INFS7900 Module 3 Assignment

Due: 19 May 2023 @ 4:00 PM AEST

Weighting: 30%

Full Name	Student ID (8 digits)
Vic, Hong	**47523483**

Overview

The purpose of this assignment is to test your ability to use and apply SQL concepts to complete tasks in a real-world scenario. Specifically, this assessment will examine your ability to use SQL Data Manipulation Language to return specific subsets of information which exist in a database and Data Definition Language to create new relational schema. The assignment is to be done **individually**

Submission

All submissions for **Sections A – D** must be made through an electronic marking tool called Gradescope, which will also be used for providing feedback. You **must** record all your answers in the spaces provided in this document. Altering the format or layout of this document in anyway will attract penalties. All submissions must have the above boxes filled out to be identified. **Section E** is to be completed through the RiPPLE platform (link available on Blackboard).

Marking

The module 3 assignment is worth **30 course marks** (of 100 course marks total for all assessment. The marking distribution per section is as follows:

1. Section A – SQL DDL: 4 marks
2. Section B – SQL DML (UPDATE, INSERT, DELETE): 5 marks
3. Section C – SQL DML (SELECT): 16 marks
4. Section D – Critical Thinking: 2.5 marks
5. Section E – RiPPLE Task: 2.5 marks

Plagiarism

The University has strict policies regarding plagiarism. Penalties for engaging in unacceptable behaviour range from loss of grades in a course through to expulsion from UQ. You are required to read and understand the policies on academic integrity and plagiarism in the course profile (Section 6.1). If you have any questions regarding acceptable level of collaboration with your peers, please see either the lecturer or your tutor for guidance. Remember that ignorance is not a defence!

In particular, you are permitted to use generative AI tools to help you complete this assessment task. However, if you do, please provide complete copies of your interactions with the AI tool in the space provided at the end of your submission. Please note that if you use generative AI but fail to acknowledge this by attaching your interaction to the end of the assignment, it will be considered misconduct as you are claiming credit for work that is not your own.

6. Task

For this assignment you will be presented with the simplified schema of an event management application. The goal of the application is to track both the events attended by users and relationships between users and other users. The system is then able to use this data to effectively market recommended events to users based on the events their friends have attended. A sample database of this system has been provided here which will allow you to test your queries.

Assignment Specification

Events Inc. is a small start-up company which provides its users with an event tracking and recommendation platform for various local community activities. A simplified version of their database schema has been provided below including foreign key constraints.

Relational Schema

USER [id, first_name, last_name, date_of_birth, phone, email, nationality, significant_other]

EVENT [title, event_location, event_date, description, sponsor]

ATTENDS [user_id, title, event_location, event_date, travel_method]

FRIENDS [requestor, requestee, requested_date, accepted_date]

Foreign Keys

USER.significant_other references USER.id

ATTENDS.{title, event_location, event_date} references EVENT.{title, event_location, event_date}

ATTENDS.user_id references USER.id

FRIENDS.requestor references USER.id

FRIENDS.requestee references USER.id

For this assignment you will be required to write SQL queries to answer to complete the following tasks. Please use the submission boxes provided to record your answers. For queries with a returning relation of more than 10 tuples, you can use the **LIMIT 10** clause to only capture the first 10 tuples of the table.

Example	
Task	Return the first name and last name of all users.
Explanation	This query should return a table with two columns, one for first name and one for last name.

SQL Solution	`SELECT first_name, last_name`
	`FROM USER`
	`LIMIT 10;`

Output Screenshot	

fName	lName
Eduard	Khil
Mikhail	Mishustin
Lucy	Ali
John	Monarch
Ursula	Smith
Marcus	Jacobs
Nevena	Ivanovic
Leo	Montgomery
Edi	Rama
Jamie	Sleeman

Section A – SQL DDL

Question 1

Task

Write a SQL DDL query to implement the following relational schema and associated foreign keys.

