DBMS – SQL

JOINS:

1. INNER JOIN: will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be same.

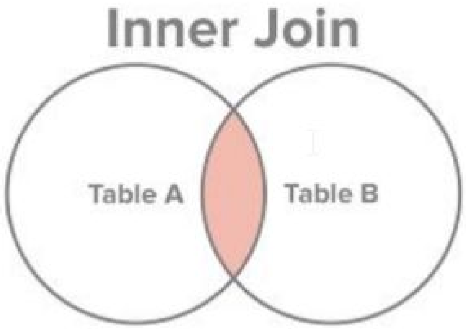
Syntax:

select column_names

from table1

inner join table2

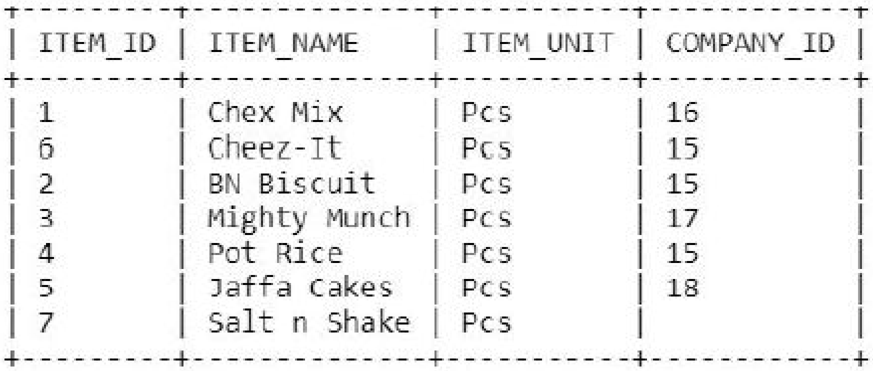
on table1.column_name = table2.column_name;



SQL Code:

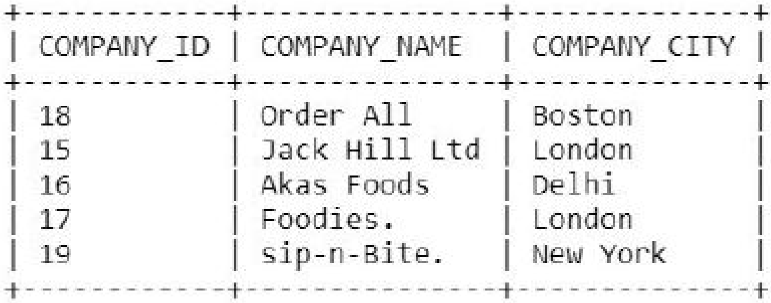
```
SELECT foods . item_name , foods . item_unit ,
company . company_name , company . company_city
FROM foods
17
INNER JOIN company
ON foods . company_id = company . company_id ;
```

Sample table: foods



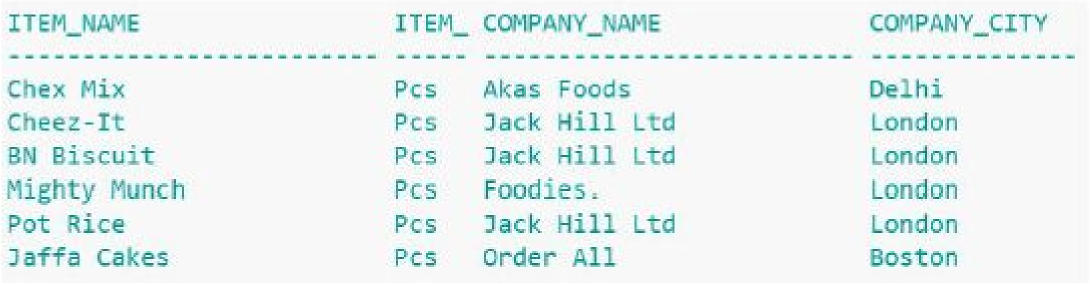
```
+---------+--------------+-----------+------------+
| ITEM_ID | ITEM_NAME    | ITEM_UNIT | COMPANY_ID |
+---------+--------------+-----------+------------+
| 1       | Chex Mix     | Pcs       | 16         |
| 6       | Cheez-It     | Pcs       | 15         |
| 2       | BN Biscuit   | Pcs       | 15         |
| 3       | Mighty Munch | Pcs       | 17         |
| 4       | Pot Rice     | Pcs       | 15         |
| 5       | Jaffa Cakes  | Pcs       | 18         |
| 7       | Salt n Shake | Pcs       |            |
+---------+--------------+-----------+------------+
```

