**CONTEST LUYỆN TẬP SQL SỐ 1.**

**Địa chỉ thi:** [**https://sql.toiyeuptit.com/**](https://sql.toiyeuptit.com/)**. Thời gian thực hiện: 15/12 – 28/12/2024**

**MỤC LỤC**

[BÀI 1. CẬP NHẬT LƯƠNG 1](#_Toc185842401)

[BÀI 2. KHÁCH HÀNG ĐÃ ĐẾN NHƯNG KHÔNG THỰC HIỆN GIAO DỊCH 2](#_Toc185842402)

[BÀI 3. TĂNG NHIỆT ĐỘ 3](#_Toc185842403)

[BÀI 4. SINH VIÊN VÀ NHỮNG MÓN QUÀ 4](#_Toc185842404)

[BÀI 5. QUẢN LÝ CÓ ÍT NHẤT 5 CẤP DƯỚI TRỰC TIẾP 6](#_Toc185842405)

[BÀI 6. TỶ LỆ XÁC NHẬN 7](#_Toc185842406)

[BÀI 7. PHÂN TÍCH DOANH SỐ SẢN PHẨM III 9](#_Toc185842407)

[BÀI 8. KHÁCH HÀNG MUA TẤT CẢ CÁC SẢN PHẨM 10](#_Toc185842408)

[BÀI 9. CÁC SỐ LIÊN TIẾP 11](#_Toc185842409)

[BÀI 10. NGƯỜI CUỐI CÙNG CÓ THỂ LÊN XE BUÝT 12](#_Toc185842410)

[BÀI 11. PHI VỤ ĐẦU TƯ 2016 13](#_Toc185842411)

[BÀI 12. PHÂN TÍCH DOANH SỐ SẢN PHẨM I 14](#_Toc185842412)

[BÀI 13. THAY THẾ MÃ NHÂN VIÊN BẰNG MÃ NHẬN DẠNG DUY NHẤT 16](#_Toc185842413)

[BÀI 14. INVALID TWEETS 17](#_Toc185842414)

[BÀI 15. ARTICLES VIEW I 18](#_Toc185842415)

[BÀI 16. SẢN PHẨM TÁI CHẾ VÀ ÍT BÉO 19](#_Toc185842416)

[BÀI 17. ĐẤT NƯỚC LỚN 20](#_Toc185842417)

[BÀI 18. LOẠI EMAIL TRÙNG LẶP 21](#_Toc185842418)

[BÀI 19. MỨC LƯƠNG CAO THỨ HAI 21](#_Toc185842419)

[BÀI 20. NHÂN VIÊN 22](#_Toc185842420)

# **BÀI 1. CẬP NHẬT LƯƠNG**

**Có 2 bảng: Employees và Departments.**   
Bạn cần cập nhật mức lương của các nhân viên dựa trên thông tin từ bảng Departments. Cụ thể:  
Nếu một nhân viên làm việc trong phòng "Sales", tăng mức lương của họ lên 10%.  
Nếu một nhân viên làm việc trong phòng "HR", tăng mức lương của họ lên 5%.  
Nếu nhân viên làm việc trong các phòng ban khác, không thay đổi mức lương.  
**Bảng Employees:**  
*Column Name          Type*  
emp\_id                  INT  
emp\_name                VARCHAR  
salary                  DECIMAL  
dept\_id                INT  
**Bảng Departments:**  
*Column Name          Type*  
dept\_id              INT  
dept\_name            VARCHAR  
Cột emp\_id trong bảng Employees là khóa chính.  
Cột dept\_id trong bảng Employees là khóa ngoại tham chiếu đến bảng Departments (cột dept\_id).  
**Yêu cầu:**  
Cập nhật mức lương của nhân viên dựa trên phòng ban họ làm việc.  
Chú ý chỉ viết 1 câu lệnh **UPDATE**

