|  |  |
| --- | --- |
| **Ex. No: 05** | **SUBQUERIES AND CORRELATED SUBQUERIES** |
| Date | 12.09.2023 |

**Objective:**

To execute given query using the concept of subqueries and correlated

subqueries

**Software Required:**

Oracle 10g

**Description:**

Subquery is usually added in the WHERE Clause of the sql statement. Most

of the time, a subquery is used when you know how to search for a value using a

SELECT statement, but do not know the exact value. Subqueries are an alternate

way of returning data from multiple tables.

Subqueries can be used with the following sql statements along with the comparison

operators like =, >, <, <=, >= etc.

* Update
* Insert
* Select
* Delete

**Correlated sub-query**

A correlated sub-query is a term used for specific types of queries in SQL in

computer databases. It is a sub-query (a query nested inside another query) that

uses values from the outer query in its WHERE clause. The sub-query is evaluated

once for each row processed by the outer query.

**Detailed Procedure:**

Here is an example for a typical correlated sub-query. In this example we are finding

the list of employees (employee number and names) having more salary than the

average salary of all employees in that employee department.

SELECT employee\_number, name FROM employee AS e1

WHERE salary &gt; (SELECT avg(salary)FROM employee

WHERE department = e1.department);

In the above query the outer query is,

SELECT employee\_number, name FROM employee AS e1

WHERE salary >=

And the inner query is,

(SELECT avg(salary)

FROM employee

WHERE department = e1.department);

In the above nested query the inner query has to be executed for every

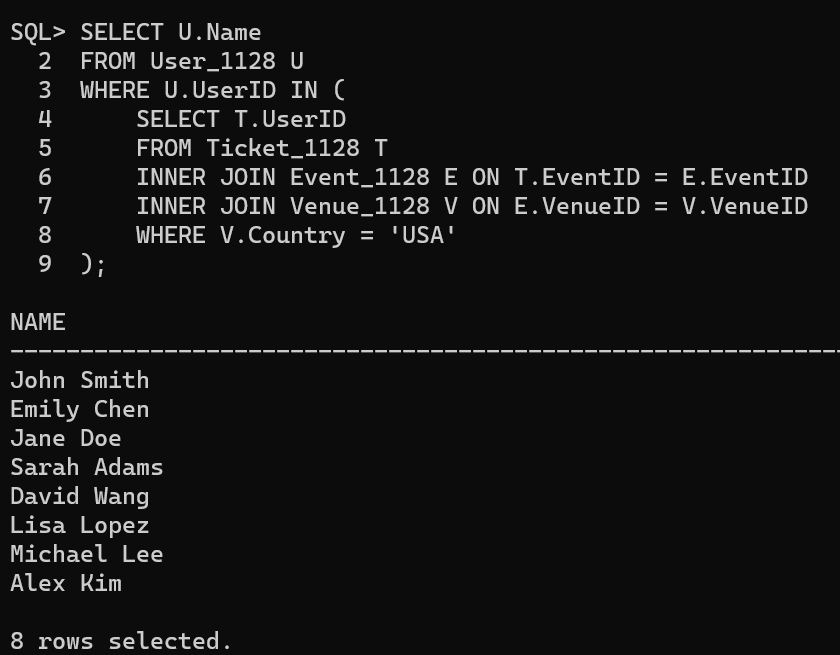
employee as the department will change for every row. Hence the average salary will

also change. The effect of correlated sub-queries can also be obtained using outer

