ISLAMIC UNIVERSITY OF TECHNOLOGY

Organization of Islamic Cooperation

Board Bazar, Gazipur

Assignment 2

CSE 4308

Mohammad Anas Jawad

## Part 1: SQL Queries

1. What is the number of customers whose name starts with 'S'?

Query:

SELECT COUNT(cust\_name) FROM customer WHERE cust\_name LIKE 'S%';

SQL

Result:

COUNT(CUST\_NAME)  
----------------  
 6

1. What is the name of the agent who is assigned to the customer with cust\_id=C00013?

Query:

SELECT agent\_name FROM customer INNER JOIN agents ON customer.agent\_code = agents.agent\_code WHERE cust\_code = 'C00013';

SQL

Result:

AGENT\_NAME  
----------------------------------------  
Alex

1. Which customer has the highest opening amount?

Query:

SELECT cust\_name FROM customer WHERE opening\_amt = (SELECT MAX(opening\_amt) FROM customer);

SQL

Result:

CUST\_NAME  
----------------------------------------  
Shilton

1. Show the names of the customers and their corresponding agents in such a way that customers with the same agents are displayed one after another.

Query:

SELECT agent\_name, cust\_name FROM customer INNER JOIN agents ON customer.agent\_code = agents.agent\_code ORDER BY agent\_name;

SQL

Result:

AGENT\_NAME  
----------------------------------------  
CUST\_NAME  
----------------------------------------  
Alex  
Holmes

Alex  
Stuart

Alford  
Albert

Alford  
Micheal

Alford  
Bolt

Anderson  
Fleming

Anderson  
Winston

Anderson  
Jacks

Benjamin  
Charles

Ivan  
Shilton

Ivan  
Martin

Ivan  
Karolina

Lucida  
Steven

McDen  
Cook

McDen  
Karl

Mukesh  
Avinash

Mukesh  
Ramesh

Mukesh  
Sasikant

Ramasundar  
Venkatpati

Ramasundar  
Srinivas

Ravi Kumar  
Ravindran

Santakumar  
Sundariya

Santakumar  
Yearannaidu

Santakumar  
Ramanathan

Subbarao  
Rangarappa

1. Which customers have Santakumar as their agent?

Query:

SELECT cust\_name FROM customer INNER JOIN agents ON customer.agent\_code = agents.agent\_code WHERE agent\_name = 'Santakumar';

SQL

Result:

CUST\_NAME  
----------------------------------------  
Sundariya  
Yearannaidu  
Ramanathan

1. What is the number of customers each agent serves?

Query:

SELECT agent\_name, COUNT(cust\_name) FROM customer INNER JOIN agents ON customer.agent\_code = agents.agent\_code GROUP BY agent\_name;

SQL

Result:

AGENT\_NAME COUNT(CUST\_NAME)  
---------------------------------------- ----------------  
Ramasundar 2  
Alford 3  
Ravi Kumar 1  
Subbarao 1  
Benjamin 1  
Mukesh 3  
Lucida 1  
Alex 2  
McDen 2  
Santakumar 3  
Ivan 3  
Anderson 3

1. Which customer of London has the lowest outstanding amount?

Query:

SELECT cust\_name FROM customer WHERE cust\_city = 'London' AND outstanding\_amt = (SELECT MIN(outstanding\_amt) FROM customer WHERE cust\_city = 'London');

SQL

Result:

CUST\_NAME  
----------------------------------------  
Karl

1. Which customer names start with 'A'?

Query:

SELECT cust\_name FROM customer WHERE cust\_name LIKE 'A%';

SQL

Result:

CUST\_NAME  
----------------------------------------  
Albert  
Avinash

1. How many citizens work in Chennai?

Query:

SELECT SUM (citizens) FROM (SELECT COUNT(working\_area) AS citizens FROM customer WHERE working\_area = 'Chennai'

UNION

SELECT COUNT(working\_area) AS citizens FROM agents WHERE working\_area = 'Chennai');

SQL

Result:

SUM(CITIZENS)  
-------------  
 4

1. What is the number of customers residing in London?

Query:

