Interrogation d'informatique
2ème année - Novembre 2021

Durée totale : *60mn*
Documents autorisés : *Aucun.*

INSA

— Le barème est sur 20 points.
— Le sujet est sur 7 pages - il y a un exercice.

Exercice 1 Requêtes SQL (13 pts)

Afin d'optimiser leurs coûts, des agences de voyages mettent en commun des informations sur les voyages qu'elles proposent et sur leurs clients. **Les voyages sont proposés par toutes les agences**, mais un client s'inscrit à un voyage auprès d'une agence particulière, qui encaisse alors le prix du voyage. La base de données est la suivante :

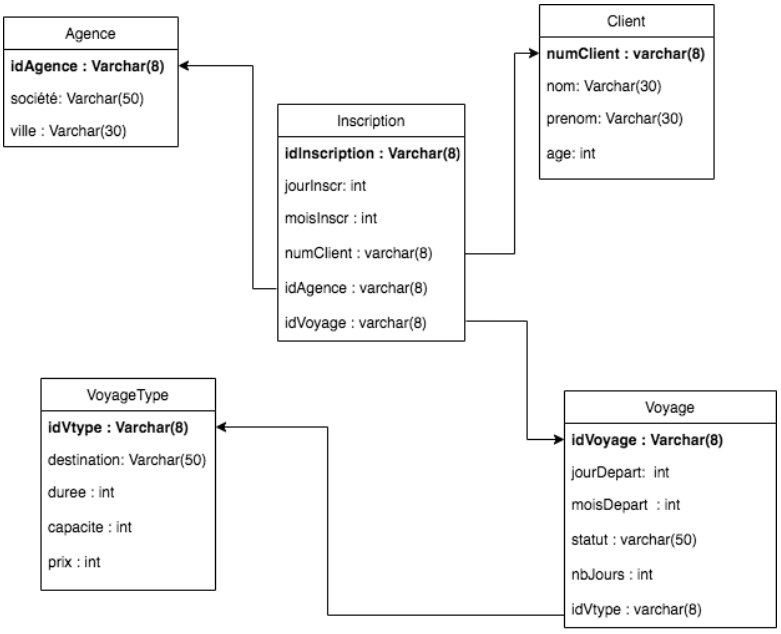

FIGURE 1 – Vue graphique du schéma relationnel de la base.

Dans le schéma relationnel correspondant, les attributs formant la clé sont soulignés.

— `Agence(idAgence varchar(8), societe varchar(50), ville varchar(30)`
Une agence est identifiée de manière unique par un identifiant. On mémorise son identifiant, la société à laquelle elle appartient (par exemple, "Globe Trotters") et la ville où elle se situe. Une société peut posséder plusieurs agences dans une même ville.

— `Client(numClient varchar(8), nom varchar(30), prenom varchar(30), age int(11))`
Chaque client est identifié de manière unique par son identifiant. On mémorise également son nom, son prénom et son âge. On suppose que cette information est à jour.

— `Voyage-Type(idVtype varchar(8), destination varchar(50), duree int(2), capacite int(2), prix int(6))`
Tous les voyages-types proposés par les agences sont représentés dans cette relation. Chaque voyage-type est identifié de manière unique par un identifiant. On mémorise

également sa destination qui correspond à un pays, sa durée en nombre de jours, sa capacité (nombre maximal de personnes par voyage-type) et son prix unitaire en euros. Exemple de voyage-type : un voyage-type d'identifiant VT-BRES01, pour le Brésil qui dure 15 jours, avec une capacité maximum de 20 personnes et dont le prix unitaire est 3000 €.

— `Voyage(idVoyage varchar(8), jourDepart int(2), moisDepart int(2), statut varchar(50), nbjours int(2), idVtype varchar(8))`
Chaque voyage représente un voyage-type programmé à une certaine date (représentée par deux informations `jourDepart` et `moisDepart` : on suppose pour simplifier que la base de données contient uniquement des voyages programmés en 2020). Chaque voyage a un statut dont la valeur peut être '*planifié*', '*annulé*' ou '*interrompu*'. En cas d'interruption, l'attribut `nbJours` est renseigné pour préciser au bout de combien de jours le voyage s'est interrompu. Dans tous les autres cas, ce champ est à NULL.
`idVtype` est une clé étrangère référençant la relation `Voyage-Type`.
Exemple de voyage : le voyage d'identifiant VBR0102, correspondant au voyage-type d'identifiant VT-BRES01, programmé le 1er Février 2020 (jour = 1, mois = 2), planifié (nbjours étant donc à NULL).
Le prix journalier d'un voyage peut être obtenu en divisant le prix unitaire par la durée indiqués pour son voyage-type.
Dès qu'un voyage est ouvert aux inscriptions, il est créé dans la base de données sans attendre l'inscription du premier voyageur.