Explanation

The relational schema for this the table is as follows:

Table: ATTENDS			
Column	**Data Type**	**Allow Nulls?**	**Primary Key?**
user_id	INT	No	Yes
title	VARCHAR(50)	No	Yes
event_location	VARCHAR(50)	No	Yes
event_date	DATE	No	Yes
travel_method	{“Car”, “Bus”, “Train”, “Other”}	No	No

Additionally, no user should be attending two different events on the same day

The foreign keys for this new table are as follows:
ATTENDS.user_id references USER.id
ATTENDS.{title, event_location, event_date} references
EVENT.{title, event_location, event_date}

Foreign key constraints should be implemented such that:

- Updates to an *event* title, date, or location are automatically updated in the *ATTENDS* table as well.
- A *User* cannot be deleted if they are attending an event

Note: You may wish to consult the MySQL documentation on the enum datatype.
You may create this table using the name NEW_ATTENDS
You may want to consult this tutorial and the use of the following command
COLLATE latin1_swedish_ci;

SQL Solution

```
CREATE TABLE NEW_ATTENDS(
    user_id INT NOT NULL,
    title VARCHAR(200) NOT NULL,
    event_date DATE NOT NULL,
    event_location VARCHAR(200) NOT NULL,
    travel_method ENUM("Car", "Bus", "Train", "Other") NOT NULL,
    PRIMARY KEY(user_id, title, event_date, event_location),
    CONSTRAINT user_id_fk FOREIGN KEY(user_id) REFERENCES USER(id) ON DELETE RESTRICT,
    CONSTRAINT other_fk FOREIGN KEY(title, event_date, event_location) REFERENCES EVENT(title, event_date,
    event_location) ON UPDATE CASCADE
)
COLLATE latin1_swedish_ci;
```


Question 2	
Task	Following a marketing research, Event Inc. has decided to target their events exclusively towards Millennials and Gen Z. Assuming all their users are currently from these generations, update the USER table to restrict the insertion of new users to Millennials and Gen Z.
Explanation	For the sake this question, we use the date ranges below for Millennials and Gen Z: • Millennials: Born 1981-1996 • Gen Z: Born 1997-2012 The following resources may be useful when answering this question: Check constraints
SQL Solution	`ALTER TABLE USER` `ADD CONSTRAINT birthdate_check CHECK (date_of_birth >= "1981-01-01" AND date_of_birth <= "2012-12-31");`

Section B – SQL DML (UPDATE, DELETE, INSERT)

Question 1	
Task	Due to significant Olympic developments, car traffic in Woolloongabba is heavily restricted for the month of May 2023. Delete the attendee registration for anybody using a car to attend an event at 'The Gabba' in May.
Explanation	To identify events held at 'The Gabba' you should check for the presence of 'gabba' within the event_location. MONTH Function YEAR Function LOWER Function

SQL Solution	`DELETE`
	`FROM ATTENDS`
	`WHERE YEAR(ATTENDS.event_date) = 2023`
	`  AND MONTH(ATTENDS.event_date) = 5`
	`  AND ATTENDS.travel_method = "Car"`
	`  AND ATTENDS.event_location LIKE "%gabba%";`

Question 2	
Task	Coca-Cola has decided to increase their advertisement in Australia by taking over the sponsorship of any events that happens at the "Sydney Opera House". Update the database to reflect this change.

SQL Solution	`UPDATE EVENT`
	`SET EVENT.sponsor = "Coca-Cola"`
	`WHERE EVENT.event_location = "Sydney Opera House";`

Section C – SQL DML (SELECT)

Question 1	
Task	Find an alphabetical list of all the event locations where people are arriving via train.

SQL Solution	`SELECT DISTINCT event_location`
	`FROM ATTENDS`
	`WHERE travel_method = "Train"`
	`ORDER BY event_location ASC`
	`LIMIT 10;`

Output Screenshot	←T→ | event_location ▲ 1 ☐ 編輯 複製 刪除 600 Gregory Tce, Bowen Hills ☐ 編輯 複製 刪除 Flemington Racecourse ☐ 編輯 複製 刪除 St Lucia, QLD 4072 ☐ 編輯 複製 刪除 Surfer's Paradise, QLD ☐ 編輯 複製 刪除 Sydney Opera House