Sample table: company



```
+------------+---------------+--------------+
| COMPANY_ID | COMPANY_NAME  | COMPANY_CITY |
+------------+---------------+--------------+
| 18         | Order All     | Boston       |
| 15         | Jack Hill Ltd | London       |
| 16         | Akas Foods    | Delhi        |
| 17         | Foodies.      | London       |
| 19         | sip-n-Bite.   | New York     |
+------------+---------------+--------------+
```

Output:



```
ITEM_NAME                 ITEM_ COMPANY_NAME              COMPANY_CITY
------------------------- ----- ------------------------- -------------
Chex Mix                  Pcs   Akas Foods                Delhi
Cheez-It                  Pcs   Jack Hill Ltd             London
BN Biscuit                Pcs   Jack Hill Ltd             London
Mighty Munch              Pcs   Foodies.                  London
Pot Rice                  Pcs   Jack Hill Ltd             London
Jaffa Cakes               Pcs   Order All                 Boston
```

2. NATURAL JOIN: It is a special case of equijoin in which equality condition hold on all attributes which have same name in relations R and S (relations on which join operation is applied). While applying natural join on two relations, there is no need to write equality condition explicitly.

Syntax:

SELECT * FROM table1 **NATURAL JOIN** table2;

3. LEFT OUTER JOIN:

This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain null.

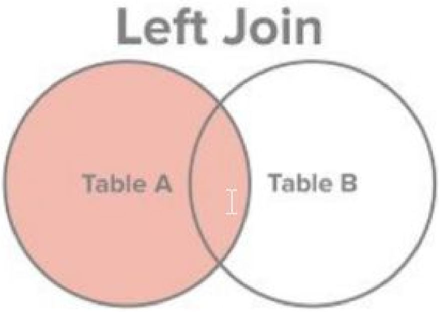
Syntax:

select column_names

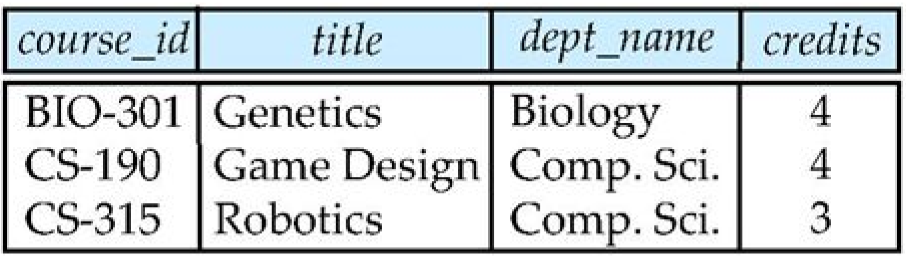
from table1

left join table2

on table1.column_name = table2.column_name;

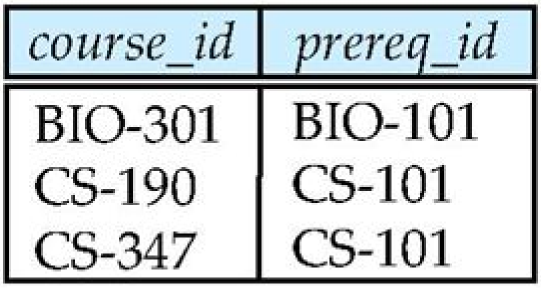


Course:



course_id	*title*	*dept_name*	*credits*
BIO-301	Genetics	Biology	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3

Prereq:

