**Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology**

**(Deemed to be University Estd. u/s 3 of UGC Act, 1956)**

**School of Computing**

**B.Tech. – Computer Science and Engineering**

**VTR UGE2021- (CBCS)**

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name: Database Management Systems

Slot No : S2L5

DBMS TASK4-REPORT

**Title:** Using Functions in Queries and Writing Subqueries

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **VTUNO** | **REGISTER NUMBER** | **STUDENT NAME** |
| VTU29510 | 24UECS0249 | P.RAM CHARAN |

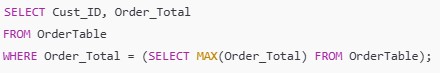
**Task 4: Using Functions in Queries and Writing Subqueries**

**Case Study**: Online Food Ordering System

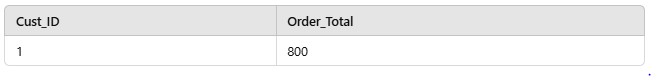
**Objective**: To perform advanced query processing and test heuristics by designing optimal correlated and nested subqueries, such as finding summary statistics, for the Online Food Ordering System.

1. **Using Aggregate Functions with Subqueries**

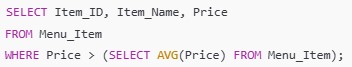
**Query 1: Find the customer(s) who placed the highest order total.**



**Output:**



**Query 2: List all menu items whose price is above the average price of all menu items.**

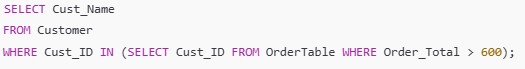


**Output:**

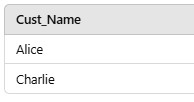


1. **Nested Subqueries**

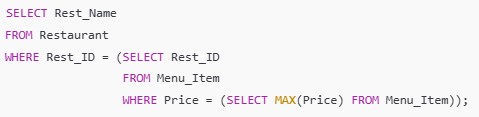
**Query 1: Find the names of customers who placed orders worth more than 600.**



**Output:**



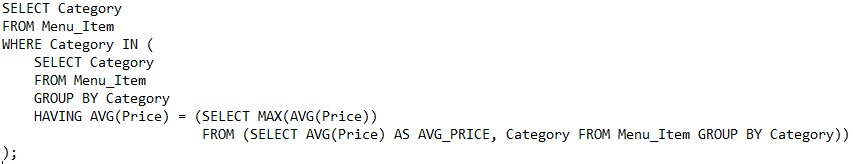
**Query 2: Retrieve the name of the restaurant(s) offering the most expensive menu item.**



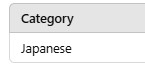
**Output:**



**Query 1: Retrieve the category of menu items with the highest average price.**



**Output:**

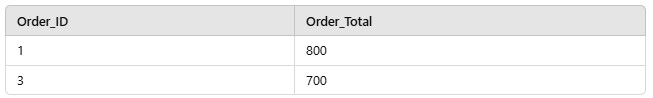


1. **Correlated Subqueries**

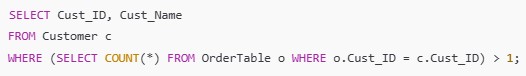
**Query 1: Find all orders where the total is greater than the average total of all orders.**



**Output:**



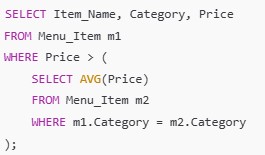
**Query 2: Find customers who have placed more than one order.**



**Output:**



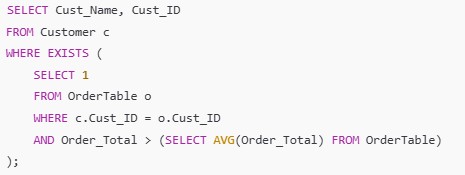
**Query 3: Retrieve the list of menu items priced above the average price for their category.**



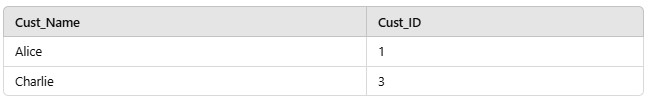
**Output:**



**Query 4: Find customers who have placed orders with totals higher than the average order total of all customers.**

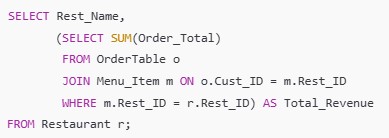


**Output:**

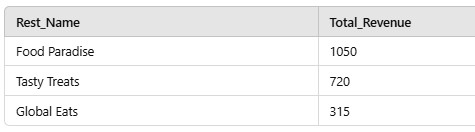


1. **Summary Statistics with Subqueries**

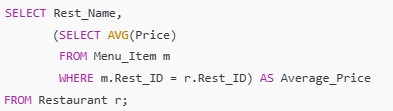
**Query 1: Retrieve the total revenue generated by each restaurant.**



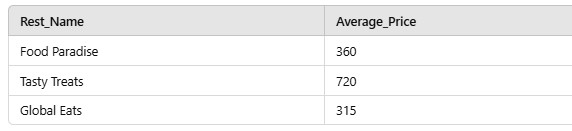
**Output:**



**Query 2: Find the average price of menu items for each restaurant.**



**Output:**



RESULT:Thus the task has been executed and verified successfully.