# **BÀI 2. KHÁCH HÀNG ĐÃ ĐẾN NHƯNG KHÔNG THỰC HIỆN GIAO DỊCH**

**Table: Visits**  
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| visit\_id | int |   
| customer\_id | int |   
+-------------+---------+   
`visit\_id` là cột chứa giá trị duy nhất cho bảng này.   
Bảng này chứa thông tin về các khách hàng đã đến thăm trung tâm mua sắm.   
**Table: Transactions**   
+----------------+---------+   
| Column Name | Type |   
+----------------+---------+   
| transaction\_id | int |   
| visit\_id | int |   
| amount | int |   
+----------------+---------+   
`transaction\_id` là cột chứa giá trị duy nhất cho bảng này.   
Bảng này chứa thông tin về các giao dịch được thực hiện trong suốt `visit\_id`.   
**Yêu cầu:** Viết lệnh để tìm các `visit\_id` của những người dùng đã đến thăm mà không thực   
hiện bất kỳ giao dịch nào và số lần họ thực hiện những chuyến thăm như vậy.   
**Sắp xếp customer\_id theo thứ tự tăng dần**  
**Example 1:**  
Input:   
Visits   
+----------+-------------+   
| visit\_id | customer\_id |   
+----------+-------------+   
| 1 | 23 |   
| 2 | 9 |   
| 4 | 30 |   
| 5 | 54 |   
| 6 | 96 |   
| 7 | 54 |   
| 8 | 54 |   
+----------+-------------+   
Transactions   
+----------------+----------+--------+   
| transaction\_id | visit\_id | amount |   
+----------------+----------+--------+   
| 2 | 5 | 310 |   
| 3 | 5 | 300 |   
| 9 | 5 | 200 |   
| 12 | 1 | 910 |   
| 13 | 2 | 970 |   
+----------------+----------+--------+   
Output:   
+-------------+----------------+   
| customer\_id | count\_no\_trans |   
+-------------+----------------+   
| 54 | 2 |   
| 30 | 1 |   
| 96 | 1 |   
+-------------+----------------+   
**Giải thích:**   
- Khách hàng có id = 23 đã đến thăm trung tâm mua sắm một lần và   
thực hiện một giao dịch trong chuyến thăm với `visit\_id = 12`.   
- Khách hàng có id = 9 đã đến thăm trung tâm mua sắm một lần   
và thực hiện một giao dịch trong chuyến thăm với `visit\_id = 13`.   
- Khách hàng có id = 30 đã đến thăm trung tâm mua sắm một lần và không thực hiện bất kỳ giao dịch nào.   
- Khách hàng có id = 54 đã đến thăm trung tâm mua sắm ba lần. Trong 2 chuyến thăm,   
họ không thực hiện bất kỳ giao dịch nào, và trong một chuyến thăm, họ đã thực hiện 3 giao dịch.   
- Khách hàng có id = 96 đã đến thăm trung tâm mua sắm một lần và không thực hiện bất kỳ giao dịch nào.   
Như vậy, người dùng với id = 30 và 96 đã đến thăm trung tâm mua sắm một lần mà không thực hiện bất kỳ giao dịch nào.   
Đồng thời, người dùng 54 đã đến thăm trung tâm mua sắm hai lần mà không thực hiện bất kỳ giao dịch nào.

# **BÀI 3. TĂNG NHIỆT ĐỘ**

**Table: Weather**   
+---------------+---------+   
| Column Name | Type |   
+---------------+---------+   
| id | int |   
| recordDate | date |   
| temperature | int |   
+---------------+---------+   
`id` là cột chứa giá trị duy nhất cho bảng này.   
Không có các dòng khác nhau với cùng một `recordDate`.   
Bảng này chứa thông tin về nhiệt độ vào một ngày nhất định.   
**Yêu cầu:** Viết một giải pháp để tìm tất cả `id` của các ngày có nhiệt độ cao hơn   
so với ngày trước đó (ngày hôm qua).   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**  
Input:   
Weather table:   
+----+------------+-------------+   
| id | recordDate | temperature |   
+----+------------+-------------+   
| 1 | 2015-01-01 | 10 |   
| 2 | 2015-01-02 | 25 |   
| 3 | 2015-01-03 | 20 |   
| 4 | 2015-01-04 | 30 |   
+----+------------+-------------+   
Output:   
+----+   
| id |   
+----+   
| 2 |   
| 4 |   
+----+   
**Giải thích:**   
- Vào ngày 2015-01-02, nhiệt độ cao hơn so với ngày hôm trước (10 < 25).   
- Vào ngày 2015-01-04, nhiệt độ cao hơn so với ngày hôm trước (20 < 30). ​​

# **BÀI 4. SINH VIÊN VÀ NHỮNG MÓN QUÀ**

**Table: Students**   
+---------------+---------+   
| Column Name | Type |   
+---------------+---------+   
| student\_id | int |   
| student\_name | varchar |   
+---------------+---------+   
`student\_id` là khóa chính (cột chứa giá trị duy nhất) cho bảng này.   
Mỗi hàng trong bảng này chứa ID và tên của một học sinh trong trường.   
**Table: Subjects**   
+--------------+---------+   
| Column Name | Type |   
+--------------+---------+   
| subject\_name | varchar |   
+--------------+---------+   
`subject\_name` là khóa chính (cột chứa giá trị duy nhất) cho bảng này.   
Mỗi hàng trong bảng này chứa tên của một môn học trong trường.   
**Table: Examinations**   
+--------------+---------+   
| Column Name | Type |   
+--------------+---------+   
| student\_id | int |   
| subject\_name | varchar |   
+--------------+---------+   
Không có khóa chính (cột chứa giá trị duy nhất) cho bảng này. Bảng có thể chứa các bản sao.   
Mỗi học sinh trong bảng `Students` tham gia vào mỗi môn học từ bảng `Subjects`.   
Mỗi hàng trong bảng này chỉ ra rằng một học sinh với ID `student\_id` tham gia kỳ thi của môn học `subject\_name`.   
Viết một giải pháp để tìm số lần mỗi học sinh tham gia kỳ thi của mỗi môn học.   
Trả về bảng kết quả theo thứ tự `student\_id` và `subject\_name`. (ORDER BY s.student\_id, sub.subject\_name;)  
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
Input:   
Students table:   
+------------+--------------+   
| student\_id | student\_name |   
+------------+--------------+   
| 1 | Alice |   
| 2 | Bob |   
| 13 | John |   
| 6 | Alex |   
+------------+--------------+