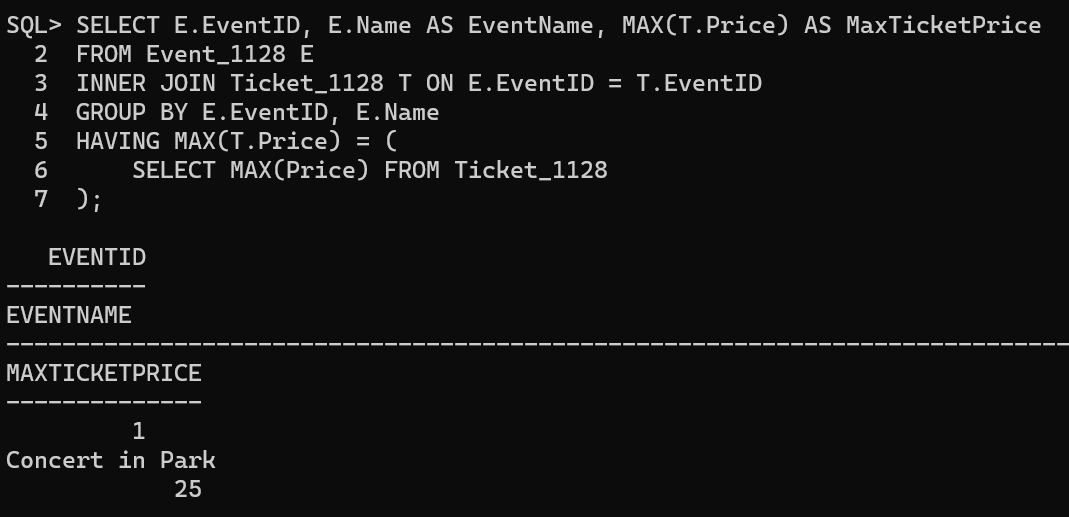
Joins.

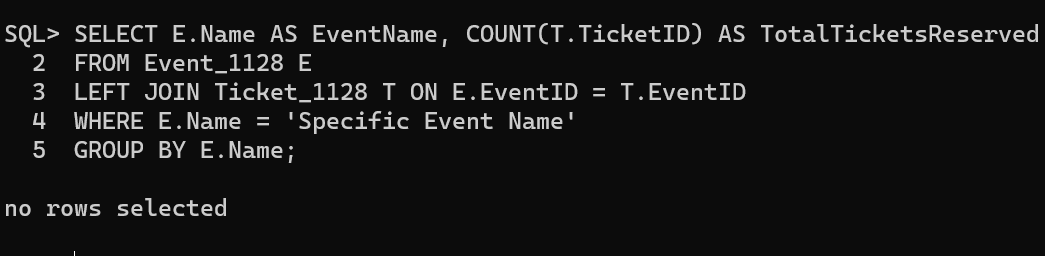
**Questions:**

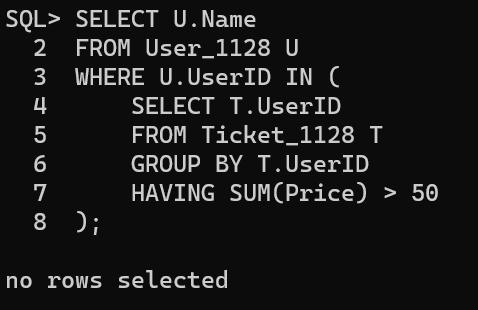
1. List all users who have made reservations for events that are taking place in a specific venue (e.g., "USA").



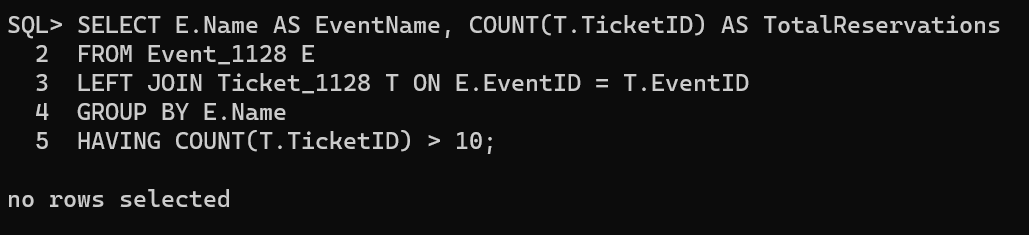
1. Find the events with the highest ticket prices.



