



Durée : 2h. Tous documents autorisés.

Soyez précis dans vos réponses, toute réponse non justifiée sera considérée comme fausse.

Partie 1 : SQL

Le site MaHotte gère son système de gestion de commandes avec une base de données relationnelle dont le schéma est donné ci-dessous. L'attribut LIVRAISON.STATUT peut prendre les valeurs 'prête à partir', 'en cours' ou 'livrée'.

CLIENTS (NUMCLI, NOM, ADRESSE, TELEPHONE)
COMMANDES (NUMCO, NUMCLI, DATE COM)
LIGNES_COM (NUMCO, NUMPROD, FOURNISSEUR, QUANTITÉ)
PRODUITS (NUMPROD, NOMPROD, TYPE, COULEUR, PRIX)
LIVRAISON (NUMCO, DATE, STATUT) //

- Question 1 (1 point): faire la déclaration SQL de la table LIGNES_COM avec l'ensemble de ses contraintes d'intégrité (clés primaires et étrangères).
- Question 2 (1 points): Donner le droit à tous les lutins du Père Noël de visualiser la table LIGNES_COM et de mettre à jour l'attribut STATUT de la table LIVRAISON. Pour cela, on créera un rôle 'Lutin' et on donnera à 'Jacquot' et 'Ludo' le droit de jouer ce rôle.
- Question 3 (1 point): Ecrire la requête SQL donnant le nom de tous les clients dont au moins une commande n'a pas été livrée.
- Question 4 (2 points): Ecrire la requête SQL de mise à jour faisant passer le statut de toutes les commandes du client 'PèreNoël' à l'état 'livrée', et ce à la date du jour.
- Question 5 (2 points): Ecrire la requête SQL qui retrouve le nom de tous les clients qui ont commandé un ensemble de produits pour un montant total supérieur à 10.000€.
- Question 6 (2 points): Ecrire la requête SQL qui retrouve le nom de tous les clients qui ont commandé tous les produits existants dont le prix unitaire est supérieur à 1000€.

Partie 2 : index bitmap

Question 2.1 (1 point): Supposons une table T (A, B, C). On construit un index bitmap sur T.B, attribut variant sur un domaine de 64 valeurs. Supposons que les pages disques aient une taille de 4Ko et que T contienne 100.000 tuples. Quel est la taille de l'index bitmap ?

Question 2.2 (2 points): Combien d'entrées/sorties disque (1 E/S = 1 page lue) sont générées pour répondre à une requête du type (T.B=valeur)? Combien y a-t-il de tuples résultats ?

Question 2.3 (2 points): Supposons un deuxième index bitmap sur T.C variant sur un domaine de 256 valeurs. Combien d'entrées/sorties et de tuples résultats sont générés par une requête du type (T.B=valeur1 and T.C=valeur2) ?

Question 2.4 (1 point): Pourquoi n'a-t-on pas construit un index secondaire de type B-Tree sur T.B et T.C ? Par ailleurs, bitmap et B-Tree permettent-ils d'optimiser les mêmes types de recherche ?

Partie 3 : Quiz

Question 3.1 (5 points): Les affirmations suivantes sont-elles vraies ou fausses ? Toute réponse non justifiée sera considérée comme fausse.

1. Dans un B-Tree d'ordre m , chaque nœud (hormis la racine) contient entre m et $2m$ clés. (a) cette règle a pour objectif d'assurer un taux de remplissage d'environ 75% de chaque nœud, (b) cette règle est indispensable pour que l'arbre soit équilibré.
2. Une transaction s'exécutant avec un degré d'isolation 'Repeatable Read' pose plus de verrous que si elle s'exécutait en mode 'Read Committed'.
3. La table T contient 50.000 tuples répartis dans 1000 pages. Si la sélectivité d'un prédicat est inférieure à 5%, j'ai tout intérêt à utiliser un index secondaire.
4. Déclarer une contrainte d'intégrité en mode 'DEFERRABLE' permet de rendre le contrôle de cette contrainte optionnel.
5. Dans certains cas extrêmes (ex: quand la mémoire allouée est très petite), l'algorithme de jointure Block-Nested-Loop marche moins bien que Nested-Loop.