

**Examen final**

**Durée : 02H00**

**Documents interdits**

**Question de cours (4 points) :**

A. Répondez aux questions suivantes :

1. Quelles sont les propriétés que doit assurer une clé d'une relation dans le modèle relationnel ? (0,5pt)
2. Citez les axiomes d'Armstrong (1pt)
3. Donner le nom complet de SQL (0,5pt)

B. Répondez par Vrai ou Faux aux questions suivantes en justifiant vos réponses :

1. Une relation en forme normale Boyce et Codd n'est pas forcément en 3<sup>ème</sup> forme normale. (1pt)
2. Une dépendance fonctionnelle est dite directe si et seulement si le membre droit est un sous-ensemble du membre gauche (0,5 pt)
3. L'opération Project en algèbre relationnelle permet de filtrer quelques enregistrements d'une relation selon une condition. (0,5pt)

<p>A.</p> <p>1. une clé est un groupe d'attributs minimum ET détermine de façon unique tous les enregistrements de la relation <b>0,5 pt</b></p> <p>2. Réflexivité (<math>Y \subseteq X \Rightarrow X \rightarrow Y</math>) Augmentation (<math>X \rightarrow Y \Rightarrow XZ \rightarrow YZ</math>) Transitivité (<math>X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z</math>) <b>1pt</b></p> <p>3. Structured query language <b>0,5pt</b></p>	<p><b>Total: 4pts (réponses + justification)</b></p>
<p>B.</p> <p>1. Faux une relation en BCNF est en 2FN, la BCNF élimine les DF dont la partie gauche n'est pas une clé primaire, elle élimine donc les DF transitives (de la 3FN) <b>1pt</b></p> <p>2. Faux. Une DF est dite directe, s'il n'existe pas un attribut C qui engendrerait une dépendance fonctionnelle transitive <math>A \rightarrow C \rightarrow B</math> <b>0,5pt</b></p> <p>3. Faux. L'opérateur Project permet de récupérer uniquement les colonnes mentionnées dans le project. <b>0,5pt</b></p>	

**Exercice 1 (6 points) :**

Soit la relation suivante (nous supposons que tous les attributs mentionnés sont atomiques) :

$R(A,B,C,D,E,F,G,H,I)$

Soit l'ensemble F des DF définies sur R :

$A,B \rightarrow C; C,E \rightarrow D; D \rightarrow I; G \rightarrow E, G \rightarrow F; F \rightarrow H$

**Questions :** Il est demandé de justifier vos réponses pour toutes les questions ci-dessous :

1. Donner la couverture minimale de F. **(2pt)**
2. En quelle forme normale est la relation R ? **(3pt)**
3. On souhaite appliquer l'algorithme de décomposition sur la relation R pour assurer la forme normale de Boyce et Codd, quelles sont les dépendances fonctionnelles de F qu'on pourrait utiliser pour commencer cette décomposition. **(1pt)**

<p>1. Couverture minimale : <math>R(A,B,C,D,E,F,G,H,I)</math>  <math>A,B \rightarrow C; C,E \rightarrow D; D \rightarrow I; G \rightarrow E, G \rightarrow F; F \rightarrow H</math></p> <p><math>AB + (F - \{AB \rightarrow C\}) = \{A,B\}</math> non redondante car C n'appartient pas à cet ensemble  <math>CE + (F - \{CE \rightarrow D\}) = \{C,E\}</math> non redondante...  <math>D + (F - \{D \rightarrow I\}) = \{D\}</math> non redondante...  <math>G + (F - \{G \rightarrow E\}) = \{G,F,H\}</math> non redondante...  <math>G + (F - \{G \rightarrow F\}) = \{G,E\}</math> non redondante...  <math>F + (F - \{F \rightarrow H\}) = \{F\}</math> non redondante...</p>	<p>2 points si toutes les fermetures sont données</p>
<p>2. la FN de la relation R : il faut d'abord chercher la clé  Sources : A, B, G  Puits : H, I, F, D, C, E  Une relation qui n'a que des attributs sources et puits n'a qu'une seule clé, contenant l'ensemble des sources, donc : ABG est la clé <math>R(\underline{A,B,G},C,D,E,F,H,I)</math>  La relation est en 1FN (car existence de DF partielle comme : <math>AB \rightarrow C</math>)</p>	<p>Sources : 0,5  Puits : 1  Clé : 1  Réponse FN: 0,5</p>
<p>3. Toutes les DF ne respectent pas la bcnf, on peut commencer par n'importe laquelle</p>	<p>1 pt</p>

**Exercice 2 (10 points) :**

Une association qui propose diverses sorties (randonnée, camping, balade à vélo, escalade, etc.) dans les villes algériennes, gère ses données via la base de données suivante nommée DZTour :

**sortie** (idsortie int, designations text(20), tarif float, ville text(20), datesortie date)

**activité** (idactivité int, idsortie int, nomactivité text(20))

activité.*idsortie* référence sortie.*idsortie*

**participant** (idparticipant int, nom text(20), prénom text(20), age int)

**participer** (idsortie int, idparticipant int, note int)

*participer.idsortie référence sortie.idsortie*

*participer.idparticipant référence participant. idparticipant*

**equipement** (idequipement int, nomequipement)

**equipsortie** (idequipement int, idsortie int)

*equipsortie.idsortie référence sortie.idsortie*

*equipsortie. idequipement référence equipement.idequipement*

Veuillez noter que :

