Bangladesh University of Business and Technology
Department of Computer Science and Engineering

CSE 232: Database Systems Lab

Lab 05 Tasks - SQL on Multiple Relations (Cartesian Product, JOIN)

Cartesian Product

In case you ever want to display the content of multiple tables at once, SQL allows you to do that as well. This is known as cartesian product between tables.

```sql
SELECT * FROM
table1,table2...
```

NATURAL JOIN

Unlike the Cartesian product of two relations, which concatenates each tuple of the first relation with every tuple of the second, natural join considers only those pairs of tuples with the same value on those attributes that appear in the schemas of both relations.
A natural join operation between two relations *Instructor(i_id, name, salary)* and *Teaches(course_code, section, i_id)* would look something like this:

```sql
SELECT name, course_code
FROM Instructor, Teaches
WHERE Instructor.i_id = Teaches.i_id;

--Equivalent to--
SELECT name, course_code
FROM Instructor NATURAL JOIN Teaches;
```

You can also use 'INNER JOIN'. But in that case, you would need to specify the attributes on which you want to perform the join operation using 'ON' clause-

```sql
SELECT name, course_code
FROM Instructor
INNER JOIN teaches
ON Instructor.i_id = Teaches.i_id;
```

LEFT OUTER JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). If there is no match, the result is 0 records from the right side.

```sql
SELECT name, course_code
FROM Instructor
LEFT JOIN teaches
ON Instructor.i_id = Teaches.i_id;
```

RIGHT OUTER JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). If there is no match, the result is 0 records from the left side.

```sql
SELECT name, course_code
FROM Instructor
RIGHT JOIN teaches
ON Instructor.i_id = Teaches.i_id;
```

Tasks

Run the codes in Lab5.txt [Assume each order purchase is of 1 unit of the product]

1. Apply Cartesian Product on customers and orders.
2. Apply the different join operations on customers and orders.
3. Join all the tables using natural join.
4. How many different products are there?
5. Sort the products based on their price.
6. What is the price of the most expensive product?
7. How many units of the most available product are there?
8. In total, how many units of products are available?
9. What are the most expensive products?
10. How many items of each of the most expensive products are available?
11. Which product(s) did 'Ayman' order?
12. Among the products that 'Wendy' ordered, what is the price of the most expensive product?
13. How much money does 'Charrlie' has to pay?
14. Show a list of the amount each customer will have to pay.
15. After the purchase of the 1st customer, how many product items will be left?
16. (Bonus) Among the products that the 9th customer ordered, which is the most expensive?