Subjects table:   
+--------------+   
| subject\_name |   
+--------------+   
| Math |   
| Physics |   
| Programming |   
+--------------+   
Examinations table:   
+------------+--------------+   
| student\_id | subject\_name |   
+------------+--------------+   
| 1 | Math |   
| 1 | Physics |   
| 1 | Programming |   
| 2 | Programming |   
| 1 | Physics |   
| 1 | Math |   
| 13 | Math |   
| 13 | Programming |   
| 13 | Physics |   
| 2 | Math |   
| 1 | Math |   
+------------+--------------+   
Output:   
+------------+--------------+--------------+----------------+   
| student\_id | student\_name | subject\_name | attended\_exams |   
+------------+--------------+--------------+----------------+   
| 1 | Alice | Math | 3 |   
| 1 | Alice | Physics | 2 |   
| 1 | Alice | Programming | 1 |   
| 2 | Bob | Math | 1 |   
| 2 | Bob | Physics | 0 |   
| 2 | Bob | Programming | 1 |   
| 6 | Alex | Math | 0 |   
| 6 | Alex | Physics | 0 |   
| 6 | Alex | Programming | 0 |   
| 13 | John | Math | 1 |   
| 13 | John | Physics | 1 |   
| 13 | John | Programming | 1 |   
+------------+--------------+--------------+----------------+   
**Giải thích:**   
- Bảng kết quả cần chứa tất cả học sinh và tất cả các môn học.   
- Alice tham gia kỳ thi Toán 3 lần, kỳ thi Vật lý 2 lần và kỳ thi Lập trình 1 lần.   
- Bob tham gia kỳ thi Toán 1 lần, kỳ thi Lập trình 1 lần và không tham gia kỳ thi Vật lý.   
- Alex không tham gia bất kỳ kỳ thi nào.   
- John tham gia kỳ thi Toán 1 lần, kỳ thi Vật lý 1 lần và kỳ thi Lập trình 1 lần. ​​

# **BÀI 5. QUẢN LÝ CÓ ÍT NHẤT 5 CẤP DƯỚI TRỰC TIẾP**

**Table: Employee**   
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| id | int |   
| name | varchar |   
| department | varchar |   
| managerId | int |   
+-------------+---------+   
`id` là khóa chính (cột chứa giá trị duy nhất) cho bảng này.   
Mỗi hàng trong bảng này chỉ ra tên của một nhân viên, phòng ban của họ và ID của người quản lý của họ.   
Nếu `managerId` là null, có nghĩa là nhân viên không có người quản lý.   
Không có nhân viên nào là quản lý của chính mình.   
Viết một giải pháp để tìm các người quản lý có ít nhất năm nhân viên trực tiếp báo cáo.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
Input:   
Employee table:   
+-----+-------+------------+-----------+   
| id | name | department | managerId |   
+-----+-------+------------+-----------+   
| 101 | John | A | null |   
| 102 | Dan | A | 101 |   
| 103 | James | A | 101 |   
| 104 | Amy | A | 101 |   
| 105 | Anne | A | 101 |   
| 106 | Ron | B | 101 |   
+-----+-------+------------+-----------+   
Output:   
+------+   
| name |   
+------+   
| John |   
+------+​​ ​​

# **BÀI 6. TỶ LỆ XÁC NHẬN**

**Table: Signups**   
+----------------+----------+   
| Column Name | Type |   
+----------------+----------+   
| user\_id | int |   
| time\_stamp | datetime |   
+----------------+----------+   
`user\_id` là cột chứa giá trị duy nhất cho bảng này.   
Mỗi hàng chứa thông tin về thời gian đăng ký của người dùng có ID `user\_id`.   
**Table: Confirmations**   
+----------------+----------+   
| Column Name | Type |   
+----------------+----------+   
| user\_id | int |   
| time\_stamp | datetime |   
| action | VARCHAR |   
+----------------+----------+   
(`user\_id`, `time\_stamp`) là khóa chính (kết hợp các cột chứa giá trị duy nhất) cho bảng này.   
`user\_id` là khóa ngoại (cột tham chiếu) tới bảng `Signups`.   
`action` là một ENUM (danh mục) có kiểu ('confirmed', 'timeout').   
Mỗi hàng trong bảng này chỉ ra rằng người dùng với ID `user\_id` đã yêu cầu một tin nhắn xác nhận   
tại `time\_stamp` và tin nhắn xác nhận đó đã được xác nhận ('confirmed') hoặc hết hạn mà không xác nhận ('timeout').   
Tỷ lệ xác nhận của một người dùng là số tin nhắn 'confirmed' chia cho tổng số tin nhắn xác nhận đã yêu cầu.   
Tỷ lệ xác nhận của một người dùng không yêu cầu tin nhắn xác nhận nào là 0.   
Làm tròn tỷ lệ xác nhận đến hai chữ số thập phân.   
Viết một giải pháp để tìm tỷ lệ xác nhận của mỗi người dùng.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
Input:   
**Signups table:**   
+---------+---------------------+   
| user\_id | time\_stamp |   
+---------+---------------------+   
| 3 | 2020-03-21 10:16:13 |   
| 7 | 2020-01-04 13:57:59 |   
| 2 | 2020-07-29 23:09:44 |   
| 6 | 2020-12-09 10:39:37 |   
+---------+---------------------+   
Confirmations table:   
+---------+---------------------+-----------+   
| user\_id | time\_stamp | action |   
+---------+---------------------+-----------+   
| 3 | 2021-01-06 03:30:46 | timeout |   
| 3 | 2021-07-14 14:00:00 | timeout |   
| 7 | 2021-06-12 11:57:29 | confirmed |   
| 7 | 2021-06-13 12:58:28 | confirmed |   
| 7 | 2021-06-14 13:59:27 | confirmed |   
| 2 | 2021-01-22 00:00:00 | confirmed |   
| 2 | 2021-02-28 23:59:59 | timeout |   
+---------+---------------------+-----------+   
Output:   
+---------+-------------------+   
| user\_id | confirmation\_rate |   
+---------+-------------------+   
| 6 | 0.00 |   
| 3 | 0.00 |   
| 7 | 1.00 |   
| 2 | 0.50 |   
+---------+-------------------+   
**Giải thích:**   
- Người dùng 6 không yêu cầu bất kỳ tin nhắn xác nhận nào. Tỷ lệ xác nhận là 0.   
- Người dùng 3 đã thực hiện 2 yêu cầu và cả hai đều hết hạn. Tỷ lệ xác nhận là 0.   
- Người dùng 7 đã thực hiện 3 yêu cầu và tất cả đều được xác nhận. Tỷ lệ xác nhận là 1.   
- Người dùng 2 đã thực hiện 2 yêu cầu, trong đó một yêu cầu được xác nhận và yêu cầu còn lại hết hạn.   
Tỷ lệ xác nhận là 1 / 2 = 0.5. ​​