- Les clés primaires sont soulignées et les clés étrangères sont indiquées à la suite à chaque relation.
- L'attribut **note** est une appréciation donnée par le participant pour la sortie à laquelle il a participé. Trois notes sont possibles par ordre d'appréciation : 1 (le participant n'a pas apprécié la sortie), 2 (le participant a moyennement apprécié), 3 (le participant a beaucoup apprécié)

1. Donner la requête ALGEBRIQUE répondant au besoin suivant **(1 pt)** :

- La liste des identifiants et noms des équipements de la sortie « Escalade Montagne Hoggar »

<pre>r1&lt;-restrict(sortie,designations='escalade montagne hoggar') r2&lt;-join(r1,equipsortie) r3&lt;-sjoin(equipement,r2)</pre>	1 pt
--	------

2. Donner les requêtes SQL (sans utiliser les vues) répondant aux besoins suivants **(6 pt)** :

- Classer les sorties par ordre croissant selon le nombre de participants. Le résultat doit afficher l'identifiant de la sortie, sa désignation et le nombre de participants.
- Afficher les sorties (identifiant et désignation) ainsi que leur popularité. La popularité est définie en fonction du cumul (ou somme) des notes accordées par les participants à ces sorties. Les sorties populaires sont celles qui ont un cumul des notes supérieur à 10, les sorties moyennement populaires sont celles qui ont un cumul des notes entre 5 et 10, les sorties non populaires sont celles qui ont un cumul des notes inférieur à 5.
- Répondre par « 1 » si les sorties portant les identifiants 1 et 2 ont des participants en commun, « 0 » sinon

a	<pre>select idsortie, designations, count(idparticipant) nb from participer natural join sortie group by idsortie order by nb ASC</pre>	2 pts
b	<pre>select idsortie, 'Populaire' Popularite from</pre>	2 pts

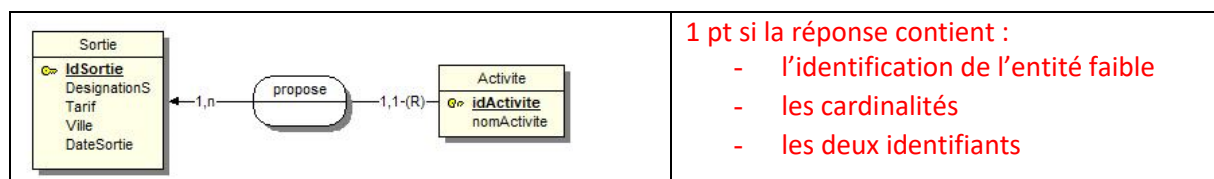
	<pre>(select idsortie, sum(note) nb from participer group by idsortie having nb&gt;10) union select idsortie, 'Moyennement populaire' Popularite from (select idsortie, sum(note) nb from participer group by idsortie having nb&lt;=10 )  **Avec case** select idsortie, (case when nb&gt;10 then 'populaire' else 'moyennement populaire' end) popularite from (select idsortie, sum(note) nb from participer group by idsortie)</pre>	
c	<pre>select exists( select idparticipant from participer where idsortie=1 intersect select idparticipant from participer where idsortie=2) reponse</pre>	2pts

3. Donner le schéma E/A correspondant aux deux relations Sortie et Activité **(1 pt)**

**sortie** (idsortie int, designations text(20), tarif float, ville text(20), datesortie date)

activité (idactivité int, idsortie int, nomactivité text(20))

activité.idsortie référence sortie.idsortie



4. Soit les vues suivantes définies sur la base de données DZTour implémentée sur le SGBD Mysql.

Create View V1 AS

Select \* From Sortie

Where tarif > 2000

with check option;

Create View V2 AS

Select \* From Sortie

Where tarif > 1500

En supposant que les vue V1 et V2 et également la table Sortie soient vides :

- a) Quel sera le contenu de la vue V1 et de la table Sortie après cette insertion (justifier votre réponse) :

*Insert into V1 values (1,'randonnée forêt,3000, 'Alger', '2020-06-30')*

- b) Quel sera le contenu de la vue V2 et de la table Sortie après cette seconde insertion (justifier votre réponse) :

*Insert into V2 values (2,'camping',1500,'Alger','2021-05-21')*

a)	<p>Contenu V1 : <i>(1, 'randonnée forêt,3000, 'Alger', '2020-06-30')</i></p> <p>Contenu Sortie : <i>(1, 'randonnée forêt,3000, 'Alger', '2020-06-30')</i></p> <p>Justification : la clause 'with check option' permet l'insertion dans la vue et la table si la condition du where est vérifiée, ce qui est le cas ici, l'insertion se répercute sur la table Sortie</p>	<p><b>1 pt</b> si les réponses + justification sont correctes,</p>
b)	<p>Si les deux insertions sont faites successivement :</p> <p>Contenu de V2 : <i>V2 contient uniquement la ligne insérée précédemment (1, 'randonnée forêt,3000, 'Alger', '2020-06-30') mais pas la seconde car la condition du where n'est pas respectée</i></p> <p>Contenu Sortie : <i>(1, 'randonnée forêt,3000, 'Alger', '2020-06-30')</i> <i>ET (2, 'camping',1500,'Alger','2021-05-21')</i> La seconde insertion est rejetée dans la vue uniquement, mais pas dans la table</p> <p>Si les deux insertions sont faites indépendamment : Contenu de V2 : <i>vide car la condition du where n'est pas respectée</i></p> <p>Contenu Sortie : <i>(2, 'camping',1500,'Alger','2021-05-21')</i> La seconde insertion est rejetée dans la vue uniquement, mais pas dans la table</p>	<p><b>1 pt</b> si les réponses + justification sont correctes (les deux réponses ont été acceptées: insertions successives ou séparées)</p>