TD20 - base de données relationnelle

Exercice 1 ★ ☆ ☆☆

Une compagnie aérienne dispose d'une base de données formée de deux tables :

• la table *pilotes* composée des attributs suivants :
– numpil (de type INTEGER) : numéro du pilote,
– nompil (de type TEXT) : nom du pilote,
– aeroport (de type TEXT) : nom de l'aéroport où travaille le pilote,
– salaire (de type INTEGER) : salaire du pilote.

• la table *avions* composée des attributs suivants :
– numav (de type INTEGER) : numéro de l'avion,
– marque (de type TEXT) : marque de l'avion,
– capacite (de type INTEGER) : nombre de passagers maximal de l'avion,
– aeroport (de type TEXT) : nom de l'aéroport où est basé l'avion.

Pour chacun des cas suivants, écrire une requête SQL permettant d'effectuer la tâche ci-dessous :

1)Extraire la liste des numéros d'avions dont la capacité est supérieure à 300,

2)Extraire les numéros et la marque des avions basés à Londres,

3)Extraire toutes les informations sur les pilotes de la compagnie,

4)Extraire le nom des pilotes qui travaillent à l'aéroport de Londres et dont le salaire est inférieur à 100000.

5)Mettre à jour la table *pilotes* en multipliant par 2 le salaire des pilotes parisiens.

Exercice 2 ★ ★ ☆☆

Dans la base de données d'une université, on dispose de trois tables gérant les notes des étudiants en L1 à l'examen de passage en L2 :

• la table *etudiants* composée des attributs suivants :
– id_etudiant (de type INTEGER) : numéro d'identification de l'étudiant,
– nom (de type TEXT) : nom de l'étudiant,
– redoublant (de type TEXT) : oui ou non.

• la table *controles* composée des attributs suivants :
– id_controle (de type TEXT) : numéro d'identification du controle,
– matiere (de type TEXT),
– coefficient (de type INTEGER).

• la table *evaluations* composée des attributs suivants :
– id_etudiant,
– id_controle,
– note (de type INTEGER).

Les étudiants font deux épreuves d'anglais (A1 et A2), deux épreuves de philosophie (P1 et P2), deux épreuves d'économie (E1 et E2) et deux épreuves de maths (M1 et M2).

1)Donner un exemple d'enregistrement de la table *etudiants*.

2)Préciser les clés primaires et les clés étrangères de chaque table.

3)Pour chacun des cas suivants, écrire une requête SQL permettant d'effectuer la tâche ci-dessous :

a)créer les tables etudiants, controles et evaluations,

b)extraire le nombre total d'étudiants,

c)extraire parmi l'ensemble de toutes les notes récoltées à l'examen celle qui est la plus haute et celle qui est la plus basse,

d)extraire toutes les notes obtenues à l'examen, ordonnées dans l'ordre croissant,

e)obtenir le nom de tous les étudiants redoublants.

4)Expliquer les requêtes ci-dessous :

a)UPDATE evaluations SET note=10 WHERE note<10

b)SELECT nom FROM etudiants ORDER BY nom

c)UPDATE evaluations SET note=note+1 FROM controles
WHERE matiere="Maths"

d)SELECT note FROM controles INNER JOIN evaluations
WHERE controles.id_controle=evaluations.id_controle
AND controles.matiere="économie"

e)SELECT note FROM etudiants INNER JOIN evaluations
WHERE nom="Durand" AND etudiants.id_etudiant=evaluations.id_etudiant

Exercice 3 ★ ★ ☆ ☆

Une boutique en ligne possède une base de données comportant les tables *articles*, *fournisseurs* et *achats* décrites ci-dessous.
articles(noart,libelle,stock,prix)
fournisseurs(nofour,nomfour,adrfour)
achats(nofour,noart,prix,delai)

Pour chacun des cas suivants, écrire une requête SQL permettant d'effectuer la tâche ci-dessous :

1)Extraire les numéros et libellés des articles dont le stock est inférieur à 10.

