utilisateur améliorée.

26. Quelle est la fonction principale de l'opérateur 'GROUP BY' en SQL? /1

- ☐ Regrouper les résultats d'une requête par une ou plusieurs colonnes pour permettre des agrégations comme 'SUM', 'AVG', etc.
- ☐ Trier les résultats d'une requête en fonction de l'ordre alphabétique ou numérique d'une colonne.
- ☐ Filtrer les résultats d'une requête en fonction d'une condition spécifiée.
- ☐ Joindre deux tables ou plus en se basant sur une ou plusieurs colonnes communes.
- ☐ Dupliquer les lignes d'une table en fonction d'une colonne spécifiée.

27. Quelle est la syntaxe correcte pour ordonner les résultats d'une requête SQL en fonction de la colonne 'name' dans un ordre décroissant? /2

- ☐ SELECT * FROM players ORDER BY name DESC;
- ☐ SELECT * FROM players SORT BY name DESCENDING;
- ☐ SELECT * FROM players ORDER name DESC;
- ☐ SELECT * FROM players WHERE ORDER BY name DESC;
- ☐ SELECT * FROM players SORT BY name DESC;
- ☐ SELECT * FROM players ORDER BY name DESCENDING;

28. Considérez les données suivantes dans la table employees: /2

| id | name | birth_date | salary |
|----|---------|------------|--------|
| 1 | Alice | 1990-02-15 | 5000 |
| 2 | Bob | 1985-10-23 | 6000 |
| 3 | Charlie | 1995-05-28 | 4500 |
| 4 | David | 1992-11-14 | 6200 |
| 5 | Eve | 1993-04-03 | 5700 |

Quelle requête retourne le nombre total d'employés et le salaire moyen, ainsi que le nom de l'employé le plus âgé et le plus jeune (en fonction de la date de naissance) ?

☐

```
SELECT COUNT(id) as Total_Employees, AVG(salary) as Average_Salary,  
MAX(name) OVER (ORDER BY birth_date ASC) as Oldest_Employee,  
MIN(name) OVER (ORDER BY birth_date DESC) as Youngest_Employee  
FROM employees;
```

☐

```
SELECT COUNT(id) as Total_Employees, AVG(salary) as Average_Salary,  
name WHERE birth_date = MIN(birth_date),  
name WHERE birth_date = MAX(birth_date)  
FROM employees;
```

☐

```
SELECT COUNT(id) as Total_Employees, AVG(salary) as Average_Salary,  
FIRST_VALUE(name) OVER (ORDER BY birth_date ASC) as Oldest_Employee,  
FIRST_VALUE(name) OVER (ORDER BY birth_date DESC) as Youngest_Employee  
FROM employees;
```

☐

```
SELECT COUNT(id) as Total_Employees, SUM(salary)/COUNT(id) as Average_Salary,  
MAX(name) FROM employees WHERE birth_date = MIN(birth_date),  
MAX(name) FROM employees WHERE birth_date = MAX(birth_date)  
FROM employees;
```

A Annexe : Schéma de la base de données

```
tictactoe=# \d players
      Table "public.players"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
id      | uuid |           | not null | uuid_generate_v4()
name    | text |           | not null |
Indexes:
"players_pkey" PRIMARY KEY, btree (id)
Referenced by:
TABLE "games" CONSTRAINT "games_player_o_id_fkey" FOREIGN KEY (player_o_id) REFERENCES
    ↳ players(id)
TABLE "games" CONSTRAINT "games_player_x_id_fkey" FOREIGN KEY (player_x_id) REFERENCES
    ↳ players(id)
TABLE "games" CONSTRAINT "games_winner_fkey" FOREIGN KEY (winner) REFERENCES players(id)

tictactoe=# \d games
      Table "public.games"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
id      | uuid |           | not null | uuid_generate_v4()
player_x_id | uuid |           | not null |
player_o_id | uuid |           | not null |
winner    | uuid |           |          |
status     | text |           | not null |
Indexes:
"games_pkey" PRIMARY KEY, btree (id)
Check constraints:
"games_status_check" CHECK (status = ANY (ARRAY['ongoing'::text, 'finished'::text]))
Foreign-key constraints:
"games_player_o_id_fkey" FOREIGN KEY (player_o_id) REFERENCES players(id)
"games_player_x_id_fkey" FOREIGN KEY (player_x_id) REFERENCES players(id)
"games_winner_fkey" FOREIGN KEY (winner) REFERENCES players(id)
Referenced by:
TABLE "moves" CONSTRAINT "moves_game_id_fkey" FOREIGN KEY (game_id) REFERENCES games(id)

tictactoe=# \d moves
      Table "public.moves"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
id      | uuid |           | not null | uuid_generate_v4()
game_id | uuid |           | not null |
x_pos   | integer |           | not null |
y_pos   | integer |           | not null |
movement_index | integer |           | not null |
Indexes:
"moves_pkey" PRIMARY KEY, btree (id)
"moves_game_id_movement_index_key" UNIQUE CONSTRAINT, btree (game_id, movement_index)
Foreign-key constraints:
"moves_game_id_fkey" FOREIGN KEY (game_id) REFERENCES games(id)
```