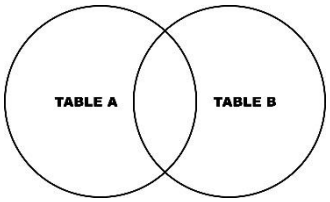
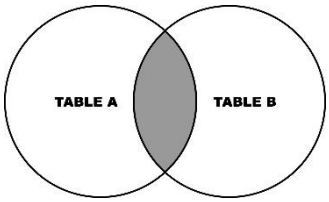
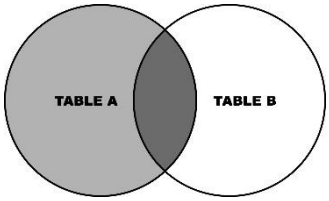
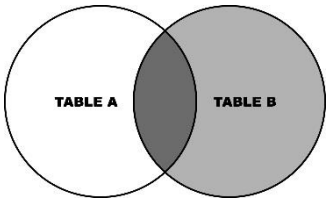
JOINS

A SQL join clause is used to combine data from two or more tables. There are four basic types of SQL joins: inner, left, right, and full. Following table clearly distinguishes between all four joins.

Let's suppose we have two sets of data from TABLE A and TABLE B. Assume both the tables have some common field between them. The result of joining these tables can be visually represented by following Venn diagram.



Now the extent of overlap is determined by how many records in TABLE A matches the records from TABLE B. Now based on what data we want to select from the resulting dataset following cases arise:

Type	Venn Diagram	Description
INNER JOIN		Select all records from TABLE A and TABLE B where join condition is met
LEFT JOIN		Select all records from TABLE A, along with records from TABLE B where join condition is met
RIGHT JOIN		Select all records from TABLE B, along with records from TABLE A where join condition is met

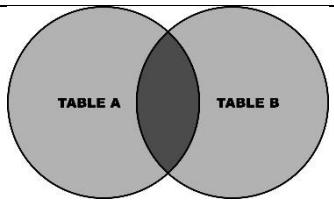
FULL JOIN		Select all records from TABLE A and TABLE B, regardless of whether join condition is met or not

Table 1: employee
Note: We'll refer this table for all the queries in this sheet

employee_id [integer]	first_name [characters]	last_name [characters]	salary [integer]	department [characters]	city [characters]
1	John	Abraham	1000000	Banking	Delhi
2	Michael	Clarke	800000	Insurance	Bangalore
3	Roy	Thomas	700000	Banking	Gujarat
4	Tom	Jose	600000	Insurance	Delhi
5	Jerry	Pinto	650000	Insurance	Bangalore
6	Philip	Mathew	750000	Services	Chandigarh
7	Amir	Khan	650000	Services	Delhi

Table 2: incentives

id [integer]	date [characters]	amount [integer]
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
4	01-JAN-13	4500
5	01-JAN-13	3500
9	01-MAR-13	2000
10	01-MAR-13	2500

INNER JOIN

Example 1: Get employee first names and their corresponding incentives of only those employees who got incentives.

Query 1.1:

```
1  SELECT first_name, amount
2  FROM employee
3  INNER JOIN incentives
4  ON employee.employee_id = incentives.id;
```

Result:

```
+------------+--------+
| first_name | amount |
+------------+--------+
| John       | 5000   |
| Michael    | 3000   |
| Roy        | 4000   |
| Tom        | 4500   |
| Jerry      | 3500   |
```

```
+------------+--------+
5 rows in set (0.07 sec)
```

Query 1.2: Using Table Alias

```
1  SELECT first_name, amount
2  FROM employee AS e
3  INNER JOIN incentives AS i
4  ON e.employee_id = i.id;
```

Result:

```
+------------+--------+
| first_name | amount |
+------------+--------+
| John       |   5000 |
| Michael    |   3000 |
| Roy        |   4000 |
| Tom        |   4500 |
| Jerry      |   3500 |
+------------+--------+
5 rows in set (0.07 sec)
```

Example 2: Display details like employee id, department name, and incentive from employee and incentives table. And order the rows by department name.

Query:

```
1  SELECT
2      employee_id AS emp_id,
3      department,
4      amount
5  FROM employee
6  INNER JOIN incentives
7  ON employee.employee_id = incentives.id
8  ORDER BY department ASC;
```

Result:

```
+--------+------------+--------+
| emp_id | department | amount |
+--------+------------+--------+
|      1 | Banking    |   5000 |
|      3 | Banking    |   4000 |
|      2 | Insurance  |   3000 |
|      4 | Insurance  |   4500 |
|      5 | Insurance  |   3500 |
+--------+------------+--------+
5 rows in set (0.07 sec)
```

Example 3: Display details like first name, and incentive from employee and incentives table and amount greater than 3000.

