

Partiel Bases de Données

2025

Durée : 2h Une feuille A4 manuscrite avec des notes de cours/TD/TP autorisée, calculatrice interdite

Exercice 1

Dans cet exercice, on désire modéliser de façon très simplifiée le système d'une base de données pour gérer les ventes de voitures de magasins. La base de données gère les achats de véhicules d'occasion et les ventes de véhicules à des clients. Les vendeurs sont des employés affectés à un seul magasin. La base de données gère aussi les transferts de véhicules d'un magasin à un autre magasin.

Les clés primaires sont soulignées et les clés étrangères sont précédées par un caractère "#".

*La convention de nommage (typographique) utilisée est le **snake_case**, il y a un "_" entre les mots dans les noms des attributs (colonnes) d'une table et certains noms de table.*

Soit le modèle relationnel suivant :

achat_client = (#id_voiture, #id_client, #id_vendeur, date_achat, prix_achat, reprise);
vente_magasin = (#id_voiture, #id_client, #id_vendeur, date_vente, prix_vente, duree_garantie);
transfert = (id_transfert, date_depart, date_arrivee, #id_magasin_depart, #id_magasin_arrivee, #id_voiture);
modele = (id_modele, marque, denomination);
magasin = (id_magasin, nom_magasin, adresse, ville);
client = (id_client, nom, prenom);
vendeur = (id_vendeur, nom, prenom, #id_magasin);
voiture = (id_voiture, num_serie, couleur, prix_voiture, cout_voiture, #id_modele);

* **transfert** est une table pour saisir les informations d'un transfert de véhicule d'un magasin à un autre magasin.

* **achat_client** est une table pour saisir les informations sur l'achat d'un véhicule à un client par un magasin (reprise).

* **reprise** est un booléen qui permet de définir si un véhicule a été racheté dans le cas de reprise pour une vente de véhicule neuf.

* **cout_voiture** est un numérique qui représente le coût de véhicule : prix d'achat + entretiens voir réparations pour remettre le véhicule en état de vente.

* **duree_garantie** est un numérique qui représente la durée de garantie d'un véhicule en mois.

Question 1

Proposer un modèle conceptuel des données associé au modèle relationnel précédent : modèle relationnel complet. **Utiliser le document réponse.**

Ne pas utiliser d'identifiant relatif sauf si il n'y a pas d'autre solution dans une relation binaire.

Question 2

* Proposer le script de création de la table «vente_magasin» dans un SGBD respectant strictement la norme SQL avec des contraintes d'**intégrité référentielle** sur les autres Tables.

* hypothèse préalable : avant de lancer l'exécution du script de creation de cette table, quelles sont les tables à créer **avant au minimum ? Indiquer l'ordre de création.**

* donner le nom des tables qu'il n'est pas nécessaire de créer avant la table ci-dessus.

Question 3

Écrire les requêtes SQL suivantes (bien numéroter vos requêtes) :

ATTENTION :

- Faire des **sauts de ligne** au minimum avant les mots clés « SELECT », « FROM », « INNER JOIN », « WHERE », « GROUP BY », « ORDER BY » (sinon pénalité)
- **Souligner** les mots clés ci-dessus (sinon pénalité)
- Écrire le code complet de chaque requête (sinon pénalité importante)
- Utiliser uniquement les tables nécessaires
- Ne pas utiliser de jointure dite « naturel » (« natural join »)

1. Donner le nombre de vendeurs qui ont vendu une voiture du 1 avril 2025 au 7 avril 2025. (*affichage : un numérique uniquement*). Utiliser 1 seule table (*ventes effectuées par le magasin*).
2. Donner le nombre de vendeurs affectés dans chaque magasin dans la ville de 'belfort' . (*affichage : nom du magasin, nombre de vendeurs, le résultat sera trié selon l'ordre lexicographique (alphabétique) inverse sur le nom de magasin*).
3. Donner le nom des magasins de 'belfort' où il y a plus de 4 ventes en avril 2025 (ventes du magasin à un client). (*affichage : nom du magasin, nombre de ventes, le résultat sera trié selon l'ordre lexicographique (alphabétique) inverse sur le nom du magasin*).
4. Donner le nombre de vendeurs dans chaque magasin pour les magasins où il y a plus de 2 vendeurs.

