

Corrigé — SGBD · SQL · Exercices

Source : « SGBD_Langage_SQL_Exercices.pdf ». Les libellés des questions et les données des tables *Agences* et *Voitures* proviennent de l'énoncé.

Exercice 1 — SGBD / QCM

- Q1 **1970 (b)** — C.E. Codd publie le modèle relationnel en 1970.
 - Q2 **Une requête (b)** — C'est l'instruction envoyée au SGBD pour interroger.
 - Q3 **Un langage pour interagir avec une base (b)** — SQL permet de définir, manipuler et interroger les données.
 - Q4 **Le modèle relationnel (a)** — C'est le paradigme dominant dans les SGBD classiques.
-

Exercice 2 — SQL / QCM

- Q1 **INSERT INTO (c)** — Forme canonique pour ajouter une ligne.
 - Q2 **SELECT (d)** — Extraction de lignes d'une table.
 - Q3 **SELECT * FROM Eleves; (c)** — L'astérisque sélectionne toutes les colonnes.
-

Exercice 3 — SQL avancé / QCM

- Q1 **ORDER BY (c)** — Tri (avec ASC / DESC).
 - Q2 **Plusieurs tables (c)** — Les jointures combinent n tables (au moins 2).
 - Q3 **WHERE (d)** — Introduit des conditions de sélection.
-

Exercice 4 — Modélisation « Agence de location de voitures »

Données fournies dans le document : tables *Agences*(*id_agence*, *Lieu*, *Département*) et *Voitures*(*id_voiture*, *Marque*, *Modèle*, *Kms*, *Couleur*, *id_agence*). :contentReference[oaicite:1]{index=1}

1) Justifier l'existence de *id_agence* dans *Agences*

Justification : *id_agence* est une clé primaire numérique qui identifie *sans ambiguïté* chaque agence, même si des attributs comme *Lieu* sont dupliqués (ex. deux agences à Paris). Elle sert aussi de *clé étrangère* dans *Voitures* pour relier chaque véhicule à son agence (intégrité référentielle).

2) Requête : lieux où des agences sont présentes

```
SELECT DISTINCT Lieu
FROM Agences
ORDER BY Lieu ASC;
```

3) Requête : véhicules avec moins de 15 000 kms

```
SELECT id_voiture, Marque, Modèle, Kms, Couleur, id_agence
FROM Voitures
WHERE Kms < 15000
ORDER BY Kms ASC;
```

4) (*) Requête : véhicules présents dans au moins l'une des agences parisiennes

Par une jointure :

```
SELECT v.id_voiture, v.Marque, v.Modèle, v.Kms, v.Couleur, v.id_agence
FROM Voitures AS v
JOIN Agences AS a ON a.id_agence = v.id_agence
WHERE a.Lieu = 'Paris'
ORDER BY v.id_voiture;
```

Résultat attendu (d'après les tables de l'énoncé)

id_voiture	Marque	Modèle	Kms	Couleur	id_agence
1	Renault	Clio	12000	Rouge	1
3	Peugeot	205	20000	Noir	1
2	Dacia	Sandero	11500	Blanche	3

Les agences de Paris ont les identifiants 1 et 3 dans les données de l'énoncé, d'où la sélection des véhicules rattachés à ces id_agence .

Rappels utiles (syntaxe)

- Tri : ORDER BY colonne ASC|DESC
- Déduplication : SELECT DISTINCT colonne ...
- Filtre : WHERE condition (ex. Kms < 15000)
- Jointure équivalente : JOIN ... ON ... ⇔ WHERE avec égalité de clés