# **BÀI 7. PHÂN TÍCH DOANH SỐ SẢN PHẨM III**

**Table: Sales**   
+-------------+-------+   
| Column Name | Type |   
+-------------+-------+   
| sale\_id | int |   
| product\_id | int |   
| year | int |   
| quantity | int |   
| price | int |   
+-------------+-------+   
(`sale\_id`, `year`) là khóa chính (kết hợp các cột chứa giá trị duy nhất) cho bảng này.   
`product\_id` là khóa ngoại (cột tham chiếu) đến bảng `Product`.   
Mỗi hàng trong bảng này đại diện cho một giao dịch bán hàng của sản phẩm có `product\_id`   
trong một năm nhất định.   
Lưu ý rằng giá trị là giá mỗi đơn vị sản phẩm.   
**Table: Product**   
+--------------+---------+   
| Column Name | Type |   
+--------------+---------+   
| product\_id | int |   
| product\_name | varchar |   
+--------------+---------+   
`product\_id` là khóa chính (cột chứa giá trị duy nhất) cho bảng này.   
Mỗi hàng trong bảng này đại diện cho tên sản phẩm của mỗi sản phẩm.   
Viết một giải pháp để chọn `product\_id`, `year`, `quantity` và `price` cho năm đầu tiên của mỗi sản phẩm được bán.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
**Input:**   
Sales table:   
+---------+------------+------+----------+-------+   
| sale\_id | product\_id | year | quantity | price |   
+---------+------------+------+----------+-------+   
| 1 | 100 | 2008 | 10 | 5000 |   
| 2 | 100 | 2009 | 12 | 5000 |   
| 7 | 200 | 2011 | 15 | 9000 |   
+---------+------------+------+----------+-------+   
Product table:   
+------------+--------------+   
| product\_id | product\_name |   
+------------+--------------+   
| 100 | Nokia |   
| 200 | Apple |   
| 300 | Samsung |   
+------------+--------------+   
Output:   
+------------+------------+----------+-------+   
| product\_id | first\_year | quantity | price |   
+------------+------------+----------+-------+   
| 100 | 2008 | 10 | 5000 |   
| 200 | 2011 | 15 | 9000 |   
+------------+------------+----------+-------+ ​​

# **BÀI 8. KHÁCH HÀNG MUA TẤT CẢ CÁC SẢN PHẨM**

**Table: Customer**   
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| customer\_id | int |   
| product\_key | int |   
+-------------+---------+   
Bảng này có thể chứa các hàng trùng lặp.   
`customer\_id` không phải là NULL.   
`product\_key` là khóa ngoại (cột tham chiếu) đến bảng `Product`.   
**Table: Product**   
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| product\_key | int |   
+-------------+---------+   
`product\_key` là khóa chính (cột chứa giá trị duy nhất) cho bảng này.   
Viết một giải pháp để báo cáo các `customer\_id` từ bảng `Customer` đã mua tất cả các sản phẩm trong bảng `Product`.   
Trả về bảng kết quả theo bất kỳ thứ tự nào.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
Input:   
Customer table:   
+-------------+-------------+   
| customer\_id | product\_key |   
+-------------+-------------+   
| 1 | 5 |   
| 2 | 6 |   
| 3 | 5 |   
| 3 | 6 |   
| 1 | 6 |   
+-------------+-------------+   
Product table:   
+-------------+   
| product\_key |   
+-------------+   
| 5 |   
| 6 |   
+-------------+   
Output:   
+-------------+   
| customer\_id |   
+-------------+   
| 1 |   
| 3 |   
+-------------+   
**Explanation:**   
Những khách hàng đã mua tất cả các sản phẩm (5 và 6) là khách hàng có ID là 1 và 3. ​​