2)Extraire la liste des articles dont le prix est compris entre 100 et 300.

3)Extraire les noms et adresses des fournisseurs qui proposent des articles pour lesquels le délai d'approvisionnement est supérieur à 20 jours.

4)Obtenir le nombre d'articles référencés.

5)Obtenir la valeur financière du stock.

Exercice 4 ★ ★ ☆ ☆

On dispose d'une base de données comportant deux tables *vehicule* et *annonce* décrites ci-dessous.

• La table *vehicule* recense des informations sur les modèles de véhicules en vente sur le marché. Elle est composée des attributs suivants :
– id_vehicule (de type INTEGER) : un code permettant d'identifier de façon unique chaque référence de véhicule (marque et modèle),
– marque (de type TEXT) : le nom du constructeur du véhicule,
– modele (de type TEXT) : le modèle du véhicule, le constructeur proposant en général plusieurs modèles de véhicules à la vente,
– prix_neuf (de type INTEGER) : prix de vente du véhicule neuf.

• La table *annonce* regroupe des informations sur un grand nombre d'annonces de véhicules d'occasion. Chaque enregistrement correspond à une annonce et possède les attributs suivants :

– id_annonce (de type INTEGER) : un code permettant d'identifier chaque annonce de façon unique,
– id_vehicule (de type INTEGER) : l'identifiant du modèle de véhicule vendu, qui correspond à l'identifiant utilisé dans la table vehicules,
– annee (de type INTEGER) : année de première mise en circulation du véhicule,
– km (de type INTEGER) : nombre de kilomètres parcourus par le véhicule au moment de la revente,
– prix_occasion (de type INTEGER) : prix de vente du véhicule d'occasion.

1)En justifiant brièvement, identifier une clef primaire dans chacune des tables *vehicule* et *annonce*, ainsi qu'une clef étrangère dans la table annonce.

2)Ecrire une requête SQL permettant d'extraire les noms de tous les modèles de véhicules mis en vente par le constructeur Dubreuil Motors.

3)Expliquer le fonctionnement de la requête SQL suivante et préciser l'effet éventuel de cette requête sur chacune des tables *vehicule* et *annonce*.

```
UPDATE annonce
SET prix_occasion = prix_neuf FROM vehicule
WHERE vehicule . id_vehicule = annonce . id_vehicule
AND vehicule . prix_neuf < annonce . prix_occasion
```

4)A l'aide d'une jointure, écrire une requête SQL permettant d'obtenir, sur une même table, la liste de toutes les annonces de la table *annonce* avec les attributs suivants :
– l'identifiant de l'annonce id_annonce,
– le kilométrage km,
– le prix de vente du véhicule neuf prix_neuf,
– le prix de l'annonce d'occasion prix_occasion.

Réponses

Exercice 1

1)SELECT numav FROM avions WHERE CAPACITE>350

2)SELECT numav,marque FROM avions WHERE aeroport="Londres"

3)SELECT * FROM pilotes

4)SELECT nompil FROM pilotes WHERE ville="Londres" AND salaire<100000

5)UPDATE pilotes SET salaire=2*salaire WHERE aeroport="Paris"

Exercice 2

1)(1,"Durand","oui") est un exemple d'enregistrement de la table *etudiants*.

2)Recherche des clés :

– l'attribut id_etudiant est une clé primaire de la table *etudiants*, (en effet, le numéro d'identification de l'étudiant détermine de façon unique la valeur des deux autres attributs de cette table, à savoir nom et redoublant),

– l'attribut id_controle est une clé primaire de la table *controles*,

– le couple d'attributs (id_etudiant,id_controle) est une clé primaire de la table *evaluations*. En effet, la connaissance de ce couple permet d'obtenir de façon unique la note de l'étudiant au controle,

– l'attributs id_etudiant et id_controle sont des clés étrangères de la table *evaluation* car ce sont respectivement la clé primaire de la table *etudiants* et la clé primaire de la table *controles*.