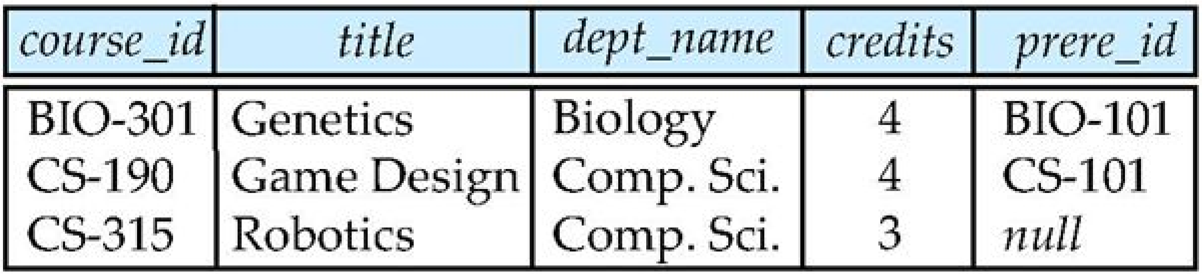


course_id	*prereq_id*
BIO-301	BIO-101
CS-190	CS-101
CS-347	CS-101

Code:

```
SELECT *
FROM course
LEFT OUTER JOIN prereq on course.course_id = prereq.course_id;
```

Output :



course_id	*title*	*dept_name*	*credits*	*prere_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*

4.RIGHT OUTER JOIN:

This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain null.

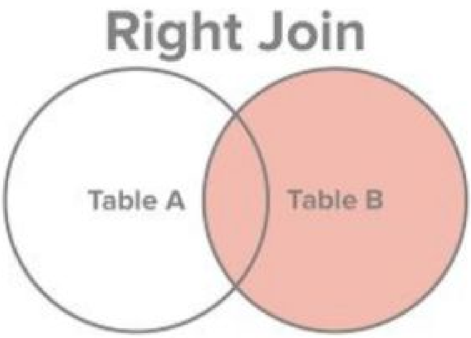
Syntax:

select column_names

from table1

right join table2

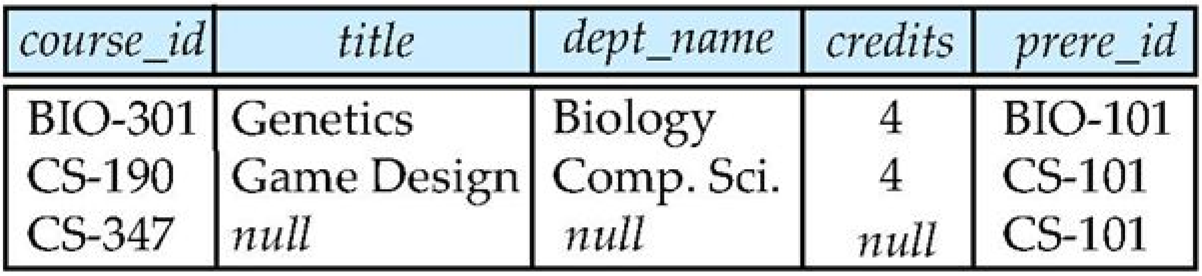
on table1.column_name = table2.column_name;



Code:

```
SELECT * FROM course
RIGHT OUTER JOIN prereq ON course.course_id= prereq.course_id;
```

Output:



course_id	*title*	*dept_name*	*credits*	*prere_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-347	*null*	*null*	*null*	CS-101

5. FULL OUTER JOIN:

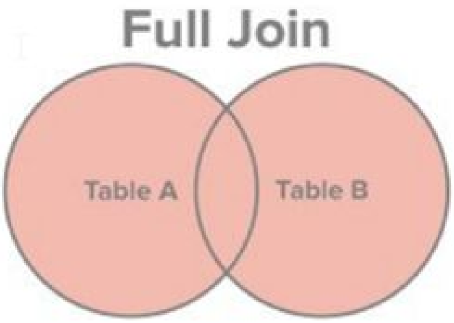
This join creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. The rows for which there is no matching, the result-set will contain NULL values.