# **BÀI 9. CÁC SỐ LIÊN TIẾP**

**Table: Logs**   
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| id | int |   
| num | varchar |   
+-------------+---------+   
Trong SQL, `id` là khóa chính cho bảng này.   
`id` là cột tự tăng bắt đầu từ 1.   
Tìm tất cả các số xuất hiện ít nhất ba lần liên tiếp.   
Trả về bảng kết quả theo bất kỳ thứ tự nào.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
**Input:**   
Logs table:   
+----+-----+   
| id | num |   
+----+-----+   
| 1 | 1 |   
| 2 | 1 |   
| 3 | 1 |   
| 4 | 2 |   
| 5 | 1 |   
| 6 | 2 |   
| 7 | 2 |   
+----+-----+   
Output:   
+-----------------+   
| ConsecutiveNums |   
+-----------------+   
| 1 |   
+-----------------+   
**Giải thích:** 1 là số duy nhất xuất hiện liên tiếp ít nhất ba lần. ​​

# **BÀI 10. NGƯỜI CUỐI CÙNG CÓ THỂ LÊN XE BUÝT**

**Table: Queue**   
+-------------+---------+   
| Column Name | Type |   
+-------------+---------+   
| person\_id | int |   
| person\_name | varchar |   
| weight | int |   
| turn | int |   
+-------------+---------+   
Cột person\_id chứa các giá trị duy nhất.   
Bảng này có thông tin về tất cả những người đang chờ xe buýt.   
Cột person\_id và turn sẽ chứa tất cả các số từ 1 đến n, trong đó n là số hàng trong bảng.   
Cột turn xác định thứ tự mà người sẽ lên xe buýt, trong đó turn=1 là người đầu tiên lên xe   
và turn=n là người cuối cùng lên xe.   
Cột weight là trọng lượng của người đó tính bằng kilogram.   
Có một hàng người đang chờ lên xe buýt. Tuy nhiên, xe buýt có giới hạn trọng lượng là 1000   
kilogram, vì vậy có thể có một số người không thể lên xe.   
Viết một giải pháp để tìm tên của người cuối cùng có thể lên xe mà không vượt quá giới hạn trọng lượng.   
Các bài kiểm tra được tạo ra sao cho người đầu tiên không vượt quá giới hạn trọng lượng.   
Lưu ý rằng chỉ có một người có thể lên xe tại mỗi lượt.   
Định dạng kết quả giống ví dụ sau.   
**Example 1:**   
**Input:**   
Queue table:   
+-----------+-------------+--------+------+   
| person\_id | person\_name | weight | turn |   
+-----------+-------------+--------+------+   
| 5 | Alice | 250 | 1 |   
| 4 | Bob | 175 | 5 |   
| 3 | Alex | 350 | 2 |   
| 6 | John Cena | 400 | 3 |   
| 1 | Winston | 500 | 6 |   
| 2 | Marie | 200 | 4 |   
+-----------+-------------+--------+------+   
Output:   
+-------------+   
| person\_name |   
+-------------+   
| John Cena |   
+-------------+   
**Giải thích:** Bảng sau được sắp xếp theo thứ tự turn để dễ hiểu.   
+------+----+-----------+--------+--------------+   
| Turn | ID | Name | Weight | Total Weight |   
+------+----+-----------+--------+--------------+   
| 1 | 5 | Alice | 250 | 250 |   
| 2 | 3 | Alex | 350 | 600 |   
| 3 | 6 | John Cena | 400 | 1000 | (last person to board)   
| 4 | 2 | Marie | 200 | 1200 | (cannot board)   
| 5 | 4 | Bob | 175 | \_\_\_ |   
| 6 | 1 | Winston | 500 | \_\_\_ |   
+------+----+-----------+--------+--------------+ ​​

# **BÀI 11. PHI VỤ ĐẦU TƯ 2016**

**Table: Insurance**   
+-------------+-------+   
| Column Name | Type |   
+-------------+-------+   
| pid | int |   
| tiv\_2015 | float |   
| tiv\_2016 | float |   
| lat | float |   
| lon | float |   
+-------------+-------+

**Ràng buộc:**  
​- pid là khóa chính (cột có giá trị duy nhất) cho bảng này.   
- Mỗi hàng trong bảng này chứa thông tin về một hợp đồng bảo hiểm, với các cột sau:   
 + pid là ID của hợp đồng bảo hiểm của người mua bảo hiểm.   
 + tiv\_2015 là tổng giá trị đầu tư trong năm 2015

+ tiv\_2016 là tổng giá trị đầu tư trong năm 2016.   
 + lat là vĩ độ của thành phố nơi người mua bảo hiểm cư trú. Đảm bảo lat không là NULL.   
​ + lon là kinh độ của thành phố nơi người mua bảo hiểm cư trú. Đảm bảo lon không là NULL.   
**Yêu cầu:**

Viết một câu truy vấn để báo cáo tổng giá trị đầu tư trong năm 2016 (tiv\_2016)   
cho tất cả các người mua bảo hiểm, những người không nằm trong cùng thành phố với bất kỳ chủ hợp đồng bảo hiểm nào khác   
(tức là các cặp thuộc tính (lat, lon) phải là duy nhất).

