Database Mini Competition

City University of Hong Kong

October 02, 2023

Reminders (please read carefully)

1. The Computer doesn't have the compiler to run your code. However, you can use the test case in **exercise** for testing and debugging. Don't forget to submit your code to the **exam**.

2. To determine the final ranking, the students' scores will be sorted in descending order. In case of a tie in the final scores, the student who has spent less time will be assigned a higher rank.

3. The submitted code should be no longer than 7 kilobytes (around 7000 characters).

SQL Contest

Question 1

作為一個在大學中廣泛使用的在線學習平臺，Canvas一直在追蹤每一個學生的登錄時間。這個時間被存儲在 `Login_Q1` 表裏，如下：

Canvas, the online learning platform used by the university, keeps track of each student's login time. The login time is stored in the `Login_Q1` table, which has the following schema:

Column	Type
StudentID	INTEGER
LoginTime	DATETIME

管理員對在復習周（從2023-04-17到2023-04-22）中每天都登錄到Canvas上的學生感興趣。請寫一個SQL詢問來查詢這些學生，輸出格式如下：

The administrator is interested in the number of students who have logged in Canvas everyday during the revision week (from 2023-04-17 to 2023-04-22). Write a SQL query to find those students. The output of your query should be a table with the following schema:

Column	Type
StudentID	INTEGER

Question 2

在在線評測系統中有一些題，他們分別有著唯一的ID。題目的具體資訊存儲在 `Problem_Q2` 表裡，如下：

The Online Judge system consists of a number of problems, and each problem has a unique ID. The details of each problem are stored in the `Problem_Q2` table, which has the following schema:

Column	Type
ProblemID	INTEGER
ProblemName	VARCHAR(255)
ProblemDescription	VARCHAR(255)

在線評測係統還會跟蹤每一道題的每一次提交。提交細節存儲在 `Submission_Q2` 表裡，如下：

Moreover, the Online Judge system keeps track of the submissions of each problem. The details of each submission are stored in the `Submission_Q2` table, which has the following schema:

Column	Type
SubmissionID	INTEGER
ParticipantID	INTEGER
ProblemID	INTEGER
SubmissionTime	DATETIME
Score	INTEGER
Status	VARCHAR(255)

其中， `Status` 一列存的是提交的狀態，有如下幾種取值：Accepted，Wrong Answer，等等。

The `Status` column indicates the status of the submission, which can be one of the following values: Accepted, Wrong Answer, etc.

給定以上資訊，你需要計算每一道題的通過率，輸出格式如下：

Given the information above, you are required to compute the acceptance rate for each probelm, the output of which is a table with the following schema:

Column	Type
ProblemID	INTEGER
AcceptanceRate	FLOAT

定義通過率是每道題通過的提交數量除以總的提交數量。其中，通過率是NULL 意味著沒人嘗試過這道題。輸出的表應該按照 `ProblemID` 的升序排序。通過率保留兩位小數。

The AcceptanceRate is defined as the number of accepted submissions divided by the total number of submissions for each problem. Specially, the acceptance rate is NULL if no one has tried the problem yet. The output table should be sorted by `ProblemID` in ascending order. Please note that the `AcceptanceRate` should be rounded to 2 decimal places.

Question 3

在SQL比賽的同時還進行著另外一場編程比賽。這個比賽有上百個參與者，每個人有唯一的ID和用戶名。具體資訊存儲在 `Participant_Q3` 表裡，如下：

A programming contest is running concurrently with the SQL contest the SQL contest. The programming contest has hundreds of participants, and each participant has a unique ID and a username. The details of each participant are stored in the `Participant_Q3` table, which has the following schema:

Column	Type
ParticipantID	INTEGER
Username	VARCHAR(255)
Email	VARCHAR(255)
RegistrationTime	DATETIME

這個比賽有4道題，每道題有唯一的ID從1到4。具體資訊存儲在 `Problem_Q3` 表裡，如下：

The programming contest has 4 problems, and each problem has a unique ID from 1 to 4. The details of each problem are stored in the `problem_Q3` table, which has the following schema:

Column	Type
ProblemID	INTEGER
ProblemName	VARCHAR(255)
ProblemDescription	VARCHAR(255)