— `Inscription(idInscription varchar(10), jourInscr int(2), moisInscr int(2), numClient varchar(8), idAgence varchar(8), idVoyage varchar(8))`
Cette relation mémorise les inscriptions des clients à un voyage auprès des agences. Une inscription concerne toujours une seule personne. Chaque inscription est identifiée de manière unique par un identifiant et est caractérisée par une date d'inscription représentée par deux informations `jourInscr` et `moisInscr`. Toutes les inscriptions sont supposées être effectuées en 2020. `numClient` est l'identifiant du client acheteur, `IdAgence` est l'identifiant de l'agence où s'est effectué l'inscription et `IdVoyage` indique à quel voyage le client s'est inscrit. Ces 3 attributs, dans cet ordre, sont des clés étrangères référençant respectivement les relations `Client`, `Agence` et `Voyage`.

Questions : Écrivez les requêtes SQL correspondant aux recherches suivantes. Prenez soin d'éliminer les doublons lorsque le résultat ne doit pas en comporter.

1. Quels voyages-types à destination du Chili ont une durée d'au moins 10 jours et un prix ne dépassant pas 3000 € ? Affichez tous les attributs.
*SELECT * FROM Voyage-Type*
WHERE destination = 'Chili'
and duree >= 10
and prix <= 3000 ;
1,5 pt, si toutes les conditions sont données
2. Donnez pour chaque société, le nombre d'agences implantées. Le résultat doit comporter les informations *société* et le nombre correspondant d'agences.
SELECT societe, count(*) as nbagences
FROM Agence
GROUP BY societe
1,5 pt : dont 0,5pt pour le group By et 0,5 pt pour le count

3. Tous les voyages programmés à compter du 15 mars voient leur statut modifié à *"annulé"* dans la base de données. Donnez la liste des clients à contacter, triée par ordre croissant sur les voyages puis par âge décroissant. L'identifiant du voyage ainsi que l'âge des clients doivent faire partie du résultat.
select Inscription.idVoyage, nom, prenom, age
FROM Client, Inscription, Voyage
Where Client.numClient = Inscription.numClient
AND Inscription.idVoyage = Voyage.idVoyage
and statut='Annulé'
and (jourDepart >= 15 and moisDepart = 3 OR moisDepart > 3)
ORDER BY idVoyage, Age DESC
2pt : selection correcte sur les dates (0,5pt) ; tri double critère ordres respectés (0,5pt) ; tables utilisées correctes (0,5pt) ; jointures correctes(0,5pt) Ne pas pénaliser si pas de selection sur statut ='annulé' puisque la selection sur les dates d'apres l'enoncé implique que les voyages sont annulés