**Chú ý:**  
Giá trị tiv\_2015 giống như một hoặc nhiều người mua bảo hiểm khác.   
Không cùng thành phố với bất kỳ người mua bảo hiểm nào khác (tức là các cặp thuộc tính (lat, lon) phải là duy nhất).   
​Làm tròn giá trị tiv\_2016 đến hai chữ số thập phân.​   
Kết quả trả về theo định dạng như trong ví dụ dưới đây.   
**Example 1:**   
**Input:**   
Insurance table:   
+-----+----------+----------+-----+-----+   
| pid | tiv\_2015 | tiv\_2016 | lat | lon |   
+-----+----------+----------+-----+-----+   
| 1 | 10 | 5 | 10 | 10 |   
| 2 | 20 | 20 | 20 | 20 |   
| 3 | 10 | 30 | 20 | 20 |   
| 4 | 10 | 40 | 40 | 40 |   
+-----+----------+----------+-----+-----+

**Output:**+----------+   
| tiv\_2016 |   
+----------+   
| 45.00 |   
+----------+   
**Giải thích:**   
Giá trị tiv\_2015 = 10 bằng với bản ghi số 3, 4, và location của nó là unique   
Bản ghi thứ 2 không thỏa mãn yêu cầu.   
Tiv\_2015 của nó không giống bất kỳ chủ hợp đồng nào khác và vị trí của nó giống với bản ghi thứ ba  
, điều này khiến bản ghi thứ ba cũng bị lỗi  
Vậy kết quả là tổng tiv\_2016 của bản ghi đầu tiên và cuối cùng là 45  
Chú ý: Sử dụng ROUND(SUM(tiv\_2016), 2) ​​

# **BÀI 12. PHÂN TÍCH DOANH SỐ SẢN PHẨM I**

**Table: Sales**  
+-------------+-------+  
| Column Name | Type |  
+-------------+-------+  
| sale\_id | int |  
| product\_id | int |  
| year | int |  
| quantity | int |  
| price | int |  
+-------------+-------+  
(sale\_id, year) là khóa chính (kết hợp các cột chứa giá trị duy nhất) của bảng này.  
`product\_id` là một khóa ngoại (cột tham chiếu) tới bảng `Product`.  
Mỗi hàng trong bảng này thể hiện một giao dịch bán sản phẩm (`product\_id`) trong một năm nhất định.  
Lưu ý rằng `price` là giá trên mỗi đơn vị.  
**Table: Product**  
+--------------+---------+  
| Column Name | Type |  
+--------------+---------+  
| product\_id | int |  
| product\_name | varchar |  
+--------------+---------+  
`product\_id` là khóa chính (cột chứa giá trị duy nhất) của bảng này.  
Mỗi hàng trong bảng này thể hiện tên sản phẩm (`product\_name`) của từng sản phẩm.

**Yêu cầu:**  
Viết một giải pháp để báo cáo `product\_name`, `year`, và `price` cho mỗi `sale\_id` trong bảng `Sales`.  
Sắp xếp theo sale\_id tăng dần  
Định dạng kết quả giống ví dụ sau.  
**Example 1:**  
Input:   
Sales table:  
+---------+------------+------+----------+-------+  
| sale\_id | product\_id | year | quantity | price |  
+---------+------------+------+----------+-------+   
| 1 | 100 | 2008 | 10 | 5000 |  
| 2 | 100 | 2009 | 12 | 5000 |  
| 7 | 200 | 2011 | 15 | 9000 |  
+---------+------------+------+----------+-------+  
Product table:  
+------------+--------------+  
| product\_id | product\_name |  
+------------+--------------+  
| 100 | Nokia |  
| 200 | Apple |  
| 300 | Samsung |  
+------------+--------------+  
Output:   
+--------------+-------+-------+  
| product\_name | year | price |  
+--------------+-------+-------+  
| Nokia | 2008 | 5000 |  
| Nokia | 2009 | 5000 |  
| Apple | 2011 | 9000 |  
+--------------+-------+-------+  
**Giải thích:**  
- Từ `sale\_id = 1`, ta có thể kết luận rằng Nokia đã được bán với giá 5000 trong năm 2008.  
- Từ `sale\_id = 2`, ta có thể kết luận rằng Nokia đã được bán với giá 5000 trong năm 2009.  
- Từ `sale\_id = 7`, ta có thể kết luận rằng Apple đã được bán với giá 9000 trong năm 2011.

