DBMS: LAB 7

Advanced SQL

University Fest Management System

OBJECTIVE: To learn and understand **Advanced SQL queries** (Nested, Correlated, Window Functions, Full Text Search Functions).

INSTRUCTIONS:

- As a part of LAB 7, there are 4 tasks that are to be completed wrt to the case study shared earlier with ER diagram and Relational Schema of University Fest Management.
  - **TASK 1:** Write and execute **Nested** queries on the tables to retrieve the result according to the question.

  - **TASK 2:** Write and execute **Correlated** queries on the tables to retrieve the result according to the question.

  - **TASK 3:** Write and execute **Window Function** queries on the tables to retrieve the result according to the question.

  - **TASK 4:** Write and execute **Full Text Search** queries on the tables to retrieve the result according to the question. Before you write the query, you must alter the table to ensure that it is compatible for full text search

- As a part of the submission process, the following are to be submitted:
  - A **PDF** document, containing all the Screenshots for all tasks as suggested
    - Name of the file: `<your SRN>_University_Fest_DB_Lab7.pdf

  - The **".sql"** file for the same, shall contain all the commands that have been executed in the lab
    - Name of the file: `<your SRN>_University_Fest_DB_Lab7.sql

Example:

Refer to the sample submissions given below. This will give you an idea about the details that must be included in your submissions

NOTE: Screenshots can be taken either from "**MySQL workbench**" or "**Command Line**"

For every Query, **screenshot must have**:

1. SQL Query for the question
2. The result rows displayed

Sample submission:

<u>COMMAND:</u>

SELECT fest_name

   FROM fest

   WHERE year = (

      SELECT year

      FROM fest

      WHERE fest_id = 'F201'

  );

<u>Output:</u>

```
mysql> SELECT fest_name
    -> FROM fest
    -> WHERE year = (
    ->      SELECT year
    ->      FROM fest
    ->      WHERE fest_id = 'F201'
    -> );
+------------+
| fest_name  |
+------------+
| Samarpana  |
| Graduation |
+------------+
2 rows in set (0.00 sec)
```

Task 1: Nested Queries

Q1. List the names of participants who registered for events with a price above the average event price.

Q2. Show events that have more registrations than the average number of registrations per event.

Task 2: Correlated Queries

Q3. Find participants who have registered for the highest-priced event(s)

Q4. Find stalls that offer items cheaper than the average price of that item across all stalls.

Task 3: Window Functions

Q5. Show the SRN and name of participants and how many times they purchased something, along with their rank based on total quantity bought

Q6. Rank participants within their department based on the number of events they've registered for (should have 4 columns - department, name, total_number_of_events and rank).

Task 4: Full Text Search

Q7. Ranked search: Order events by relevance to the word "Tournament"

Q8. List stalls offering items whose names match "chicken" or "soup", and compute total stock per stall