SELECT COUNT(cust\_city) FROM customer WHERE cust\_city = 'London';

SQL

Result:

COUNT(CUST\_CITY)  
----------------  
 4

1. Display the number of customers in each city.

Query:

SELECT cust\_city, COUNT(cust\_city) FROM customer GROUP BY cust\_city;

SQL

Result:

CUST\_CITY COUNT(CUST\_CITY)  
----------------------------------- ----------------  
Hampshair 1  
San Jose 1  
New York 3  
London 4  
Brisban 3  
Chennai 3  
Bangalore 4  
Torento 3  
Mumbai 3

1. Display the city names with the highest number of customers.

Query:

SELECT cust\_city, customers FROM (SELECT cust\_city, COUNT(cust\_city) AS  
customers FROM customer GROUP BY cust\_city) WHERE customers = (SELECT MAX(customers) FROM (SELECT cust\_city, COUNT(cust\_city) AS customers FROM customer GROUP BY cust\_city));

SQL

Result:

CUST\_CITY CUSTOMERS  
----------------------------------- ----------  
London 4  
Bangalore 4

1. What is the number of customers having 'i' in their name?

Query:

SELECT COUNT(cust\_name) FROM customer WHERE cust\_name LIKE '%i%';

SQL

Result:

COUNT(CUST\_NAME)  
----------------  
 13

1. What is the number of customer cities with 'ew' in their name?

Query:

SELECT COUNT(DISTINCT cust\_city) FROM customer WHERE cust\_city LIKE '%ew%';

SQL

Result:

COUNT(DISTINCTCUST\_CITY)  
------------------------  
 1

1. Make a list that will show the number of customers under each grade.

Query:

SELECT grade, COUNT(cust\_name) FROM customer GROUP BY grade;

SQL

Result:

GRADE COUNT(CUST\_NAME)  
---------- ----------------  
 1 9  
 2 10  
 3 5  
 0 1

1. What is the average outstanding amount for the citizens of New York?

Query:

SELECT AVG(outstanding\_amt) FROM customer WHERE cust\_city = 'New York';

SQL

Result:

AVG(OUTSTANDING\_AMT)  
--------------------  
 5000

1. Find the total outstanding amount of high tier customers (i.e. opening amount>6000) under each grade, if there are at least 3 customers under that grade.

Query:

SELECT grade, COUNT(cust\_name), SUM(outstanding\_amt) FROM customer WHERE opening\_amt > 6000 GROUP BY grade HAVING COUNT(cust\_name) >= 3;

SQL

Result:

GRADE COUNT(CUST\_NAME) SUM(OUTSTANDING\_AMT)  
---------- ---------------- --------------------  
 1 6 51000  
 2 6 55000

## Part 2: Descriptive Questions

1. Explain Entity and Entity Set with example.

An entity is a single object, like a person or a book. It has a set of properties that allow us to uniquely identify it. For example, a person has a specific national ID, a gender, a monthly income and so on. The 'values' of these properties allows us to separate that particular object from other objects that are similar to it. An entity is not necessary a physical object. It could be something vague as well, such as a single flight. As long it has specific properties that allow us to uniquely identify it, it can be considered an entity.

An entity set is a group of entities that have similar properties. This means that the properties used to identify them are the same, not the values of the properties. For example, all the people in a country could be part of a single entity set, since they can all be identified using a national ID. However, the value of the national ID is different for each person.

1. Explain Relationship, Relationship Set and Relationship Instance with example.

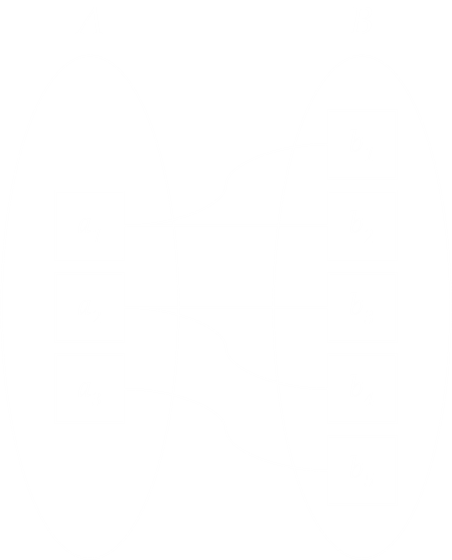
A relationship is the association between two entities. For example, a student and a teacher could have a relationship teaches. This tells us that the teacher teaches the student.