5. Donner le nom des vendeurs d'un magasin qui n'ont pas réalisé de vente en avril 2025 (affichage : nom du (des) vendeurs(s), le résultat sera trié selon l'ordre lexicographique (alphabétique) inverse).
6. Donner le modèle de voiture dont le prix est plus cher que le modèle « toyota yaris » (dénomination). (affichage : nom du (des) modèle(s) de voiture, le résultat sera trié selon l'ordre lexicographique (alphabétique) inverse). Remarque : recherché le prix de la voiture « toyota yaris » la plus cher (« prix_voiture »).
7. Donner le modèle de voiture (dénomination) vendu par au moins 2 vendeurs différents en 2025. (affichage : nom du (des) modèle(s) de voiture, le résultat sera trié selon l'ordre lexicographique (alphabétique) inverse)
8. Donner le nom des clients qui en 2025 ont effectué plus de 2 achats de véhicules.

----- requêtes plus difficiles

9. Donner la voiture qui a rapporté le plus d'argent (voiture la plus rentable). (différence entre prix de vente et le coût de la voiture maximum, sans utiliser le mot clé LIMIT).
10. Donner le nom du vendeur qui a effectué le plus grand nombre de ventes en avril 2025 (sans utiliser le mot clé LIMIT).

Question 4 : modification du MCD

On désire modifier dans le MCD.

Partie 1 : Proposer une solution qui fait apparaître une seule entité « vente » et qui remplace les associations « vente_magasin » et « achat_client ». Vous devrez différencier les ventes du magasin et les achats du magasin par un attribut prix_vente et un attribut prix_achat, cette entité disposera des autres attributs (informations) des 2 associations.

Partie 2 : Proposer une solution pour transformer l'entité « transfert » en une association. Un véhicule ne peut être transféré qu'une seule fois la même journée.

1. Utiliser le deuxième document réponse (verso). Dessiner un nouveau schéma MCD avec les entités/associations et cardinalités utiles. Dans les entités, vous pouvez faire apparaître uniquement les clés et des petits points pour indiquer qu'il y a d'autres attributs si ça ne nuit pas à la compréhension.

Si vous supprimez des attributs, faites les apparaître et rayez les.
Tenir compte des informations ci-dessous.

2. Donner les modifications dans le Modèle Logique de Données initial (page1) : donner le nom des tables à créer en plus ainsi que les attributs en indiquant bien avec les mêmes règles :

* Les clés primaires sont soulignées et les clés étrangères sont précédées par un caractère "#".

* La convention de nommage (typographique) utilisée est le **snake_case**, il y a un "_" entre les mots dans les noms des attributs (colonnes) d'une table et certains noms de table.

Exercice 2

Considérons qu'il existe les 5 tables ci-dessous dans la base de données :

Produit(idProduit, libelle, prixUnitaire)
Facture(idFacture, dateFacture, #idFournisseur)
LigneFacture(#idFacture,#idProduit, quantite)
Fournisseur(idFournisseur, raisonSociale, adresse)
PanierProduit(#idFournisseur,#idProduit, quantite)

Les colonnes idProduit, idFacture et idFournisseur sont des compteurs (« AUTO_INCREMENT »)

Considérons les nuplets (enregistrements) suivants présents dans la table **PanierProduit** :

IdFournisseur	IdProduit	Quantite
5	21	4
5	24	8
7	32	2

Le script python de l'application exécute des commandes SQL pour valider une facture. Écrire le script SQL pour valider la facture d'un fournisseur à l'aide d'une transaction en respectant les instructions ci-dessous. Le script doit fonctionner avec le numéro du fournisseur placé dans la variable @IdFournisseur et quelle que soit la date (utiliser la date d'aujourd'hui).

```
SET @IdFournisseur = 5;  
-- Votre script SQL ci-dessous
```

Instructions :

- Créer la facture avec la date d'aujourd'hui (utiliser une fonction pour récupérer la date d'aujourd'hui).
- Récupérer l'identifiant de la facture dans une variable de nom **@NewIdFacture**.
 - Consigne : Ne pas utiliser la fonction **LastInsertId()** qui n'est pas portable ; on suppose que **idFacture** est un « compteur » (l'attribut est de type INT AUTO_INCREMENT), il faut donc rechercher la valeur maxi de cet identifiant pour ce fournisseur.
- Finaliser la transaction en créant les lignes de la facture. (écrire juste les instructions pour ajouter les lignes de factures).
- Valider dans la session la transaction.

Annexe : mettre dans une variable @id_four l'identifiant d'un fournisseur :

```
SELECT fournisseur_id INTO @id_four FROM fournisseurs  
WHERE fournisseur_id = 'nom du fournisseur';
```