Question 2	
Task	Find the first and last names of all users who have had a successful friend request in 2023
Explanation	A successful friend request will have a date in the *accepted_date* column

SQL Solution	
	`SELECT first_name, last_name`
	`FROM USER`
	`JOIN FRIENDS ON USER.id = FRIENDS.requestor`
	`WHERE YEAR(FRIENDS.accepted_date) = "2023";`

Output Screenshot	 first_name | last_name Jeong–hyeok | Ri Bong–soon | Park

Question 3	
Task	List all event locations and the number of attendees, ordered by the number of attendees in descending order.
SQL Solution	

```
SELECT event_location, COUNT(*) AS attendee_count
FROM ATTENDS
WHERE event_location IN (
    SELECT DISTINCT event_location
    FROM ATTENDS
    )
GROUP BY event_location
ORDER BY attendee_count DESC;
```

Output Screenshot

event_location	attendee_count 1
600 Gregory Tce, Bowen Hills	32
St Lucia, QLD 4072	24
Flemington Racecourse	16
Sydney Opera House	12
Bathurst, NSW	11
Surfer's Paradise, QLD	7
Parliament House, Canberra	2

Question 4	
Task	Find the first and last names of all people whose significant other is of a different nationality to them.
SQL Solution	

```
SELECT user_1.first_name, user_1.last_name
FROM USER user_1
JOIN USER user_2 ON user_1.significant_other = user_2.id
WHERE user_1.nationality != user_2.nationality
LIMIT 10;
```

Output Screenshot

first_name	last_name
Mikhail	Mishustin
Lucy	Ali
John	Monarch
Ursula	Smith
Marcus	Jacobs
Nevena	Ivanovic
Leo	Montgomery
Jamie	Sleeman
Honore	Avare
Margit	Gade

Question 5	
Task	Find the first and last name of the person(s) who has attended the most events.
Explanation	Attending the same event on two different dates should count multiple times.
SQL Solution	

```
SELECT first_name, last_name
FROM USER
JOIN (
    SELECT user_id, COUNT(*) AS attend_time
    FROM ATTENDS
    GROUP BY ATTENDS.user_id
    ORDER BY attend_time DESC
    LIMIT 1) AS most_attendee ON USER.id = most_attendee.user_id;
```

Output Screenshot

first_name	last_name
Marcus	Jacobs

Question 6	
Task	Find the first and last names of all users who have a significant other, have sent at least 10 requests to *Australian* users, and have used a bus or train to attend at least 5 events.
Explanation	Attending the same event on two different dates should count multiple times.

```
SELECT user_1.first_name, user_1.last_name
FROM USER user_1
JOIN ATTENDS ON ATTENDS.user_id = user_1.id
JOIN FRIENDS ON FRIENDS.requestor = user_1.id
JOIN USER user_2 ON FRIENDS.requestee = user_2.id
WHERE user_1.significant_other IS NOT NULL
GROUP BY user_1.id
HAVING SUM(CASE WHEN ATTENDS.travel_method = "Bus" OR ATTENDS.travel_method = "Train" THEN 1 ELSE 0 END) >= 5
    AND COUNT(CASE WHEN user_2.nationality = "Australia" THEN 1 ELSE 0 END) >= 10;
```

Output Screenshot

first_name	last_name
Marcus	Jacobs
Eduard	Khil
Lucy	Ali
John	Monarch
Jamie	Sleeman

Question 7	
Task	Find the users who have attended at least all the same events as “Ursula Smith”
Explanation	Ursula Smith should also be returned.
Output Screenshot	

Question 8	
Task	Find the first & last names, as well as the number of friends each user has who have attended any event. Order the result in descending number of friends.
Explanation	The user could be either the requestee or requestor. Remember that if the user is a requestor, the other user must have accepted the request to be classed as a friend. If a user has no successful friend requests they may be excluded from the result/ **Hint**. You may want to use one or more views in your answer.