A relationship set is a group of relationships of the same type. For example, we could have a relationship set called teaches between two entity sets, students and teachers.

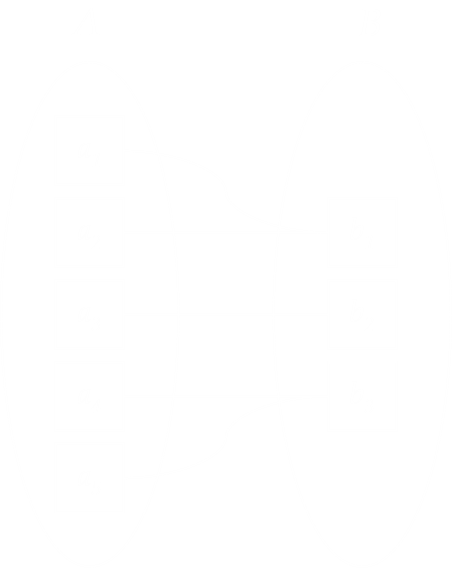
A relationship instance is simple one particular relationship. For example, if we took the relationship set we just defined, teaches, and we looked at the particular relationship between a single teacher and a single student, that would be a relationship instance.

1. Explain different types of mapping cardinalities that may exist between two entities and give proper examples with diagrams.

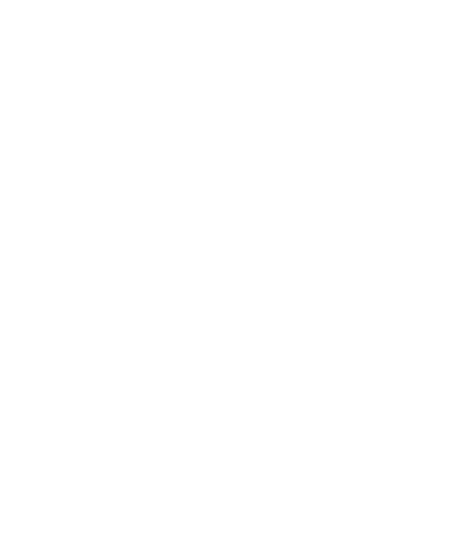
We can have four different types of mapping cardinalities. The first, and most common, is one to many. Consider the relationship between a departments relation and a students relation. A single department could be related to multiple students. This type of mapping would be established by using foreign keys. Visually, it looks something like this:



We can also have the opposite, a many to one cardinality. Here, multiple entities from one relation are related to a single entity of another relation. An example could be customers having multiple phone numbers. This relationship is also established using foreign keys. This is actually the same thing as a one to many cardinality. The thinking is just different. Visually, it looks like this:



Next, we have one to one cardinality. An entity from one relation is related to only one entity from another relation. An example is national identification numbers connected to passport numbers. There can only be one connection between any two numbers in this case. We establish one to one cardinality using foreign keys and an extra constraint, unique. Visually, it looks like this:



The last, and most complicated type of mapping cardinality, is many to many cardinality. Multiple entities from one relation are connected to multiple entities from another relation. If we consider two relations, students and courses, then we would have many to many cardinality between them. Establishing many to many cardinality becomes a little more difficult. We still use foreign keys, but since we cannot assign multiple foreign keys to a single entity at once, we need to create a new relation called the junction table. In that table, we have foreign keys pointing to both the original relations. Since we used foreign keys only in the new table, we can connect multiple foreign keys from one relation to multiple foreign keys of the other relation without any errors. To ensure no combination is repeated, we would need to create a composite primary key using both the foreign keys. The many to many cardinality itself can be represented like this:

