

<b>Question 4</b>	<b>SQL</b>	<b>6.25 pts</b>
-------------------	------------	-----------------

**Partie 1: Considérez la base de données suivante appelée AppStore avec les tables suivantes. Choisissez la bonne réponse. Marquez vos réponses sur la feuille de réponses. (2.5 pts / 0.25 pts par réponse correcte)**

```

Apps (app_id, app_name, rating, category, developer_id)

Users (user_email, user_fullname)

Downloads (app_id, user_email)

Developer (developer_id, developer_name, country, developer_email)

```

**1.1) La clé primaire des Downloads doit être:**

- a) app\_id
- b) user\_email
- c) (app\_id, user\_email)
- d) Les attributs ne sont pas suffisants pour créer une clé primaire. Nous devons ajouter downloads\_id

**1.2) Laquelle des affirmations suivantes est vraie?**

- a) developer\_id dans Developer est une clé étrangère référençant developer\_id dans Apps
- b) developer\_id dans Apps est une clé étrangère référençant developer\_id dans Developer
- c) user\_email dans users est une clé étrangère référençant user\_email dans Downloads
- d) app\_id dans apps est une clé étrangère référençant app\_id dans downloads

**1.3) Quelle requête génère les noms des utilisateurs qui ont téléchargé Viber.**

- a) `SELECT APP_NAME FROM APPS WHERE APP_ID= "VIBER"`
- b) `SELECT USER_FULLNAME FROM USERS NATURAL JOIN APPS WHERE APP_NAME="VIBER"`
- c) `SELECT USER_FULLNAME FROM USERS NATURAL JOIN APPS WHERE APP_ID="VIBER"`
- d) `SELECT USER_FULLNAME FROM USERS NATURAL JOIN APPS NATURAL JOIN DOWNLOADS WHERE APP_NAME="VIBER"`

**1.4) Quelle requête génère l'ID du développeur et le nombre d'applications qu'il a développées.**

- a) `SELECT developer_id, app_id FROM Apps SORT BY developer_id`
- b) `SELECT developer_id, app_id FROM Apps GROUP BY developer_id`
- c) `SELECT developer_id, COUNT(app_id) FROM Apps GROUP BY developer_id`
- d) `SELECT app_id, COUNT(developer_id) FROM Apps GROUP BT app_id`

**1.5) Donnez aux développeurs les noms qui ont développé, au moins 1 application "social media".**

- a) `SELECT developer_name FROM developer WHERE developer_id IN "SOCIAL MEDIA";`
- b) `SELECT developer_name FROM developer CROSS JOIN apps ON  
developer.developer_id = apps.developer_id WHERE category = "SOCIAL MEDIA";`
- c) `SELECT developer_name FROM developer WHERE developer_id EXISTS (SELECT  
developer_id FROM apps WHERE category="SOCIAL MEDIA";`
- d) `SELECT developer_name FROM developer, apps WHERE CATEGORY="SOCIAL MEDIA";`

**1.6) Quels développeurs vivent dans le même pays ?**

- a) `SELECT developer_id AS D1, developer_id AS D2, country FROM developer WHERE  
country C1 = country C2;`
- b) `SELECT D1.developer_id, D2.developer_id, country FROM developer D1,  
developer D2 WHERE D1.country=D2.country AND D1.developer_id > D2.developer_id;`
- c) `SELECT D1.developer_id, D2.developer_id, country FROM developer D1 NATURAL  
JOIN developer D2;`
- d) `SELECT country, developer_id FROM developer GROUP BY COUNTRY;`

**1.7) Quels noms d'applications commencent par le terme « insta »?**

- a) `SELECT app_name FROM Apps LIKE "insta%"`
- b) `SELECT app_name FROM Apps WHERE app_name = "insta";`
- c) `SELECT app_name FROM Apps WHERE app_name LIKE "insta_";`
- d) `SELECT app_name FROM Apps WHERE app_name LIKE "insta%";`

**1.8) Quels utilisateurs sont également des développeurs ?**

- a) `SELECT * FROM users, developer;`
- b) `SELECT * FROM users NATURAL JOIN developer;`
- c) `SELECT * FROM users, developer WHERE user_email = developer_email;`
- d) `SELECT * FROM users, developer WHERE user_fullname = developer_name;`

**1.9) Qui (id & name) a développé l'application « snapchat »?**

- a) `SELECT developer_id, developer_name FROM developer WHERE developer_id IN  
(SELECT developer_id FROM apps WHERE app_name="snapchat");`
- b) `SELECT developer_id, developer_name FROM developer NATURAL JOIN apps where  
app_name = "snapchat";`
- c) `SELECT developer_id, developer_name FROM developer, apps WHERE  
developer.developer_id=apps.developer_id AND app_name="snapchat";`
- d) Toutes les options sont correctes.

**Partie 2 : Supposons que les données suivantes étaient disponibles dans les tables de notre base de données. Choisissez la bonne réponse aux questions suivantes et marquez vos réponses sur la feuille de réponses. (4.5 pts)**

Apps

App_id	App_name	rating	category	Developer_id
1	Facebook	4.1	Social media	100
2	Twitter	4	Social media	104
3	iwork	2.2	productivity	112
4	Snapchat	3.9	Social media	106
5	Instapaper	3.1	news	109
6	Instagram	4.4	Social media	100
7	finder	3	productivity	112

Downloads

App_id	User_email
1	sara@gmail.com
1	wassim@gmail.com
1	rabih@gmail.com
6	sara@gmail.com
7	sara@gmail.com
5	rabih@gmail.com

Users

User_email	User_fullname
sara@gmail.com	Sara Jammal
wassim@gmail.com	Wassim Chamass
rabih@gmail.com	Rabih Haidar