SQL Solution	`SELECT first_name, last_name, COUNT(FRIENDS.accepted_date) AS friend_count`
	`FROM USER`
	`JOIN FRIENDS ON FRIENDS.requestor = USER.id`
	`JOIN ATTENDS ON FRIENDS.requestee = ATTENDS.user_id`
	`GROUP BY USER.id`
	`ORDER BY friend_count DESC`
	`LIMIT 10;`

Output Screenshot	

first_name	last_name	friends_count 1
Lucy	Ali	62
John	Monarch	50
Jamie	Sleeman	39
Marcus	Jacobs	34
Eduard	Khil	32
Mikhail	Mishustin	28
Sven	Kirsch	24
Ursula	Smith	12
Bong-soon	Park	12
Hye-sun	Ku	9

Section D – Critical Thinking

1. In this section, you will receive theoretical situations related to the UoD mentioned in the task description. Your task is to offer strategies to tackle the situation and write SQL queries to execute the approaches.

Question 1	
Task	The company is seeking to expand into events marketing and wants to identify 10 key influencers to assist them in boosting attendance numbers for events, both globally and locally. Propose a strategy for identifying users that may be a good fit for the role and write an SQL query to implement the strategy.
Strategies	The company wish to improve attendees, as a strategy, we could select the ten attendees who attended the most event, which means they have passion to participate such these events, ask them to mention the event that will be hold soon to their friends. This may be good to stimulate the attendees amount because the friends of these ten people's might be interested in these events.

SQL Solution	`SELECT first_name, last_name`
	`FROM USER`
	`JOIN(`
	`  SELECT user_id, COUNT(*) AS attend_time`
	`  FROM ATTENDS`
	`  GROUP BY ATTENDS.user_id`
	`  ORDER BY attend_time DESC`
	`  LIMIT 10) AS ten_most_attendees ON USER.id = ten_most_attendees.user_id;`

Question 2	
Task	The company has noticed that an abnormally large number of single tickets have been purchased for recent events. The finance team has indicated that they have enough budget to offer a sign-up bonus for users who refer their significant other to the platform. There is only enough budget to offer this to 20 users. Propose a strategy to identify the users that the company could offer this promotion to, and write an SQL query(s) to create a prioritised list of those users.
Strategies	The task aims to choose 20 users who refer their significant other into the platform to give them sign-up bonus. Therefore, we can use RAND() to randomly re-order the user position, then use LIMIT 20 choose the first 20 users in the output, which means these 20 users could be rewarded the bonus prize.

SQL Solution	`SELECT id`
	`FROM USER`
	`WHERE USER.significant_other IS NOT NULL`
	`ORDER BY RAND()`
	`LIMIT 20;`

Section E – RiPPLE Task

Using the RiPPLE online software, you must complete the following activities before the assignment due date:

- **Resource Creation**: Create one or more effective resource. For a learning resource to be considered as effective it needs to pass a moderation process which is administered by your peers and the teaching team. Teaching staff will be spot-checking to review moderations performed by just peers and change the outcome if necessary.

- **Resource Moderation**: Moderate five or more resources effectively. An effective moderation means that you have completed the moderation rubric and have provided a detailed justification for your judgement as well as constructive feedback on how the resource can be improved. Simply saying a resource is "good" does not qualify. Again, teaching staff will be spot-checking the quality of moderations and change the outcome when necessary.

- **Answering Questions**: Answer 10 or more questions correctly. To answer a resource correctly your first response must be correct. You can attempt as many questions as you want, and incorrect answers do not count against you. Only answers from the Practice tab are counted. Answering in-class RiPPLE activity questions does not count towards questions answers.

These tasks are to be completed through the RiPPLE platform, via the link available on Blackboard.

Note: For the above three activities, the resources you create, moderate and answer **must** be in the following categories on RiPPLE:

- SQL

- Functional-dependency
- Normalization

Creating, moderating or answering questions from other categories will not be counted towards your mark for the RiPPLE component of this assignment.

Documenting the use of Generative AI

Please note that if you have used generative AI in any manner, you are required to provide a transcript of your engagement with the system in this section. You can simply copy and paste your discussion with the generative AI system below. It is fine if it goes across multiple pages.

A reminder that a failure to reference AI use may constitute student misconduct under the Student Code of Conduct.