4. Nous aimerions connaître par agence le coût total des remboursements pour les voyages qui se sont interrompus quelques jours après le départ. Affichez l'identifiant de l'agence et le coût total correspondant. On suppose que chaque remboursement se fait en multipliant le prix journalier par le nombre de jours non séjournés.
Les voyages annulés ne sont pas concernés.
Exemple : *pour une inscription à un voyage supposé durer 10 jours ayant comme prix 3000 €, mais interrompu au bout de 6 jours, le remboursement au client inscrit sera de 1200€.*
SELECT idAgence, sum((prix/duree)(duree-nbJours)) as "A rembourser"*
FROM Inscription,Voyage, Voyage-Type
Where Inscription.idVoyage= Voyage.idVoyage
and Voyage.idVType= Voyage-Type.idVType
and statut = 'interrompu'
Group By Inscription.idAgence ;
2 pts : formule de calcul d'un remboursement unitaire(0,5pt) ; calcul de la somme des remboursements(0,5pt)+ group by par agence(0,5pt)+ 0,5 pour le choix de table et des jointures
5. Calculez la somme totale en euros représentée par toutes les inscriptions réalisées auprès des agences de la société 'Globe Trotters' à Bordeaux pendant le mois de Février. La réponse à cette requête ne tient pas compte des remboursements à l'issue des voyages interrompus ou annulés.
SELECT sum(prix)
from Agence, Voyage-Type, Inscription, Voyage
Where Agence.idAgence =Inscrtiption.idAgence
and Inscription.idVoyage = Voyage.idVoyage
and Voyage.idVType= Voyage-Type.idVType
and Agence.ville = 'Bordeaux'
and societe = 'Globe Trotters'
and Inscription.moisInscr= 2 and Inscription.jourInscr>= 1 and Inscription.jourInscr<=28
2pt : utilisation des 4 tables et des 3 jointures correctes(0,75pt) ; calcul de la somme(0,5pt), sélections(0,75pt)
sanctionner l'usage du group By qui du coup est inutile(-0,25pt)
6. Quels sont les voyages pour le Brésil ouverts à l'inscription, pour lesquels il n'y a eu

aucun acheteur ? Précisez l'identifiant du voyage, le jour et le mois du voyage ainsi que l'identifiant du voyage-type correspondant.

SELECT jourDepart, moisDepart, Voyage-Type.idVType, idVoyage
FROM Voyage-Type, Voyage
Where Voyage-Type.idVType= Voyage.idVType
and destination = 'Brésil'
and idVoyage NOT IN (
SELECT idVoyage
FROM Inscription)
1,5pt : requête interne (0,5pt) ; la négation (0,5pt) ; choix des tables+ jointure+selection(0,5pt)

7. Donnez pour chaque agence et pour chaque voyage *planifié*, le nombre d'inscriptions effectuées auprès de l'agence. Vous vous limiterez aux voyages (quelle que soit l'agence où s'est effectué l'inscription) dont la moyenne d'âge des participants dépasse 60 ans.
Indication : Vous pouvez par exemple commencer par identifier les voyages concernés.

SELECT idAgence, Inscription.idVoyage, count(*)
FROM Inscription, Voyage
Where Inscription.IdVoyage = Voyage.idVoyage
AND statut = 'planifié'
and Inscription.idVoyage IN
(select idVoyage
from Inscription, Client
where Inscription.numClient = Client.numClient
GROUP BY idVoyage
HAVING AVG(age)> 60)
GROUP BY idAgence,idVoyage
3pts :
1,5pt : requête interne avec group By + having (0,75pt) ; utilisation des bonnes tables + jointure (0,75pt). Pénaliser l'utilisation inutile de tables : pas besoin de la table Voyage (-0,25pt) 1,5pt : requête principale : group by sur les deux critères de jointure(0,5pt) ; utilisation du count (0,5pt), choix des bonnes relations et jointure correcte(0,5pt). Pénaliser l'utilisation inutile de la table Agence
Sur des tentatives qui ne répondent pas exactement à la question, comme par exemple :
SELECT idAgence, Inscription.idVoyage, count(*)
FROM Inscription, Client, Voyage
Where Inscription.numClient = Client.numClient
AND Inscription.idVoyage = Voyage.idVoyage
AND statut = 'planifié'
GROUP BY idAgence, Inscription.idVoyage
HAVING AVG(age)> 60
compter 1,5pt : 0,75pt : group by sur les deux critères de jointure ; 0,75 pour le reste
Cette tentative ne répond pas à la question puisque la double jointure puis groupement par voyage et agence peut conduire à des groupes où les voyageurs sont jeunes (moyenne < 60 ans) et du coup ce couple (voyage, Agence ne fera pas partie du resultat. alors que globalement la moyenne d'age pour ce voyage est supérieure à 60 ans.

