

Introduction aux Bases de Données

L2 Informatique – L2DD

Enseignants : Constance Creux, Guillaume Postic
guillaume.postic@univ-evry.fr

Les fichiers `store.sql` et `hr.sql` contiennent le code nécessaire à la construction de deux bases représentant les données d'un magasin (*store*) et d'un service de ressources humaines (*human resources*, *hr*).

Vous travaillerez sur ces bases en utilisant le système de gestion de base de données (SGBD) MySQL, via les plateformes web :

- DB Fiddle (<https://www.db-fiddle.com/>),
- ou SQL Fiddle (<http://sqlfiddle.com/>).

Vous êtes également libres d'utiliser un SGBD installé sur votre machine personnelle.

Vous devrez répondre à une série de 27 questions de deux sortes :

- les questions portant la mention **Requête**, auront pour réponses des requêtes SQL. Celles-ci devront être rendues dans un fichier `.sql` au format texte (brut).
- les réponses aux autres questions devront être rendues dans un second fichier, au format PDF.

Ces fichiers devront être envoyés avant le vendredi 16/12, 23h59, à l'adresse suivante :
cr-univ-evry@outlook.fr.

Partie 1 : base du magasin

Question 1

Soit la requête suivante :

```
SELECT
    c.customer_id,
    c.first_name,
    o.order_id
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
ORDER BY c.customer_id
```

Si vous remplacez "JOIN" par "LEFT JOIN", quelle différence observez-vous au niveau du résultat ? Comment l'expliquez-vous ?

Question 2

Même question si l'on remplace "JOIN" par "RIGHT JOIN".

Question 3 Requête

Choisir la clause de jointure correcte pour afficher toutes les informations sur les tables 'orders' et 'shippers' où il y a une correspondance.

Question 4

Inverser l'ordre des tables dans la requête : y a-t-il un changement ?

Question 5 Requête

Afficher à l'aide d'une clause de jointure tous les enregistrements de la table 'customers' ainsi que les informations correspondantes, quand elles existent, dans la table 'orders'.

Question 6

Reprendre la même requête en inversant l'ordre des tables et commenter.

Question 7 Requête

Dans la table "order_items" plusieurs enregistrements ont la même valeur de "order_id". Cet attribut ne peut donc pas être utilisé comme identifiant unique de chaque enregistrement. Il en est de même pour la colonne "product_id". Pour identifier chaque item, il faut utiliser une clef primaire composite combinant les valeurs de ces deux colonnes.

La table "order_item_notes" contient une colonne "note_id" et deux autres colonnes correspondant aux attributs précédemment mentionnés.

Écrire une requête affichant toutes les colonnes de la jointure entre les tables "order_items" et "order_item_notes"

Question 8 Requête

La table "orders" contient une colonne "shipper_id" qui identifie un fournisseur pour chaque commande. Cet identifiant correspond à un nom de fournisseur accessible via la table "shippers".

Note : pour certaines commandes, le nom du fournisseur n'est pas documenté (valeur NULL).

Modifier la requête de la Question 1 afin d'afficher :

- colonne 1, l'ID du client, pour tous les clients
- colonne 2, le prénom du client
- colonne 3, l'ID de la commande (NULL pour les clients n'ayant rien commandé)
- colonne 4, le nom du fournisseur (si documenté et si commande effectuée)

Question 9 Requête

Soit la requête suivante contenant une jointure croisée dite "explicite" (*i.e.* avec le mot clef "CROSS JOIN") :

```
SELECT
    c.first_name AS customer,
    p.name AS product
FROM customers c
CROSS JOIN products p
ORDER BY c.first_name
```

Écrire une requête sans "CROSS JOIN" mais affichant le même résultat.

Question 10 Requête

Dans la table "customers", chaque client a des points de fidélité.

On distingue trois catégories de clients :

- "Silver" pour points < 2000,
- "Gold" pour 2000 <= points <= 3000,
- "Platinum" pour points > 3000.

Écrire une requête affichant :

- colonne 1, l'ID du client
- colonne 2, son prénom
- colonne 3, ses points
- colonne 4, sa catégorie

Les lignes doivent être ordonnées par ordre alphabétique des prénoms.

Question 11 Requête

Quels sont les identifiants des clients n'ayant jamais passé de commande ?

Question 12 Requête

En vous servant du résultat de la question précédente, supprimer de la base de clients ceux qui n'ont jamais commandé et vérifier.

Question 13 Requête

Formuler une requête pour répondre simultanément aux deux questions précédentes.

Question 14

Créer deux nouveaux expéditeurs à l'aide de la commande suivante :

```
INSERT INTO shippers (name)
VALUES ('New Shipper 1'), ('New Shipper 2');
```

Vérifier qu'ils apparaissent bien dans la table.

```
SELECT *
FROM shippers;
```

Ajouter deux nouveaux statuts de commande à l'aide de la commande suivante :

```
INSERT INTO order_statuses (name)
VALUES ('Cancelled'), ('Lost');
```

Pourquoi l'insertion a-t-elle marché dans la table shippers, mais pas dans la table order_statuses ? Regarder pour cela les mots clés utilisés lors de la création des tables.

Question 15 Requête

La commande d'Ilene Dowson a été livrée. Modifier son statut dans la table 'orders', et vérifier.

Question 16 Requête

Afficher les commandes passées en janvier (format des dates : YYYY-MM-JJ).

Attention à ne pas retourner toutes les commandes qui auraient un "01" quelque part dans la date.

Question 17 Requête

Quels sont les produits les plus souvent commandés ? Les classer dans l'ordre décroissant en affichant l'identifiant du produit, son nom, et le nombre de fois qu'il a été commandé.

Question 18 **Requête**

Quels sont les produits dont la quantité totale commandée est la plus grande ? Les classer dans l'ordre décroissant en affichant l'identifiant du produit, son nom, et la quantité totale qui en a été commandée.

Question 19 **Requête**

Calculer et afficher par ordre décroissant la somme totale rapportée par chacun des produits.

Question 20 **Requête**

Y a-t-il des clients ayant commandé plusieurs fois le même produit ? Afficher leur nom et le produit.

Partie 2 : base des ressources humaines

Question 21

Dans la table "employees" :

- la première colonne est un ID de chaque employé,
- la colonne "reports_to" donne l'ID de son manager.

Comment identifier le PDG de l'entreprise ?

Question 22 **Requête**

Faire une requête affichant :

- colonne 1, l'ID de chaque employé
- colonne 2, le prénom de chaque employé
- colonne 3, le prénom de son manager

Question 23 Requête

Dans la requête précédente, le PDG n'apparaît nulle part dans la liste des employés (*i.e.* aucune ligne avec son "employee_id").

Pourquoi ? Modifier la requête pour résoudre le problème.

Partie 3 : construction des deux bases

Les questions suivantes portent sur le contenu des fichiers `store.sql` et `hr.sql`.

Question 24

En lisant le code SQL, répertorier et définir tous les types des variables présentes dans les tables des bases `store.sql` et `hr.sql`.

Question 25

Donner la signification des instructions suivantes utilisées pour construire les bases :

- ENGINE=InnoDB
- AUTO_INCREMENT=6
- DEFAULT CHARSET=utf8mb4
- ON UPDATE CASCADE

Question 26 Requête

Après qu'une base a été créée, quelle commande permet de lister toutes les tables présentes dans cette base ?

Question 27

Dessiner le schéma conceptuel de la base `store.sql`.