Query:

```
1  SELECT first_name, amount
2  FROM employee AS e
3  INNER JOIN incentives as i
4  ON e.employee_id = i.id
5  WHERE amount > 3000;
```

Result:

```
+------------+--------+
| first_name | amount |
+------------+--------+
| John       |   5000 |
| Roy        |   4000 |
| Tom        |   4500 |
| Jerry      |   3500 |
+------------+--------+
4 rows in set (0.01 sec)
```

LEFT JOIN

Example 4: Select first name and incentive amount from employee and incentives table for all employees even if they didn't get incentives.

Query:

```
1  SELECT first_name, amount
2  FROM employee AS e
3  LEFT JOIN incentives AS i
4  ON e.employee_id = i.id;
```

Result:

```
+------------+--------+
| first_name | amount |
+------------+--------+
| John       |   5000 |
| Michael    |   3000 |
| Roy        |   4000 |
| Tom        |   4500 |
| Jerry      |   3500 |
| Philip     |   NULL |
| Amir       |   NULL |
+------------+--------+
7 rows in set (0.00 sec)
```

Explanation:
Since we are using LEFT JOIN, the resulting table only contain details from employee table and data that is common between the two tables. Since, Philip and Amir have amount as NULL which means we do not have data for them in incentives table.

RIGHT JOIN

Example 4: Select incentive id and amount along with first name from employee and incentives table for all incentives regardless of whether employee exist or not.

Query:

```
1  SELECT
2      id,
3      first_name,
4      amount
5  FROM employee AS e
6  RIGHT JOIN incentives AS i
7  ON e.employee_id = i.id;
```

Result:

```
+----+------------+--------+
| id | first_name | amount |
+----+------------+--------+
|  1 | John       |   5000 |
|  2 | Michael    |   3000 |
|  3 | Roy        |   4000 |
|  4 | Tom        |   4500 |
|  5 | Jerry      |   3500 |
|  9 | Null       |   2000 |
| 10 | NULL       |   2500 |
+----+------------+--------+
7 rows in set (0.00 sec)
```

Explanation:
Since we are using RIGHT JOIN, the resulting table only contain details from incentives table and data that is common between the two tables. In the table we can see id 9 and 10 have first_name as NULL which means we do not have data for them in employee table.

FULL JOIN

Example 4: Get all employee id and first name, incentive id and amount from employee and incentives table regardless of whether any data is common or not.

Query:

```
1  SELECT
2      employee_id,
3      first_name,
4      id,
5      amount
6  FROM employee AS e
7  LEFT JOIN incentives AS i
8  ON e.employee_id = i.id;
9  UNION
10 SELECT
11     employee_id,
12     first_name,
13     id,
14     amount
15 FROM employee AS e
16 RIGHT JOIN incentives AS i
17 ON e.employee_id = i.id;
```

Result:

```
+-------------+------------+------+--------+
| employee_id | first_name | id   | amount |
+-------------+------------+------+--------+
|           1 | John       |    1 |   5000 |
|           2 | Michael    |    2 |   3000 |
|           3 | Roy        |    3 |   4000 |
|           4 | Tom        |    4 |   4500 |
|           5 | Jerry      |    5 |   3500 |
|           6 | Philip     | NULL |   NULL |
|           7 | Amir       | NULL |   NULL |
|        NULL | NUll       |    9 |   2000 |
|        NULL | NULL       |   10 |   2500 |
+-------------+------------+------+--------+
9 rows in set (0.00 sec)
```

Explanation:

MySQL do not support FULL JOIN keyword. So we achieve it by getting LEFT and RIGHT JOIN and then merging the two tables using UNION clause.

Problem 1	**Refer Table 1 and Table 2 and write queries for the problems below.**
Problem 1.1	**Get all details of those employees from employee table who have incentives.**
Problem 1.2	**Get first name, last name and date (of getting incentive) for each employee who got incentive more than 3500.**
Problem 1.3	**Get first names and department of those employees whose reside in Delhi and got incentive on 01-FEB-13.**
Problem 1.4	**Get details of those employees having incentives outside the range of 2500 – 4000.**
Problem 1.5	**What is the difference between left join and right join? Explain with an example. Note: You need to create your own table for this example.**
Problem 1.6	**Get first name, city and incentives of all employees present in incentive table. Give output for your query.**
Problem 1.7	**Give first name and incentive date for all employees regardless of whether employee got incentive or not. Give output for your query.**
Problem 1.8	**If id from incentives table represent an employee id. Then first name and incentives for each employee id irrespective of whether that id is present in incentives table or employee table.**