編程比賽會有大量的提交，每個提交都有唯一的ID。具體資訊存儲在 `Submission_Q3` 表裡，如下：

The programming contest will have tons of submissions, and each submission has a unique ID. The details of each submission are stored in the `submission_Q3` table, which has the following schema:

Column	Type
SubmissionID	INTEGER
ParticipantID	INTEGER
ProblemID	INTEGER

Column	Type
SubmissionTime	DATETIME
Score	INTEGER
Status	VARCHAR(255)

基於以上資訊，你需要實現“積分表”，輸出格式如下：

Given the information above, you are required to achieve the "scoreboard" utility, the output of which is a table with the following schema:

Column	Type
ParticipantID	INTEGER
TotalScore	INTEGER

其中，表要按照 `TotalScore` 的降序排序。注意，一個人的 `TotalScore` 等於這個人每道題的最高分之和。如果有兩個人的分數相同， `ParticipantID` 小的先輸出。

The rows should be sorted by `TotalScore` in descending order. Please note that TotalScore equals the sum of the highest score of each problem. If two participants have the same `TotalScore` , the one with the smaller `ParticipantID` should be placed ealier.

Question 4

ShopNet，作為電子商務領域的領導者，在多個國家擁有龐大的客戶群體，並致力於瞭解全球產品偏好。

ShopNet, a leader in e-commerce, has an extensive customer base across countries and aims to understand global product preferences.

數據庫模式:

Database Schema:

`Customers_Q4` 表裡記錄著所有註冊過的用戶。

`Customers_Q4` table stores all the registered users.

Field	Type	Description
customer_id	Integer	Unique identifier for each customer
customer_name	Varchar(255)	Name of the customer
country	Varchar(255)	Country of the customer

`Orders_Q4` 表裡記錄著所有交易記錄。

`Orders_Q4` table records all transactions.

Field	Type	Description
order_id	Integer	Unique identifier for each order
customer_id	Integer	Reference to the Customers table
order_date	Date	Date of the order
total_amount	Decimal	Total cost of the order

`Order_Items_Q4` 表裡記錄著每一次交易的物品資訊。

`Order_Items_Q4` table stores the products corresponding to each order. Note that one order may involve multiple different products.

Field	Type	Description
order_item_id	Integer	Unique identifier for each order item
order_id	Integer	Reference to the Orders table

Field	Type	Description
product_name	Varchar(255)	Name of the product
quantity	Integer	Quantity of the product ordered
unit_price	Decimal	Price per product

請編寫一個SQL 查詢，找出每個國家（country）的顧客訂單數量最多的產品（product_name）及其銷售數量（quantity）。

Please write an SQL query to find the product with the highest number of customer orders in each country (productname) and its sales quantity (quantity).

輸出格式:

Expected Output Schema:

Field	Type	Description
country	Varchar(255)	Country of the customer
product_name	Varchar(255)	Name of the most popular product
quantity	Integer	Total quantity ordered for that product

Question 5

ShopNet想要更加深入的瞭解它的訂單趨勢，特別是在某些特別的季度。

ShopNet is eager to delve deeper into its order trends, especially during specific seasons.

數據庫模式

Database Schema:

`Orders_Q5` 表記錄著每一個在**ShopNet**平臺上的交易。

`Orders_Q5` table records every transaction made on the platform.

Field	Type	Description
order_id	Integer	Unique identifier for each order
customer_id	Integer	Reference to the Customers table
order_date	Date	Date of the order
total_amount	Decimal	Total cost of the order

`Order_Items_Q5` 表記錄著每一個交易裡的商品及其資訊。

`Order_Items_Q5` table captures specifics about each product in an order. Note that one order may involve multiple different products.

Field	Type	Description
order_item_id	Integer	Unique identifier for each order item
order_id	Integer	Reference to the Orders table
product_name	Varchar(255)	Name of the product
quantity	Integer	Quantity of the product ordered
unit_price	Decimal	Price per product

請編寫一個 SQL 查詢，找出滿足以下條件的訂單資訊：

- 訂單日期在 2022-09-01 至 2023-01-31 之間。
- 訂單總金額在 200 到 400 之間。
- 訂單中至少包含一項產品的銷售數量大於等於 3。

要求查詢結果包括 order_id（訂單ID）、order_date（訂單日期）和 total_amount（訂單總金額）列。

Your challenge is to filter out orders based on the following criteria:

- Orders placed between `2022-09-01` and `2023-01-31`.
- Total order amounts ranging from 200 to 400.
- Orders containing at least one product with a quantity of 3 or more.

