**BÀI LÀM 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**

UPDATE Employees e JOIN Departments d ON e.dept\_id = d.dept\_id

SET e.salary = CASE

WHEN d.dept\_name = 'Sales' THEN e.salary \* 1.10

WHEN d.dept\_name = 'HR' THEN e.salary \* 1.05

ELSE e.salary

END;

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

SELECT v.customer\_id, COUNT(v.visit\_id) AS count\_no\_trans

FROM Visits v

LEFT JOIN Transactions t ON v.visit\_id = t.visit\_id

WHERE t.transaction\_id IS NULL

GROUP BY v.customer\_id

ORDER BY v.customer\_id;

# **BÀI 3. TĂNG NHIỆT ĐỘ**

SELECT w1.id

FROM Weather w1

JOIN Weather w2 ON w1.recordDate = DATE\_ADD(w2.recordDate, INTERVAL 1 DAY)

WHERE w1.temperature > w2.temperature;

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

SELECT s.student\_id, s.student\_name, sub.subject\_name,

COUNT(e.subject\_name) AS attended\_exams

FROM Students s

CROSS JOIN Subjects sub

LEFT JOIN Examinations e ON s.student\_id = e.student\_id AND sub.subject\_name = e.subject\_name

GROUP BY s.student\_id, s.student\_name, sub.subject\_name

ORDER BY s.student\_id, sub.subject\_name;

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

SELECT e.name

FROM Employee e

WHERE e.id IN (

SELECT managerId

FROM Employee

WHERE managerId IS NOT NULL

GROUP BY managerId

HAVING COUNT(\*) >= 5

);  
|​​

# **BÀI 6. TỶ LỆ XÁC NHẬN**

SELECT s.user\_id,

ROUND(COALESCE(SUM(CASE WHEN c.action = 'confirmed' THEN 1 ELSE 0 END), 0) /

COUNT(c.action), 2) AS confirmation\_rate

FROM Signups s

LEFT JOIN Confirmations c ON s.user\_id = c.user\_id

GROUP BY s.user\_id

ORDER BY s.user\_id;

# **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**

SELECT customer\_id

FROM Customer

GROUP BY customer\_id

HAVING COUNT(DISTINCT product\_key) = (SELECT COUNT(\*) FROM Product);

# **BÀI 9. CÁC SỐ LIÊN TIẾP**

SELECT DISTINCT num AS ConsecutiveNums

FROM (

SELECT num,

LAG(num, 1) OVER (ORDER BY id) AS prev\_num,

LEAD(num, 1) OVER (ORDER BY id) AS next\_num

FROM Logs

) AS subquery

WHERE num = prev\_num AND num = next\_num;

# **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**

SELECT p.product\_name, s.year, s.price

FROM Sales s

JOIN Product p ON s.product\_id = p.product\_id

ORDER BY s.sale\_id;

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

SELECT eu.unique\_id, e.name

FROM Employees e

LEFT JOIN EmployeeUNI eu ON e.id = eu.id;

# **BÀI 14. INVALID TWEETS**

SELECT tweet\_id FROM Tweets WHERE LENGTH(content) > 15;

# **BÀI 15. ARTICLES VIEW I**

SELECT DISTINCT author\_id AS id

FROM Views

WHERE author\_id = viewer\_id

ORDER BY id;

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

SELECT product\_id FROM Products WHERE low\_fats = 'Y' AND recyclable = 'Y';

# **BÀI 17. ĐẤT NƯỚC LỚN**

SELECT name, population, area FROM World

WHERE area >= 3000000 OR population >= 25000000;

# **BÀI 18. LOẠI EMAIL TRÙNG LẶP**

SELECT email as Email

FROM Person

GROUP BY email

HAVING COUNT(email) > 1;

# **BÀI 19. MỨC LƯƠNG CAO THỨ HAI**

SELECT salary AS SecondHighestSalary

FROM (SELECT DISTINCT salary FROM Employee ORDER BY salary DESC LIMIT 2) AS temp

ORDER BY salary ASC

LIMIT 1;

# **BÀI 20. NHÂN VIÊN**

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