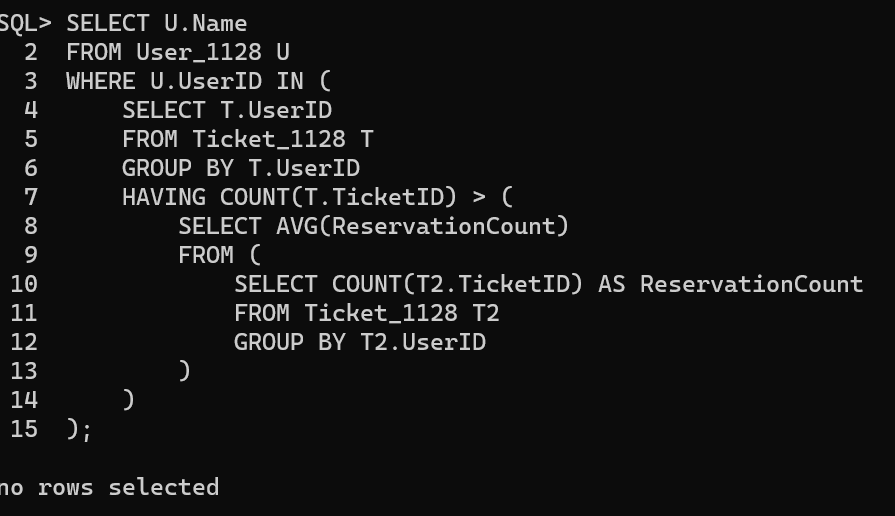
1. Find the total number of tickets reserved for a specific event. 
2. List the users who have made reservations with a total cost exceeding a certain amount (e.g., 50).



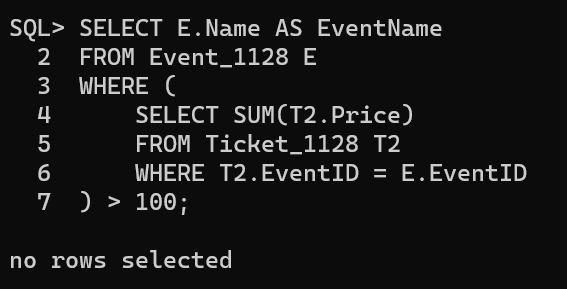
1. Retrieve the events where the number of reservations exceeds a certain threshold.



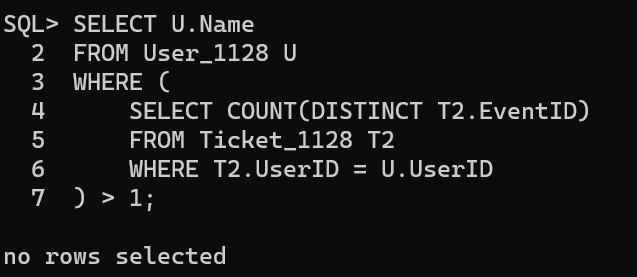
1. Find all users who have made more reservations than the average number of reservations across all users.



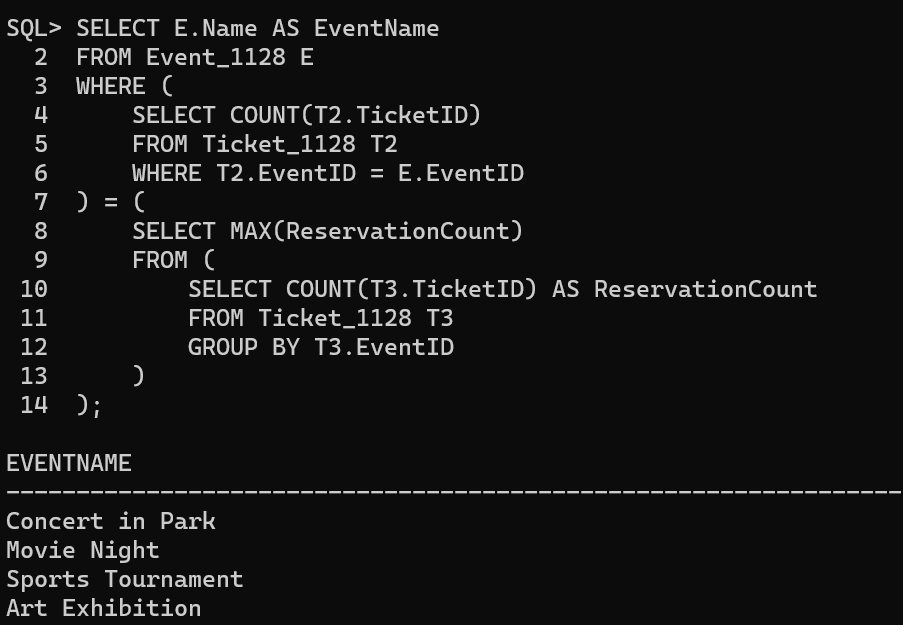
1. List all events where the total ticket price of reservations exceeds a certain amount.