輸出格式:

Expected Output Schema:

Field	Type	Description
order_id	Integer	Unique identifier for each order
order_date	Date	Date of the order
total_amount	Decimal	Total cost of the order

Question 6

為了致力於客戶服務，**ShopNet**希望保持對其客戶購買習慣的全面記錄。通過這樣做，**ShopNet**可以個性化用戶體驗並改進其產品供給。

With its commitment to customer service, **ShopNet** seeks to maintain a comprehensive record of its customers ’ buying habits. By doing so, ShopNet can personalize user experiences and refine its product offerings.

數據庫模式：

Database Schema:

`Customers_Q6` 表裡記錄著每一個平臺顧客的資訊。

`Customers_Q6` table offers a snapshot of all the platform's users.

Field	Type	Description
customer_id	Integer	Unique identifier for each customer
customer_name	Varchar(255)	Name of the customer
city	Varchar(255)	City of the customer

`Orders_Q6` 表裡記錄著這些顧客的交易記錄。

`Orders_Q6` table documents each purchase made by these customers.

Field	Type	Description
order_id	Integer	Unique identifier for each order
customer_id	Integer	Reference to the Customers table
order_date	Date	Date of the order
total_amount	Decimal	Total cost of the order

`Order_Items_Q6` 表裡記錄著每一個交易中的商品及其資訊。

`Order_Items_Q6` table breaks down the products contained in each order. Note that one order may involve multiple different products.

Field	Type	Description
order_item_id	Integer	Unique identifier for each order item

Field	Type	Description
order_id	Integer	Reference to the Orders table
product_name	Varchar(255)	Name of the product
quantity	Integer	Quantity of the product ordered
unit_price	Decimal	Price per product

請編寫一個 SQL 查詢，找出所有顧客及其對應的訂單和訂單項資訊。如果顧客沒有交易在冊，仍然需要包括相應的記錄，並在相關列中使用 NULL 值。

要求查詢結果包括 customer_name（顧客姓名）、order_id（訂單ID）、order_date（訂單日期）、product_name（產品名稱）、quantity（數量）和 unit_price（單價）列。

Given the details, you're tasked with extracting comprehensive information on every customer, alongside their respective orders and items. If there's an absence of data in either the customer, order, or item records, the result should still encompass the customer's name, filling in the missing data with NULL values.

輸出格式:

Expected Output Schema:

Field	Type	Description
customer_name	Varchar(255)	Name of the customer
order_id	Integer	Unique ID for order
order_date	Date	Date of the order
product_name	Varchar(255)	Name of the product
quantity	Integer	Quantity of product
unit_price	Decimal	Price of the product

Question 7

ShopNet，作為一個主要面向全球的電子商務平臺，也以其緊密的員工文化而自豪。為了促進部門間的互動與合作，人力資源部門發起了一個有趣的活動。他們想要找出共用相同電子郵件域的員工配對。

ShopNet, primarily a global e-commerce platform, also prides itself on its cohesive employee culture. In order to promote inter-departmental interactions and collaborations, the HR department has initiated a fun exercise. They are curious to find out pairs of employees who share the same email domain.

數據庫模式:

Database Schema:

`Employees_Q7` 記錄著所有**ShopNet**的員工的資訊。

`Employees_Q7` table contains data about all the employees working at **ShopNet**.

Field	Type	Description
employee_id	Integer	Unique identifier for each employee
employee_name	Varchar(255)	Name of the employee
email	Varchar(255)	Email ID of the employee

根據員工數據庫，您的目標是提取共用相同電子郵件域（即電子郵件地址中“@”後面的部分）的員工配對。

Given the database schema, your objective is to extract pairs of employees who share the same email domain (the part after the '@' in email addresses).

輸出格式:

Expected Output Schema:

Field	Type	Description
employee_name1	Varchar(255)	Name of the first employee
email1	Varchar(255)	Email ID of the first employee
employee_name2	Varchar(255)	Name of the second employee
email2	Varchar(255)	Email ID of the second employee

注意：每個配對只需返回一次。您可以限制輸出的表中員工1的id小於員工2的id來達到這個要求。

Note: Each pair should be returned only once. You can set employee 1's ID to be smaller than employee 2's ID in the output table to avoid duplicate records.