# **BÀI 13. THAY THẾ MÃ NHÂN VIÊN BẰNG MÃ NHẬN DẠNG DUY NHẤT**

**Table: Employees**  
+---------------+---------+  
| Column Name | Type |  
+---------------+---------+  
| id | int |  
| name | varchar |  
+---------------+---------+  
id là khóa chính (cột chứa giá trị duy nhất) cho bảng này.  
Mỗi hàng trong bảng này chứa `id` và tên (`name`) của một nhân viên trong công ty.  
**Table: EmployeeUNI**  
+---------------+---------+  
| Column Name | Type |  
+---------------+---------+  
| id | int |  
| unique\_id | int |  
+---------------+---------+  
(id, unique\_id) là khóa chính (kết hợp các cột chứa giá trị duy nhất) cho bảng này.  
Mỗi hàng trong bảng này chứa `id` và `unique\_id` tương ứng của một nhân viên trong công ty.  
Viết một giải pháp để hiển thị `unique\_id` của mỗi người dùng.   
Nếu một người dùng không có `unique\_id`, thay vào đó hiển thị giá trị `null`.  
Định dạng kết quả giống ví dụ sau.  
**Example 1:**  
**Input:**   
Employees table:  
+----+----------+  
| id | name |  
+----+----------+  
| 1 | Alice |  
| 7 | Bob |  
| 11 | Meir |  
| 90 | Winston |  
| 3 | Jonathan |  
+----+----------+  
EmployeeUNI table:  
+----+-----------+  
| id | unique\_id |  
+----+-----------+  
| 3 | 1 |  
| 11 | 2 |  
| 90 | 3 |  
+----+-----------+

**Output:**   
+-----------+----------+  
| unique\_id | name |  
+-----------+----------+  
| null | Alice |  
| null | Bob |  
| 2 | Meir |  
| 3 | Winston |  
| 1 | Jonathan |  
+-----------+----------+  
**Giải thích:**  
- Alice và Bob không có `unique\_id`, vì vậy sẽ hiển thị `null` thay thế.  
- `unique\_id` của Meir là 2.  
- `unique\_id` của Winston là 3.  
- `unique\_id` của Jonathan là 1.

# **BÀI 14. INVALID TWEETS**

**Table: Tweets**  
+----------------+---------+  
| Column Name | Type |  
+----------------+---------+  
| tweet\_id | int |  
| content | varchar |  
+----------------+---------+  
tweet\_id là khóa chính (cột chứa giá trị duy nhất) cho bảng này.  
`content` bao gồm các ký tự trên bàn phím tiếng Anh Mỹ và không chứa các ký tự đặc biệt khác.  
Bảng này chứa tất cả các tweet trong một ứng dụng mạng xã hội.

Một tweet được coi là không hợp lệ nếu số ký tự trong nội dung (`content`) của tweet lớn hơn 15.

**Yêu cầu:**  
Viết một giải pháp để tìm các `tweet\_id` của những tweet **không** hợp lệ.   
Định dạng kết quả giống ví dụ sau.  
**Example 1:**  
**Input:**   
Tweets table:  
+----------+-----------------------------------+  
| tweet\_id | content |  
+----------+-----------------------------------+  
| 1 | Let us Code |  
| 2 | More than fifteen chars are here! |  
+----------+-----------------------------------+

**Output:**   
+----------+  
| tweet\_id |  
+----------+  
| 2 |  
+----------+  
**Giải thích:**   
Tweet 1 có độ dài = 11. Đây là 1 tweet hợp lệ.  
Tweet 2 có độ dài = 33. Đây là 1 tweet không hợp lệ.​​

# **BÀI 15. ARTICLES VIEW I**

**Bảng: Views**  
+---------------+---------+  
| Tên cột | Type |  
+---------------+---------+  
| article\_id | int |  
| author\_id | int |  
| viewer\_id | int |  
| view\_date | date |  
+---------------+---------+  
Bảng này không có khóa chính. Bảng có thể chứa các dòng trùng lặp.  
Mỗi dòng trong bảng này cho biết một người xem đã xem một bài viết của tác giả nào đó vào một ngày cụ thể.  
Lưu ý rằng nếu `author\_id` và `viewer\_id` giống nhau, điều đó có nghĩa là tác giả và người xem là cùng một người.  
**Yêu cầu**:  
Viết một câu lệnh để tìm tất cả các tác giả đã xem ít nhất một trong những bài viết của chính họ.  
Trả về bảng kết quả được sắp xếp theo `id` theo thứ tự tăng dần.  
Định dạng kết quả như ví dụ sau:  
**Ví dụ 1:**  
Dữ liệu đầu vào:  
Bảng Views:  
+------------+-----------+-----------+------------+  
| article\_id | author\_id | viewer\_id | view\_date |  
+------------+-----------+-----------+------------+  
| 1 | 3 | 5 | 2019-08-01 |  
| 1 | 3 | 6 | 2019-08-02 |  
| 2 | 7 | 7 | 2019-08-01 |  
| 2 | 7 | 6 | 2019-08-02 |  
| 4 | 7 | 1 | 2019-07-22 |  
| 3 | 4 | 4 | 2019-07-21 |  
| 3 | 4 | 4 | 2019-07-21 |  
+------------+-----------+-----------+------------+

