

Create table tblDepartment

(

ID int primary key,

DepartmentName varchar(50),

Location varchar(50),

DepartmentHead varchar(50)

);

Insert into tblDepartment (ID,DepartmentName,Location,DepartmentHead)

values (1, 'IT', 'London', 'Rick'),

(2, 'Payroll', 'Delhi', 'Ron'),

(3, 'HR', 'New York', 'Christie'),

(4, 'Other Department', 'Sydney', 'Cindrella');

Create table tblEmployee

(

ID int primary key,

Name varchar(50),

Gender varchar(50),

Salary int,

DepartmentId int ,

CONSTRAINT fk\_employee foreign key(DepartmentId) references tblDepartment(Id)

);

Insert into tblEmployee (ID,Name,Gender,Salary,DepartmentId)

values (1, 'Tom', 'Male', 4000, 1),

(2, 'Pam', 'Female', 3000, 3),

(3, 'John', 'Male', 3500, 1),

(4, 'Sam', 'Male', 4500, 2),

(5, 'Todd', 'Male', 2800, 2),

(6, 'Ben', 'Male', 7000, 1),

(7, 'Sara', 'Female', 4800, 3),

(8, 'Valarie', 'Female', 5500, 1),

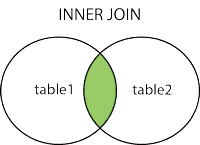
(9, 'James', 'Male', 6500, NULL),

(10, 'Russell', 'Male', 8800, NULL);

* **Joins in SQL** are used to query (retrieve) data from 2 or more related tables. In general tables are related to each other using **foreign key constraints.**
* **In SQL , there are different types of JOINS.**  
  1. CROSS JOIN  
  2. INNER JOIN   
  3. OUTER JOIN   
    
  **Outer Joins are again divided into 3 types**  
  1. Left Join or Left Outer Join  
  2. Right Join or Right Outer Join  
  3. Full Join or Full Outer Join

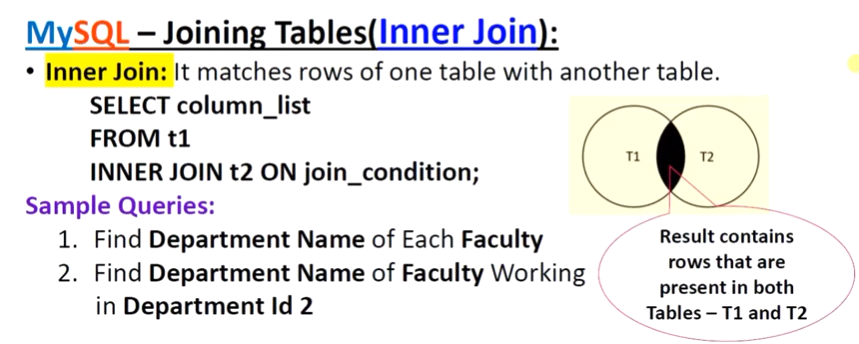
**INNER JOIN**

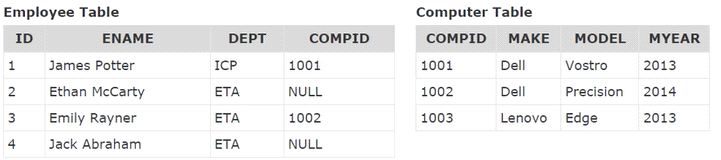
* An INNER JOIN of A and B gives the result of A intersect B. It returns all the common records between two tables. If there’s no related record, it will contain NULL.

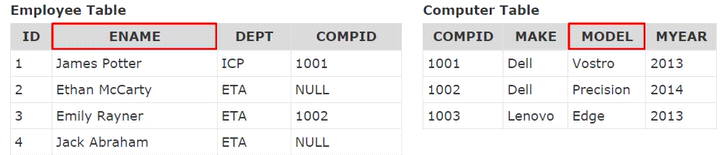


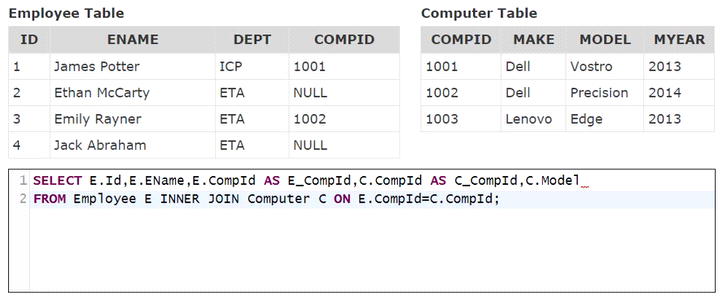
SELECT \* FROM a

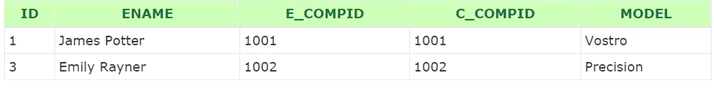
INNER JOIN b on a.id = b.id;