Syntax:

select column_names from table1

full outer join table2

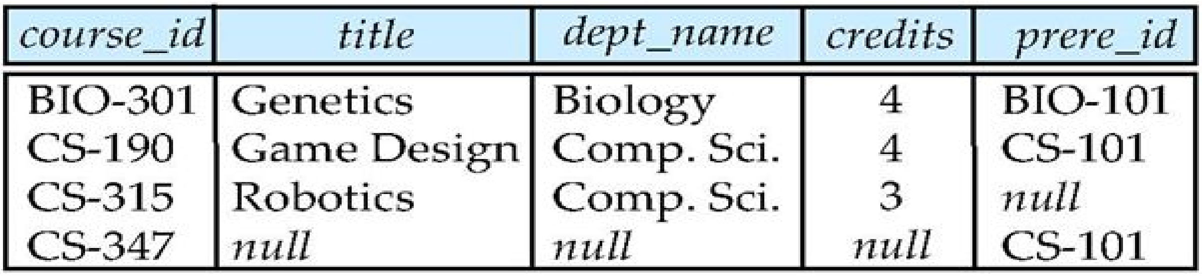
on table1.column_name = table2.column_name;



Code:

```
SELECT * FROM course
FULL OUTER JOIN prereq ON course.course_id= prereq.course_id
```

Output:



course_id	*title*	*dept_name*	*credits*	*prere_id*
BIO-301	Genetics	Biology	4	BIO-101
CS-190	Game Design	Comp. Sci.	4	CS-101
CS-315	Robotics	Comp. Sci.	3	*null*
CS-347	*null*	*null*	*null*	CS-101

Open mysql using the following command->

```
mysql -u root -p --local-infile
```

and after logging in

```
SET GLOBAL local_infile = 1;
```

For departments table:

```
1) create table departments (DEPARTMENT_ID INT PRIMARY KEY,
DEPARTMENT_NAME VARCHAR(30), MANAGER_ID INT,LOCATION_ID INT);
```

```
2) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\departments.txt"
into table departments columns terminated by ',';
```

For employees table:

```
1) create table employees(employee_id int primary key, first_name
varchar(20), last_name varchar(25), email varchar(25), phone_number
varchar(20), hire_date date, job_id varchar(10),salary float, commission_pct
float, manager_id int, department_id int);
```

```
2 ) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\employees.txt" into
table employees columns terminated by ',';
```

For locations table:

```
1) create table locations(location_id int primary key, street_address
varchar(25), postal_code varchar(12), city varchar(30), state_province
varchar(12), country_id char(2));
```

```
2 ) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\locations.txt" into
table locations columns terminated by',';
```

For job_history table:

1) create table job_history(employee_id int, start_date date, end_date date, job_id varchar(10), department_id int, primary key(employee_id,start_date));

2) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\Job_history.txt" into table job_history columns terminated by',';

For countries table:

1) create table countries(country_id char(2) primary key, country_name varchar(40), region_id int);

2) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\countries.txt" into table countries columns terminated by',';

For jobs table:

1) create table jobs (JOB_ID varchar(10) primary key, JOB_TITLE varchar(35) NOT NULL DEFAULT ' ', MIN_SALARY decimal(6,0) DEFAULT 8000, MAX_SALARY decimal(6,0) DEFAULT NULL);

2) load data local infile "C:\\Users\\suraj\\Desktop\\lab3\\jobs.txt" into table jobs columns terminated by',';

EXERSICES:

1. **Write a query to find the addresses (location_id, street_address, city, state_province, country_name) of all the departments.**

2. **Write a query to find the name (first_name, last name), department ID and name of all the employees.**

3. **Write a query to find the name (first_name, last_name), job, department ID and name of the employees who works in London.**

4. **Write a query to find the employee id, name (last_name) along with their manager_id and name (last_name).**

5. **Write a query to find the name (first_name, last_name) and hire date of the employees who was hired after 'Jones'.**

6. **Write a query to get the department name and number of employees in the department.**

7. **Write a query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.**

8. **Write a query to display the department ID and name and first name of manager.**

9. **Write a query to display the department name, manager name, and city.**

10. **Write a query to display the job title and average salary of employees.**

11. **Write a query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job.**

12. **Write a query to display the job history that were done by any employee who is currently drawing more than 10000 of salary.**

13. **Write a query to display department name, name (first_name, last_name), hire date, salary of the manager for all managers whose experience is more than 15 years.**
