

Nom : _____

Classe : _____

Bases de données relationnelles — SQL

Au contrôle, une réponse juste sans calcul ne rapporte presque rien : chaque problème se rédige en quatre temps — **justifier**, **poser** l'égalité, **mener le calcul** jusqu'au bout, **conclure** par une phrase avec l'unité. La place réservée est dimensionnée pour ces quatre temps.

1 Vocabulaire. Répondre en une phrase précise.

- Qu'est-ce qu'une clé primaire ? Une clé étrangère ?
- Qu'est-ce que l'intégrité référentielle ?
- Quel est l'intérêt d'éviter la redondance ?
- Différencier WHERE et HAVING.
- Qu'est-ce qu'une injection SQL ?

=====

2 Une table Clients compte 1 000 lignes, une table Commandes 5 000 lignes. Une requête les joint sans condition de rapprochement.

- Combien de lignes la requête renvoie-t-elle ?
- Combien en renverrait-elle avec la bonne condition, au maximum ?
- Calculer le rapport entre les deux.
- Pourquoi le moteur n'émet-il aucune erreur ?
- Pourquoi l'anomalie peut-elle passer les tests et apparaître en production ?

[illegible]

3 Construire des requêtes. Décrire en français ce que chacune doit faire, puis sa structure.



- a. Afficher les clients dont la ville est Lyon.
- b. Afficher chaque client et le nombre de ses commandes.
- c. Afficher les seuls clients ayant passé plus de trois commandes.
- d. Pourquoi la condition de c ne peut-elle pas figurer dans un WHERE ?

4 Injection SQL. Analyser.



- a. D'où vient la faille : du langage SQL ou de la façon d'écrire la requête ?
- b. Que permet d'obtenir une injection réussie ?
- c. En quoi consiste une requête préparée ?
- d. Pourquoi filtrer les caractères dangereux est-il une parade partielle ?
- e. Quel principe général cette faille illustre-t-elle ?

Corrigé détaillé sur progressschool.fr — chapitre « Bases de données relationnelles — SQL ».