

ENSTA IN206 - Examen de Bases de Données - 2021-2022

Durée : 1h. Aucun document autorisé.

Toutes les questions sont notées sur le même nombre de points, mais ne sont de la même complexité.
Soyez précis dans vos réponses, toute réponse non justifiée sera considérée comme fausse.

Une base de données nommée DOMOTIQUE est utilisée par un organisme pour gérer les événements (EVT) issus de capteurs connectés (CAP) installés dans des appartements (APP). Chaque capteur est installé à une certaine date (date_c) dans une pièce (piece_c) de l'appartement et est associé à un appareil (appareil_c, si une lampe est instrumentée, alors appareil_c = 'lampe'). Chaque événement est produit à une certaine date (date_e), il est associé à un certain type (TYP) qui décrit (desc_t) l'événement considéré, comme par exemple une mise sur off de l'appareil (desc_t = 'off'), une mise à on (desc_t = 'on') ou une mise en veille (desc_t = 'standby'). Les tables et leurs attributs sont donnés ci-dessous.

```
TYP (id_t, desc_t)
EVT (id_e, id_t, id_c, date_e)
CAP (id_c, id_a, date_c, piece_c, appareil_c)
APP (id_a, etage_a, adresse_a, ville_a)
```

Exercice 1 : Conception – modèle Entité/Association

Q1 : En utilisant le formalisme Entités/Associations, proposer un modèle conceptuel de données (MCD), avec toutes les entités, associations, propriétés et cardinalités, ayant pu aboutir à ces tables.

Q2 : Événement devrait-il plutôt être représenté par une entité ou une association ? Justifier en 3 lignes maximum.

Q3 : Des capteurs de température sont installés dans certaines pièces de certains appartements. Ils transmettent à fréquence régulière la température (valeur numérique). Compléter le modèle conceptuel pour supporter ce nouveau type de capteurs.

Exercice 2 : Modèle relationnel et algèbre

Q4 : Dessiner le modèle logique de données (MLD) relationnel correspondant au MCD proposé à la première question Q1, pour vous aider dans la partie SQL suivante.

Q5 : Indiquer les modifications imposées au MLD relationnel pour prendre en compte les modifications introduites en question Q3.

Vous considérerez les opérateurs suivants de l'algèbre relationnelle :

Union :	$R \cup S$	Produit cartésien :	$R \times S$
Différence :	$R - S$	Intersection :	$R \cap S$
Projection :	Π Liste d'attributs (R)	Jointure :	$R \bowtie S$
Restriction :	$\sigma_{\text{Critère}}(R)$	Division :	$R \div S$

Exprimer en algèbre relationnelle, à base des opérateurs ci-dessus, les ensembles définis en Q6, Q7, Q8, Q9.

Q6 : Les événements (id_e, date_e, desc_t) ayant lieu dans la cuisine (piece_c) d'un appartement parisien (ville_a).

Q7 : Les appartements (id_a, adresse_a) avec à la fois des lampes et des radiateurs (appareil_c) instrumentés.

Q8 : La liste des appareils (id_c, appareil_c) qui n'ont jamais été mis en veille.

Q9 : La liste des appartements (id_c) dont toutes les pièces disposent d'au moins une lampe instrumentée.

Exercice 3 : SQL

Ecrire les requêtes SQL répondant aux questions posées dans les questions Q10 à Q14.

Q10 : Quelle est la liste des capteurs (id_c) par appartement (id_a) ?

Q11 : Quelle est la liste des appareils (id_c, appareil_c) qui ne sont jamais mis en veille ?

Q12 : Quelle est la liste des appartements dans lesquels des radiateurs et des lampes sont instrumentées ? (votre réponse comprendra si possible une seule clause SELECT)

Q13 : Quelle est la liste des capteurs (id_c, date_e, date_c) ayant produit un événement à une date (date_e) antérieure à la date d'installation (date_c) de ce capteur ?

Q14 : Quelle est la liste des appartements dans lesquels plus de 10 capteurs sont installés ?

Exercice 4 : Programmation base de données

Q15 : Indiquer ce que fait le programme ci-dessous. Proposer une requête SQL produisant le même résultat qu'affiché par ce programme.

```

CREATE TABLE RESULTAT (capteur NUMBER, event_type varchar(100), nombre NUMBER);
DECLARE
  i NUMBER;
BEGIN
  FOR capteur IN (SELECT * from CAP ORDER BY piece_c) LOOP
    FOR typ IN (SELECT * from TYP) LOOP
      i := 0;
      FOR event IN (SELECT * from EVT) LOOP
        IF capteur.id_c = event.id_c AND event.id_t = typ.id_t THEN
          IF capteur.date_c <= event.date_e THEN
            i := i+1;
          END IF;
        END IF;
      END LOOP;
      IF i != 0 THEN
        INSERT INTO RESULTAT VALUES (capteur.id_c, typ.desc_t, i);
      END IF;
    END LOOP;
  END LOOP;
END;
/
SELECT * FROM RESULTAT;

```

Q16 : Que pensez-vous du programme de la question Q15 ? Justifier votre en 3 lignes maximum.

Exercice 5 : Propriétés transactionnelles

L'exécution présentée dans le tableau ci-dessous indique l'évolution du solde du client 'Joe' pour 2 transactions concurrentes. La séquence des commandes SQL successives traitées par le SGBD et leurs résultats sont indiqués.

	Transaction 1		Transaction 2	
	opération	résultat	opération	résultat
Etat initial	Solde = 1000			
Opération 1	Commit ;			
Opération 2			Commit ;	
Opération 3	Afficher Solde	1000		
Opération 4			Update Solde	1 row updated
Opération 5			Afficher Solde	1100
Opération 6	Afficher Solde	1000		
Opération 7			Commit ;	
Opération 8	Afficher Solde	1100		
Opération 9	Commit ;			

Avec: Update Solde = UPDATE COMPTES SET solde=solde + 100 WHERE nom_client='Joe';
Afficher Solde = SELECT solde FROM COMPTES WHERE nom_client='Joe';

Q17 : Quel est le niveau d'isolation de la transaction 1 ? Justifier en 3 lignes.

Q18 : Pouvez-vous déduire si le système est multi-versions ? Si oui, comment le déduisez-vous, si non, pourquoi ne pouvez-vous pas le déduire ? (3 lignes max)

Q19 : Un organisme de transport propose la vente de trajets (en train ou en bus) entre différentes villes de France à ses clients. L'organisme stocke ses informations dans une base de données transactionnelle. L'objectif de l'organisme est que les nombreux clients puissent, via une application web, consulter l'ensemble des places disponibles et leur prix pour faire leur choix et acheter des places. Un maximum d'achats doivent pouvoir être réalisés de manière concurrente, par de nombreux individus connectés en même temps à l'application. La proposition qui est faite est de définir une transaction par utilisateur, qui démarre à la connexion de l'utilisateur, lui permet de consulter les différentes places disponibles, et se termine (par une validation ou un abandon) juste après l'achat de ses billets. Que pensez-vous de cette solution ? Quelle solution alternative proposeriez-vous s'il fallait augmenter le nombre d'achats concurrents de billets ?