3)a)CREATE TABLE etudiants(id_etudiant INTEGER,nom TEXT,redoublant TEXT)

CREATE TABLE controles(id_controle INTEGER,matiere TEXT,coefficient INTEGER)

CREATE TABLE evaluations(id_etudiant INTEGER,id_controle INTEGER,note INTEGER)

b)SELECT COUNT(*) FROM etudiants

c)Pour obtenir la note la plus haute : SELECT MAX(Note) FROM evaluations

Pour obtenir la note la plus basse : SELECT MIN(Note) FROM evaluations

d)SELECT note FROM evaluations ORDER BY note

e)SELECT nom FROM etudiants WHERE redoublant="oui"

4)a)La requête met à jour la table *evaluations* en remplaçant toutes les notes en-dessous de la moyenne par 10.

b)La requête extrait le nom de tous les étudiants par ordre alphabétique.

c)La requête fait une mise à jour de la table évaluations en mettant 1 point plus à toutes les notes de maths.

d)La requête effectue une jointure des tables *controles* et *evaluations* :

– elle extrait de cette jointure tous les couples d'enregistrements formés d'un enregistrement de la table *controles* et d'un enregistrement de la table *evaluations*,

– de ces couples, elle ne garde que ceux pour lesquels le numéro d'identification du controle est le même pour les deux tables et pour lesquels la matière est l'économie.

Finalement, le but de la requête est d'extraire toutes les notes d'économie obtenues à l'examen.

e)Même idée qu'en 4)d). La requête extrait toutes les notes de l'élève Durand.

Exercice 3

1)SELECT noart,libelle FROM articles WHERE stock <10

2)SELECT *FROM articles WHERE prix >100 AND prix < 300

3)SELECT nomfour,adrfour FROM fournisseurs INNERJOIN achats
ON fournisseurs.nofour=acheter.nofour AND delai >20

4)SELECT COUNT(*) FROM articles

5)SELECT SUM(stock*prix) FROM articles

Exercice 4

1)Recherche des clés :

– l'attribut id_vehicule est une clé primaire de la table *vehicule* (en effet, deux enregistrements différents de la table *vehicule* ne peuvent posséder le même identifiant. Dit autrement, la connaissance de l'identifiant du véhicule permet de déterminer de façon unique toutes les autres caractéristiques de ce véhicule),

– l'attribut id_annonce est une clé primaire de la table *annonce* (même raisonnement),

– l'attribut id_vehicule de la table *annonce* est une clef étrangère car c'est la clé primaire d'une autre table (la table *vehicule*).

2)SELECT modele FROM vehicule WHERE marque = "Dubreuil Motors"

3)– la commande UPDATE signifie qu'on fait une mise à jour des enregistrements de la table annonce,

– par la commande WHERE, la requête répertorie les couples d'enregistrements formés d'un enregistrement de la table *vehicule* et d'un enregistrement de la table *annonce* possèdant chacun le même identifiant de véhicule,

– par la commande AND, la requête ne garde que les couples d'enregistrements pour lesquels le prix neuf de la table *vehicule* est inférieur au prix occasion de la table *annonce*,

– par la commande SET, pour chacun des couples d'enregistrements précédents, la requête remplace le prix d'occasion de la table *annonce* par le prix neuf de la table *vehicule*.

Autrement dit, cette requête permet de plafonner le prix de vente d'occasion en s'assurant qu'il ne dépasse pas le prix neuf.

A l'issue de la requête, la table vehicule n'est pas modifiée.

4)

SELECT id_annonce , km , prix_neuf , prix_occasion
FROM $\underbrace{\text{annonce INNER JOIN vehicule}}_{\text{jointure}}$
ON annonce . id_vehicule = vehicule . id_vehicule