Exercice 2 Modèle conceptuel et modèle relationnel (7 pts)

Le schéma entité-association correspondant à la base de données est illustré sur la figure 2 suivante :

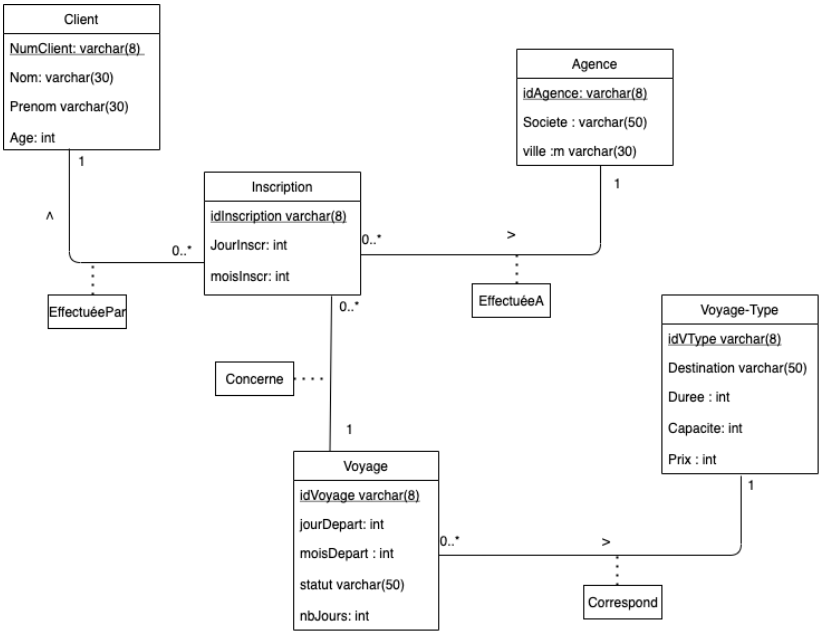

FIGURE 2 – Schema E/A correspondant au schema relationnel fourni

(Q2.1) Extension du modèle entité-association - partie 1

On souhaite étendre la base de données pour mémoriser les détails des hébergements prévus par les voyages-types. Pour cela, la base de données doit permettre de mémoriser pour chaque hôtel, son nom, sa catégorie (exemple : 3*, 4*,...etc) et la ville où il se situe. Chaque hôtel possède un identifiant unique. Chaque voyage-type fait appel à 1 ou plusieurs hôtels (c'est le cas des circuits par exemple) dans un ordre et avec un nombre de jours de séjour qui lui sont propres. Par exemple, un voyage en Malaisie peut prévoir d'abord (ordre = 1) un séjour de 5 jours à l'hôtel *Petaling Jaya* à *Kuala Lumpur* et (ordre = 2) un séjour de 6 jours à l'hôtel *Chulia Mansion* à *Penang*. Un hôtel peut être "utilisé" par plusieurs voyages-types, dans un ordre et avec des durées de séjour différents. Un hôtel peut bien entendu (c'est le cas des circuits) être utilisé deux fois dans le cadre d'un même voyage-type (au début et à la fin par exemple).

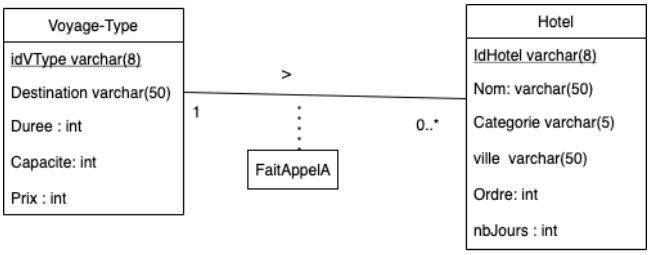

FIGURE 3 – Extension proposée du modèle entité/association pour la gestion des hébergements

Question : Selon vous, le schéma de la figure 3 est-il cohérent avec la description précédente ? Si ce n'est pas le cas, expliquez pourquoi et proposez les modifications de schéma nécessaires.

sur 2 points
0,5pt : cardinalité coté hotel est 1.. au lieu de 0..* : Chaque voyage-type fait appel à 1 ou plusieurs hôtels*
0,5pt : cardinalité côté voyage est de 1.. au lieu de 1 : Un hôtel peut, sur une même destination être "utilisé" par plusieurs voyages-types*
1pt : ordre et nbJours caractérisent le lien entre un voyage et un hôtel donc font partie des propriété de l'association FaitAppel.
(bonus : 0,5 pt pour toute reflexion pertinente sur le fait que l'utilisation multiple du même hotel au niveau du même voyage-type n'est pas prise en compte au niveau du schema conceptuel mais se traduira par l'ajout de Ordre dans la clé de la future table FaitAppel quand on passera au relationnel, pour autoriser cette possibilité

