**National University of Computer and Emerging Sciences, Lahore Campus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Course:** | **Database Systems** | **Course Code:** | **CS2005** |
|  | **Program:** | **BS ( CS)** | **Semester:** | **Fall 2023** |
|  | **Out Date:** |  | **Total Marks:** | **78** |
|  | **Due Date:** |  | **Weight:** |  |
|  | **Section** |  | **Page(s):** | **3** |
|  | **Assignment:** | **1 (Relational Model)** |  |  |

**Instructions:** • This assignment is an individual assignment.

* Clearly mention any assumption you have made.
* You are required to submit the hard copy of your assignment at the start of your class.
* For any query, please contact your TA.

**TOPIC: Relational Data Model**

**Q1. Consider the following database state: (total points =20 )**

**Users Cards UserCard CardType**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | UserId | Name | City | | 1 | William | Paris | | 2 | Henry | London | | 3 | John | Dublin | | 4 | Maria | Moscow | | 5 | James | Valencia | | 6 | William | Venice | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | cardNum | cardTypeId | Pin | Balance |  | UserId | cardNum | | 1234 | 1 | 2281 | 248162 | 1 | 1234 | | 2851 | 1 | 4159 | 319875 | 2 | 2851 | | 3948 | 2 | 3478 | 215734 | 3 | 3948 | | 4236 | 2 | 4159 | 198521 | 5 | 4236 | | 5725 | 2 | 1955 | 106780 | 1 | 5725 | | |  |  | | --- | --- | | cardTypeId | Name | | 1 | Debit | | 2 | Credit | |

Primary keys are underlined. CardTypeId is foreign key in Cards. UserId and cardNum are foreign keys in UserCard.

Apply following operations on the above database. State if the operation would be carried out successfully or not. In case of successful operation indicate the changes that will be made to the above database. Also state all the integrity constraints violated by each operation, if any. Note that all following operations are independent.

* **Insert Operations**
* Insert into Users values (10,‘Marsh’,Venice)
* Insert into cards values (‘1422’,3,’1255’,254036)
* Insert into UserCard values (1,’2449’)
* Insert into UserCard values (2,’2851’)
* Insert into CardType values (‘3’,‘Master Card’)

* **Update Operations** (Assume that foreign keys with on update cascade )
* Update CardType Set cardTypeId=3 where cardTypeId=1
* Update Users Set UserId=9 where UserId=6
* Update Cards Set cardNum= 1111 where cardNum=2851
* Update UserCard set UserId= 6 where UserId = 3
* Update Cards set cardTypeId=2 where CardNum=1234

* **Delete Operations** (Assume that foreign keys with Set Null Option is Implemented)
* Delete from CardType where cardtypeId > 1
* Delete from UserCard where userId=1
* Delete from Users where userId < 3
* Delete from Cards where cardTypeId=2
* Delete from Users where Name=’William’

* **Delete Operations** (Assume that foreign keys with Cascade option is Implemented)
* Delete from Users where UserId > 3
* Delete from UserCard where UserId < 2
* Delete from CardType where cardTypeId<3
* Delete from Cards where Pin=4159
* Delete from Users where Name= ’James’

**QUESTION NO 2 ( 7X4 each 28 POINTS )**

1.write a sql query that uses a subquery to find the users who have cards with a balance greater than the balance of all users.

2.Write a SQL query that uses a subquery to find the card type of MemberID = 1.

3.Write a SQL query that uses a subquery to find the card type of members from the city 'Dublin.'

4. Write a SQL subquery to find the total balance across all cards for the user with the highest balance among the users in Moscow

5. Write a SQL subquery to find the names of users who have Debit card and a card balance greater

than average card balance of all users.

6. Write a SQL subquery to find the names of the users who have both Debit and credit cards.

7. Write a SQL subquery to find the avg balance of the users who have Debit card and live in

cities starting with letter V.

**QUESTION NO 3 (TOTAL POINTS =15)**

* **Give output for the following queries(also please show your working in the solution as well )**

1. SELECT u.Name AS Member\_Name, c.cardNum AS Card\_Number

FROM Users u

JOIN UserCard uc ON u.UserId = uc.UserId

JOIN Cards c ON uc.cardNum = c.cardNum

WHERE u.Name LIKE 'William%';

2. SELECT u.Name AS Member\_Name, ct.Name AS Card\_Type

FROM Users u

JOIN UserCard uc ON u.UserId = uc.UserId

JOIN Cards c ON uc.cardNum = c.cardNum

JOIN CardType ct ON c.cardTypeId = ct.CardTypeId

WHERE ct.Name = 'Credit';

3. SELECT u.Name AS Member\_Name, uc.cardNum AS Debit\_Card\_Number

FROM Users u

JOIN UserCard uc ON u.UserId = uc.UserId

JOIN CardType ct ON c.cardTypeId = ct.CardTypeId

WHERE ct.Name = 'Debit';

**(TOTAL POINTS =15)**

**Q2.** Snap Fitness is a fast-expanding gym company that now has several branches throughout the Americas, with plans to open up another 12 branches over the next two years. To date, the company has used a paper-based system to store details of its members, staff, and fitness classes. Currently, when members attend the gym, they have to sign in when they attend the gym. Due to ambitious expansion plans senior management has appointed you to design and implement a database system that will hold details of members including attendance, staff, and fitness classes. When a member joins the gym, they have to provide a range of personal details. They also have to specify what their main fitness goal is. Members can choose from a range of membership types that provide them with access to just their local gym or they can choose a membership that gives them access to all gyms within the group. Additionally, there is a peak and off-peak membership type. Peak members can attend the gym at any time of the day or night, whereas off-peak members can only attend between the hours of 9 am-3 pm and 8 pm – 6 am. With the new system, members will be provided with a membership card – similar to a credit or debit card that has a chip embedded into it. When they attend the gym, they will have to place their membership card on a chip reader which allows them access to the gym and repeat the process when they leave the gym. This will record the date and time of each member when they arrived and when they left the gym, this will help management identify at what times the gym is busiest and quietest. When members sign up for the gym, they have to set up a direct debit, and payments are taken on the month anniversary of their joining date. If a member misses a payment their subscription is temporarily suspended until the payment is received (which may be made by card payment). Each gym runs its own timetable of fitness classes which is run by a member of staff from that gym. Members must book in advance to attend the class. Classes are free as they form part of the monthly membership cost. With the current manual recordkeeping process, it has been noted that there are data inconsistencies, such as members not signing in, and loss of data integrity as the current system has evolved. This, along with growth, has resulted in a situation where Snap Fitness requires a relational database system to store, process, and report on the gym’s data.

**You are required to:**

* Identify the relations/tables required. The attributes of each relation must have the appropriate data types/domains.
* Identify the primary keys, secondary keys (if any), and foreign keys (if any) in each relation.
* Populate each relation with at least 5 sample tuples so that none of the constraints is violated.