Dữ liệu đầu ra:  
+------+  
| id |  
+------+  
| 4 |  
| 7 |  
+------+

# **BÀI 16. SẢN PHẨM TÁI CHẾ VÀ ÍT BÉO**

**Bảng: Products**  
Column Name       Type  
product\_id              INT  
low\_fats                 VARCHAR  
recyclable              VARCHAR  
product\_id là khóa chính (cột chứa giá trị duy nhất) cho bảng này.  
low\_fats là một VARCHARcó kiểu ('Y', 'N') trong đó 'Y' có nghĩa là sản phẩm này ít chất béo và 'N' có nghĩa là sản phẩm này không ít chất béo.  
recyclable là một VARCHAR có kiểu ('Y', 'N') trong đó 'Y' có nghĩa là sản phẩm này có thể tái chế và 'N' có nghĩa là sản phẩm này không thể tái chế.  
**Yêu cầu:**  
Viết một truy vấn SQL để tìm các product\_id của các sản phẩm vừa ít chất béo (low fat) và có thể tái chế (recyclable).  
Trả về kết quả dưới dạng bảng như ví dụ sau.  
**Ví dụ 1:**  
Dữ liệu đầu vào:  
Bảng Products:  
product\_id  low\_fats recyclable  
0                 Y             N  
1                 Y             Y  
2                 N             Y  
3                 Y             Y  
4                 N             N  
Kết quả đầu ra:  
product\_id  
1  
3  
**Giải thích:**  
Chỉ có sản phẩm có product\_id là 1 và 3 là cả hai: ít chất béo (low fat) và có thể tái chế (recyclable).

# **BÀI 17. ĐẤT NƯỚC LỚN**

**Table World**  
+-------------+---------+  
| Column Name | Type |  
+-------------+---------+  
| name | varchar |  
| continent | varchar |  
| area | int |  
| population | int |  
| gdp | bigint |  
+-------------+---------+  
name là khóa chính (cột chứa giá trị duy nhất) cho bảng này.  
Mỗi hàng trong bảng này cung cấp thông tin về tên quốc gia, châu lục mà quốc gia đó thuộc về,   
diện tích, dân số và giá trị GDP của quốc gia.  
Một quốc gia được coi là lớn nếu:  
- Quốc gia có diện tích ít nhất là ba triệu (tức là 3000000 km²), hoặc  
- Quốc gia có dân số ít nhất là hai mươi lăm triệu (tức là 25000000).  
Viết một giải pháp để tìm tên (`name`), dân số (`population`) và diện tích (`area`) của các quốc gia lớn.  
Định dạng kết quả giống ví dụ sau.  
**Example 1:**  
**Input:**   
World table:  
+-------------+-----------+---------+------------+--------------+  
| name | continent | area | population | gdp |  
+-------------+-----------+---------+------------+--------------+  
| Afghanistan | Asia | 652230 | 25500100 | 20343000000 |  
| Albania | Europe | 28748 | 2831741 | 12960000000 |  
| Algeria | Africa | 2381741 | 37100000 | 188681000000 |  
| Andorra | Europe | 468 | 78115 | 3712000000 |  
| Angola | Africa | 1246700 | 20609294 | 100990000000 |  
+-------------+-----------+---------+------------+--------------+  
Output:   
+-------------+------------+---------+  
| name | population | area |  
+-------------+------------+---------+  
| Afghanistan | 25500100 | 652230 |  
| Algeria | 37100000 | 2381741 |  
+-------------+------------+---------+​​

# **BÀI 18. LOẠI EMAIL TRÙNG LẶP**

**Table: Person**  
+-------------+---------+  
| Column Name | Type |  
+-------------+---------+  
| id | int |  
| email | varchar |  
+-------------+---------+  
id là primary key (cột với các giá trị duy nhất) của bảng này.  
Mỗi hàng trong bảng này chứa một email. Các email sẽ không chứa chữ cái viết hoa.  
Hãy liệt kê tất cả các email trùng lặp (duplicate emails).  
Lưu ý rằng trường email được đảm bảo là không NULL.  
**Example 1:**  
Input:   
Person table:  
+----+---------+  
| id | email |  
+----+---------+  
| 1 | a@b.com |  
| 2 | c@d.com |  
| 3 | a@b.com |  
+----+---------+  
Output:   
+---------+  
| Email |  
+---------+  
| a@b.com |  
+---------+  
**Giải thích:** a@b.com được lặp lại 2 lần.​​

# **BÀI 19. MỨC LƯƠNG CAO THỨ HAI**

**Table: Employee**  
+-------------+------+  
| Column Name | Type |  
+-------------+------+  
| id | int |  
| salary | int |  
+-------------+------+  
id là khóa chính ( cột với giá trị duy nhất ) cho bảng này.  
Mỗi hàng của bảng chứa thông tin về lương của mỗi nhân viên  
Hãy tìm mức lương phân biệt lớn thứ hai từ bảng Employee.  
Kết quả được định dạng theo mẫu sau :  
**Example 1:  
Input:**   
Employee table:  
+----+--------+  
| id | salary |  
+----+--------+  
| 1 | 100 |  
+----+--------+  
**Output:**   
+---------------------+  
| SecondHighestSalary |  
+---------------------+  
| null |  
+---------------------+​​​​​​​​

# **BÀI 20. NHÂN VIÊN**

Cho bảng **employees** gồm các thông tin  
ID INT PRIMARY KEY  
salary FLOAT  
department\_id INT  
Hãy cập nhật lương (salary) tăng 10% của các nhân viên có department\_id là 2

**Gợi ý**

UPDATE employees  
SET salary = salary \* 1.1  
WHERE department\_id = 2;