(Q2.2) Extension du modèle entité-association - partie 2

Certaines agences proposent au client en plus du voyage, certaines prestations supplémentaires à utiliser lors du voyage. Chaque prestation possède un identifiant, une description et un prix unitaire. Les prestations proposées varient d'une agence à l'autre. Une même prestation peut être proposée par plusieurs agences. A tout moment, on doit pouvoir connaître les prestations proposées par une agence même si aucun client ne les a achetées. On doit également pouvoir connaître les prestations "rattachées" à une inscription à un voyage. Pour chaque prestation choisie, on doit garder, dans la base de données, le nombre d'unités acquises de cette prestation (par exemple 2 tickets pour un spectacle) ainsi que la date de la reservation.

(Q2.2).1 Proposez une modification du schéma entité-association de la figure 2, pour tenir compte de cette extension. Vous reproduirez uniquement les entités concernées par ces changements.

3 pts
0,5pt : Création d'une entité prestation
0,5 : Association 'propose'. 0,5pt Cardinalités. accepter une cardinalité de 1.. ou 0..* coté agence si l'argumentation est correcte. la cardinalité coté prestation est de 0..**
0,5pt : Association 'ajoueeA'. 0,5pts propriétés de l'association et 0,25 pts par cardinalité correcte

(Q2.2).2 Transformez le schéma entité-association obtenu à la question (Q2.2).1, en schéma relationnel.

2pt
Règle 1 : 0,5 création de la relation Prestation
Prestation (idPrestation : int , type varchar (50), prix : int)
Règle 2 : pas d'association éligible
Règle 3 :
(0,75pt)Proposition(idAgence : varchar(8), idPrestation : int), idAgence clé étrangère qui référence agence, IdPrestation : clé étrangère qui référence Prestation
(0,75pt)AjoutDe(idInscription : varchar(10), idPrestation : int, dateResa : date, NbreUnites : int), IdPrestation : clé étrangère qui référence Prestation, idInscription clé étrangère qui référence Inscription.

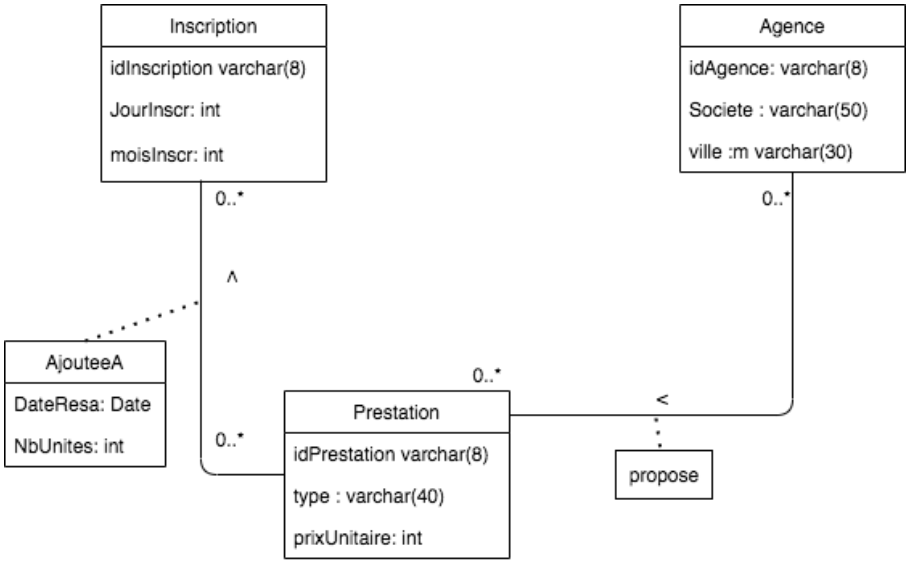

FIGURE 4 – Modèle entité/association, syntaxe UML, partie sur les prestations

Remarque : Dans la cas où le schema conceptuel est faux, noter sur 2 le fait que les règles de transformation soient appliquées correctement et de manière cohérente avec le schema conceptuel obtenu