Chapter 4. The final exercises of this team

I. Bài tập lý thuyết:

1. If we have a column, X, with values 1, 2, 3, and NULL, what is AVG(X)?
   AVG(X) = 6/3 = 2
2. Does AVG(price) = SUM(price)/COUNT(*)? Explain. If they are not equal, how can you fix it?
   Chúng không bằng nhau vì lệnh AVG và SUM bỏ qua bản ghi có giá trị thuộc tính price NULL, còn COUNT(*) đếm mọi bản ghi bao gồm cả bản ghi có giá trị thuộc tính price là NULL.
   Sửa: AVG(price) = SUM(price)/COUNT()
3. If we have a column X, when can MIN(X) = MAX(X)?
   TH1. Mọi giá trị cột X bằng nhau và khác NULL
   TH2. Mọi giá trị cột X là NULL???
4. If X is a primary key, what is COUNT(X)? How does that value compare to COUNT(*) over the same table?
   Lúc này COUNT(X) = COUNT(*) vì mọi bản ghi có giá trị thuộc tính X là độc nhất nên số bản ghi có X = NULL là 0, tức COUNT(X) = số bản ghi có trong bảng
5. For each aggregate function, what is the result if the SQL statement includes WHERE 1 = 2?
   Result sẽ là 0 dòng.
6. If the result of the query is the same, which is better, eliminating rows with a WHERE clause or eliminating groups with a HAVING clause? Why?
   Nên loại bỏ việc gom nhóm và các điều kiện của mệnh đề HAVING. Vì WHERE sẽ thực hiện kiểm tra điều kiện với n bản ghi trước. Trong khi việc gom nhóm sẽ duyệt qua n bản ghi để gom nhóm. Sau đó lại duyệt m (m <=n) bản ghi để kiểm tra điều kiện của mệnh đề HAVING. Như vậy trong trường hợp kết quả ra như nhau thì việc sử dụng 1 mệnh đề WHERE có hiệu năng hơn việc sử dụng gom nhóm với 1 mệnh đề HAVING.
7. Is there any relationship between the number of rows in a table and the number of rows generated by a GROUP BY?
   Vì GROUP BY có làm việc với các bản ghi có giá trị thuộc tính đang sắp xếp là NULL nên số lượng bản ghi của bảng sẽ bằng số bản ghi được sắp xếp bởi GROUP BY (trong trường hợp mệnh đề WHERE trước đó - nếu có - không loại bỏ đi 1 bản ghi nào sẽ dùng để sắp xếp bởi GROUP BY).
8. What is the difference between using DISTINCT in the SELECT clause and a GROUP BY clause with the same attributes?
   DISTINCT và GROUP BY thường sẽ cho kết quả giống nhau. Tuy nhiên, mỗi cái có cách sử dụng riêng của nó. Ta dùng GROUP BY khi ta áp dụng các phép toán tập hợp, còn nếu mục đích chỉ là loại các dòng trùng lặp ta dùng DISTINCT.

9. What can the HAVING clause contain that the WHERE clause cannot?
   HAVING có thể chứa các hàm tập hợp còn WHERE thì không
10. How many rows are returned if the GROUP BY clause contains the primary key of a table?
    Bằng số hàng của bảng
11. WHERE is to rows as HAVING is to **Groups**
12. Using the Restaurant Database, assume the FROM clause contains the vendors table. If the GROUP BY clause contains the vendorid and replname attributes, list all attributes that can appear by themselves in the SELECT, WHERE, HAVING, and ORDER BY clauses.
    SELECT chỉ chứa vendorid, replname
    WHERE, HAVING, ORDER BY có thể chứa mọi thuộc tính của bảng
13. If a HAVING clause appears without a GROUP BY clause, what is the maximum number of rows that can be in the result? What is the minimum number of rows?
    Nếu HAVING không chứa hàm tập hợp thì số hàng nhiều nhất có thể là số hàng của bảng
    Nếu HAVING chứa hàm tập hợp thì số hàng nhiều nhất có thể là 1
    Số hàng ít nhất là 0
14. In statistics, the “sum-of-squares error” is used to determine the goodness of fit for an approximation. If X and Y are columns in table test , write an SQL statement to find the “sum-ofsquares error” for the data in test.
    SELECT SUM(POWER(X - Y, 2)) AS “sse”
    FROM test

II. Bài tập thực hành:

1. Find the average salary for all employees.

   SELECT AVG(salary) from employees

2. Find the minimum and maximum project revenue for all active projects that make money.

   SELECT min(revenue), max(revenue) from projects

3. Find the number of projects that are completed. You may not use a WHERE clause.

   SELECT count(*) from projects having enddate is not null

4. Find the number of projects that have been worked on or currently are being worked on by an employee.

   SELECT employeeid, COUNT(projectid)

   FROM workson

   GROUP BY employeeid

5. Find the last name of the employee whose last name is last in dictionary order.

   SELECT lastname from employees order by lastname desc limit 1

6. Compute the employee salary standard deviation.

   SELECT STDDEV(salary) AS ‘sd’

   FROM employees

7. Find the number of employees who are assigned to some department. You may not use a WHERE clause.

   SELECT count(*) from employees having deptcode is not null

8. For each department, list the department code and the number of employees in the department.

   SELECT departments.code, COUNT(*) AS sonhanvien

   FROM departments, employees

   WHERE departments.code = employees.deptcode

   GROUP BY departments.code

9. For each department that has a project, list the department code and report the average revenue and count of all of its projects.

   SELECT departments.code, AVG(projects.revenue) AS doanhthutb, COUNT(*) AS soduan

   FROM departments, projects

   WHERE departments.code = projects.deptcode

   GROUP BY departments.code

10. Modify the query from Problem 9 to only include departments with 2 or more projects.

    SELECT departments.code, AVG(projects.revenue) AS doanhthutb, COUNT(*) AS soduan

    FROM departments, projects

    WHERE departments.code = projects.deptcode

    GROUP BY departments.code

    HAVING COUNT(*) >= 2

11. Modify the query from Problem 10 to only count active projects. Sort the results in descending order by count.

    SELECT departments.code, AVG(projects.revenue) AS doanhthutb, COUNT(*) AS soduan

    FROM departments, projects

    WHERE departments.code = projects.deptcode AND projects.enddate IS NULL

    GROUP BY departments.code

    HAVING COUNT(*) >= 2

    ORDER BY COUNT(*) DESC

12. Find the employee ID of all employees where their assigned time to work on projects is 100% or more.

    SELECT employees.employeeid

    FROM employees, workson

    WHERE employees.employeeid = workson.employeeid

        AND workson.assignedtime >= 1

13. Calculate the salary cost for each department with employees that don't have a last name ending in "re" after giving everyone a 10% raise.

    SELECT departments.code, SUM(employees.salary * 1.1) AS salary_cost

    FROM employees, departments

    WHERE employees.deptcode = departments.code

                AND employees.lastname NOT LIKE '%re'

    GROUP BY departments.code