Developer

Developer_id	Developer_name	country	Developer_email
112	The coders	china	codex@gmail.com
109	Rabih Haidar	Lebanon	rabih@gmail.com
100	Fb developers	USA	fbdev@gmail.com
106	Snap developers	USA	snapdev@gmail.com

**2.1) Quel est le résultat de la requête suivante ?**

**SELECT App\_id FROM users NATURAL JOIN downloads WHERE user\_fullname= "Sara Jammal";**

a)

1
6
7

b)

1
1
1

c)

6
6
5

d)

1
---

**2.2) SELECT App\_id FROM users NATURAL JOIN downloads WHERE user\_fullname LIKE "%ma\_";**

a)

1
6
7

b)

1
1
6
7

c)

6
6
5

d)

1
---

2.3) `SELECT user_email FROM downloads where app_id=1 AND app_id=5`

- a) 

sara@gmail.com
----------------
- b) 

sara@gmail.com
rabih@gmail.com
- c) 

sara@gmail.com
rabih@gmail.com
wassim@gmail.com
- d) Aucune des réponses ci-dessus

2.4) `SELECT MIN(rating) FROM Apps WHERE developer_id <110;`

- a) 3.1
- b) 2.2
- c) 3
- d) Aucune de ces réponses

2.5) `SELECT category, MAX(rating) FROM Apps GROUP BY category;`

- a) 

category	Max(rating)
Social media	4.4
productivity	3
news	3.1
- b) 

category	Max(rating)
iwork	2.2
- c) 

category	Max(rating)
Social media	4.4
Social media	4.1
Social media	4
- d) Expression SQL illégale

2.6) `SELECT app_name, MIN(rating) FROM Apps GROUP BY category;`

- a) 

App_name	MIN(rating)
snapchat	3.9
iwork	2.2
instapaper	3.1
- b) 

App_name	MIN(rating)
Social media	4.4
- c) 

App_name	MIN(rating)
iwork	2.2
instapaper	3.1
finder	3
- d) Expression SQL illégale

2.7) **SELECT category FROM apps GROUP BY category HAVING COUNT(\*)>2;**

- |              |              |
|--------------|--------------|
| a)           | b)           |
| social media | social media |
|              | productivity |
| c)           | d)           |
| news         | productivity |

2.8) **SELECT app\_id FROM apps WHERE app\_id NOT IN (SELECT app\_id FROM downloads)**

- |   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
|---|----|----|----|--|---|---|---|---|---|---|---|---|---|---|
| a)  | b) | c) | d) |  |   |   |   |   |   |   |   |   |   |   |
| <table border="1"><tr><td>1</td></tr><tr><td>6</td></tr><tr><td>7</td></tr></table> | 1  | 6  | 7  | <table border="1"><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>6</td></tr><tr><td>7</td></tr></table> | 1 | 1 | 6 | 7 | <table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr></table> | 2 | 3 | 4 | <table border="1"><tr><td>1</td></tr></table> | 1 |
| 1   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 6   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 7   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 1   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 1   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 6   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 7   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 2   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 3   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 4   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |
| 1   |    |    |    |  |   |   |   |   |   |   |   |   |   |   |

2.9) **SELECT developer\_name FROM downloads NATURAL JOIN developer**

- a) Rabih Haidar
- b) The coders
- c) Snap developers
- d) Ensemble vide

2.10) **SELECT app\_name FROM apps WHERE app\_id IN (SELECT app\_id FROM downloads)**

- a) Cette requête tente de trouver toutes les applications qui ont été téléchargées au moins une fois.
- b) Cette requête tente de trouver toutes les applications qui n'ont pas été téléchargées.
- c) Cette requête tente de trouver les applications qui ont été développées.
- d) Aucune de ces réponses

2.11) **Quelle requête pourrait-elle être utilisée pour ajouter la contrainte de clé étrangère pour "developer\_id":**

- a) **ALTER TABLE developer ADD FOREIGN KEY(developer\_id) REFERENCES Apps(developer\_id)**
- b) **ALTER TABLE Apps ADD FOREIGN KEY(developer\_id) REFERENCES developer(developer\_id)**
- c) **INSERT INTO Apps FOREIGN KEY(developer\_id) REFERENCES developer(developer\_id)**
- d) **UPDATE Apps SET developer\_id FOREIGN KEY REFERENCES developer(developer\_id)**

**2.12) Supposons que les tables présentées ci-dessus ont été créées et remplies de données, sans définir les contraintes de clé étrangère. Plus tard, nous ajoutons les contraintes de clé étrangère (similaires à la partie 2.12), lesquelles d'entre elles donneront une erreur une fois on tente de les exécuter ?**

- a) la contrainte de developer\_id
- b) la contrainte de app\_id
- c) la contrainte de user\_email
- d) Tous seront couronnés de succès.

**2.13) Pour ajouter un nouveau tuple a la relation users :**

- a) `INSERT INTO users VALUES("Samir@gmail.com", "Samir Awad");`
- b) `UPDATE users WITH VALUES("Samir@gmail.com", "Samir Awad");`
- c) `ADD TO users VALUES("Samir@gmail.com", "Samir Awad");`
- d) `ALTER TABLE users INSERT VALUES("Samir@gmail.com", "Samir Awad");`

**2.14) Si nous voulons que chaque fois que nous mettons à jour le app\_id dans les applications, toutes les autres références soient également mises à jour, et chaque fois que nous le supprimons, nous remplacerons toutes les références avec null.**

- a) `ON DELETE CASCADE ON UPDATE CASCADE`
- b) `ON DELETE SET NULL ON UPDATE CASCADE`
- c) `ON DELETE CASCADE ON UPDATE SET NULL`
- d) `ON DELETE SET NULL ON UPDATE SET NULL`