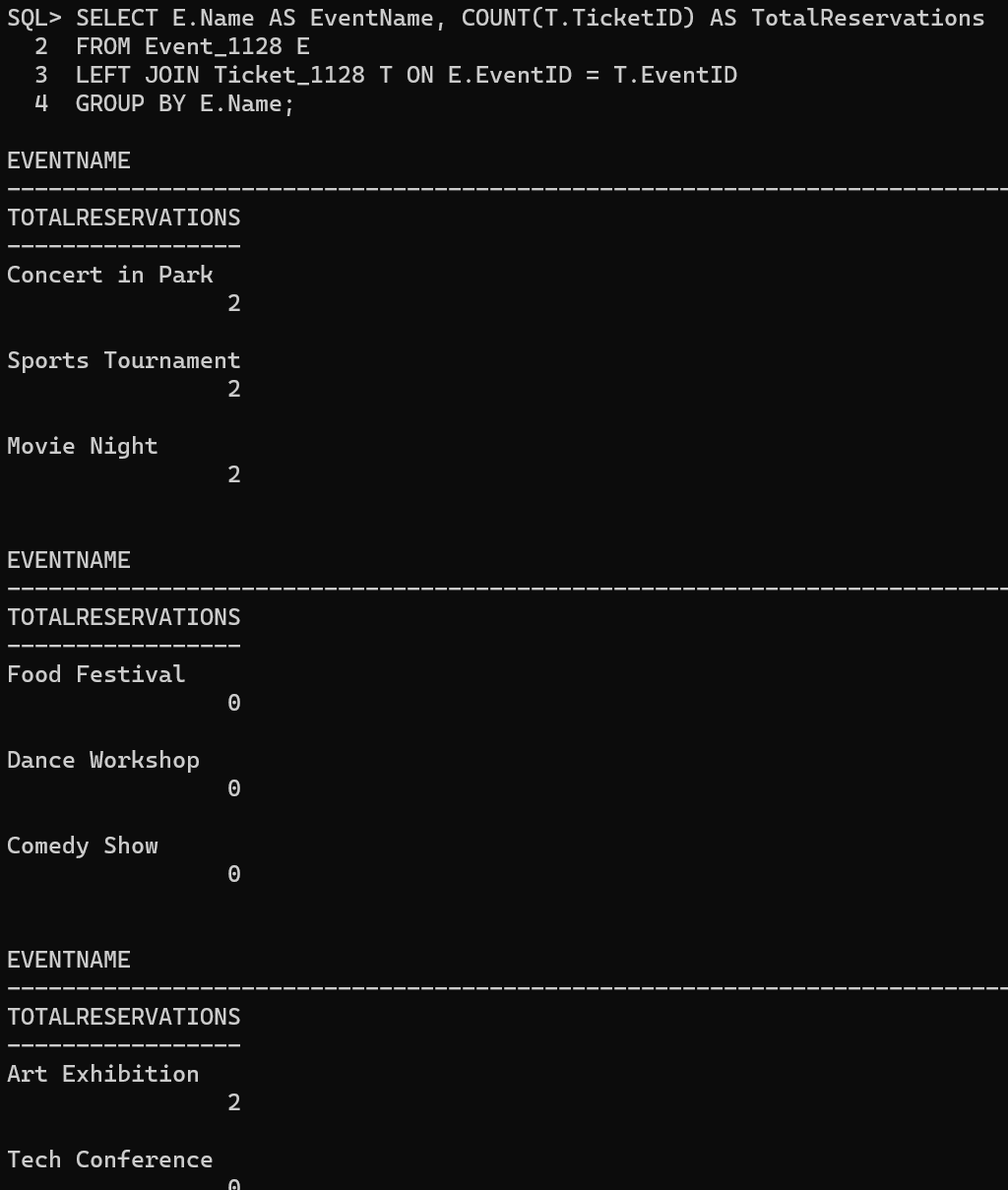


1. Find the users who have made reservations for more than one event.



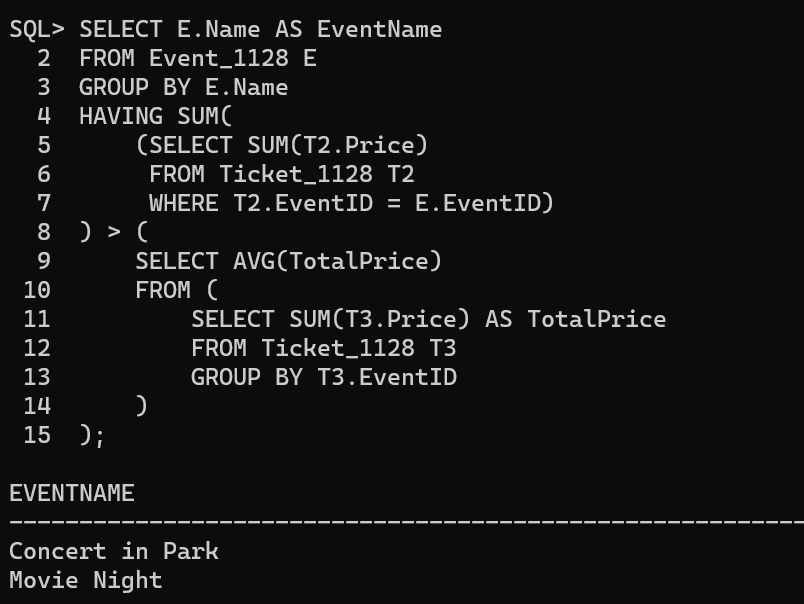
1. Retrieve the events with the highest number of reservations.



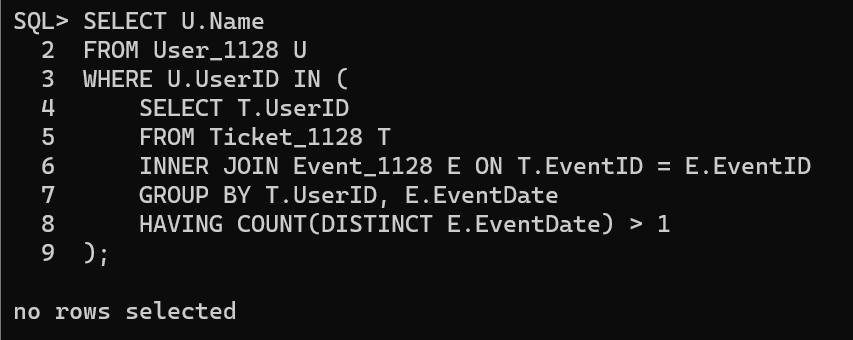
1. For each event, find the number of reservations made by users.



1. Find the events for which the total ticket price of reservations exceeds the average total ticket price for all events.



1. List users who have made reservations for multiple events on the same day.



**Result:**

The given queries were executed successfully using set operators and joins.