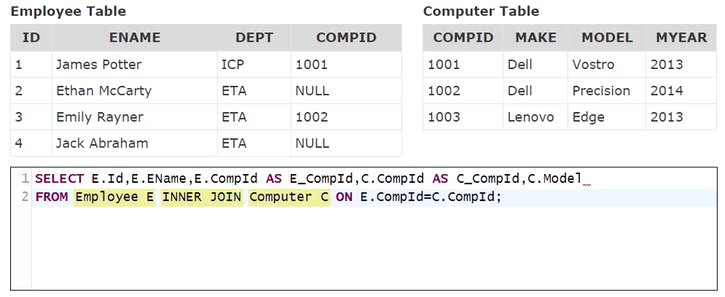


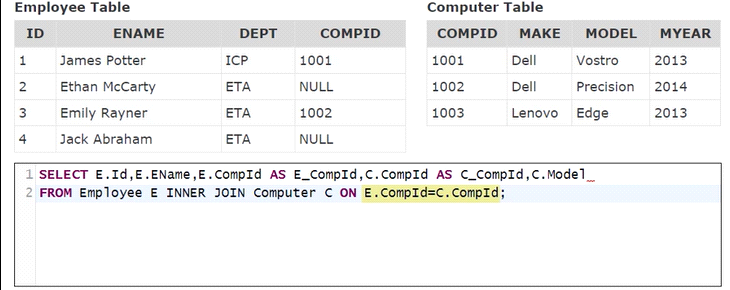




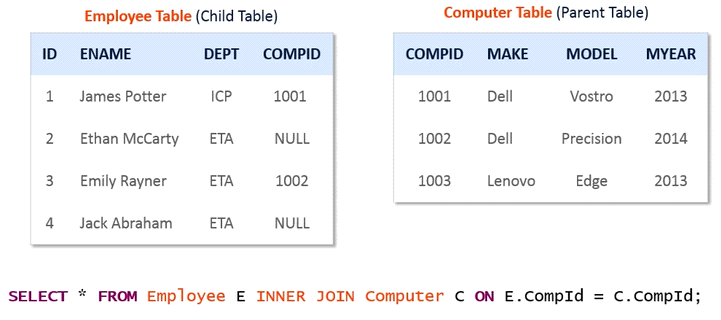


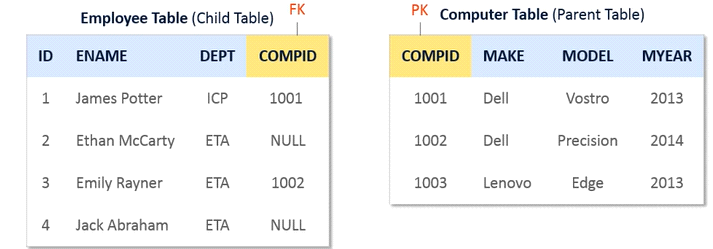


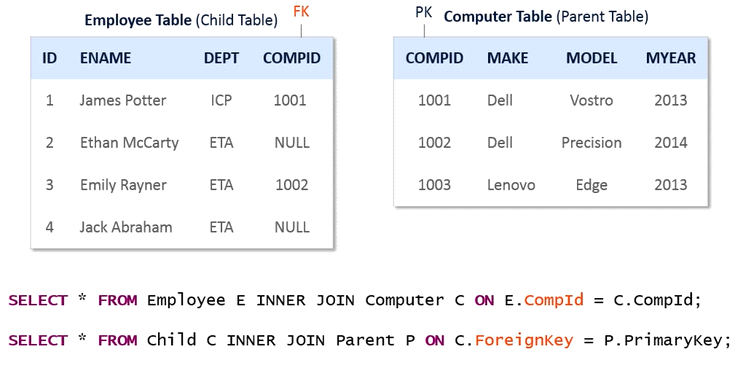


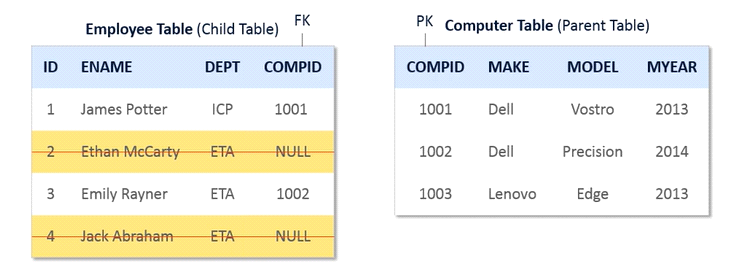


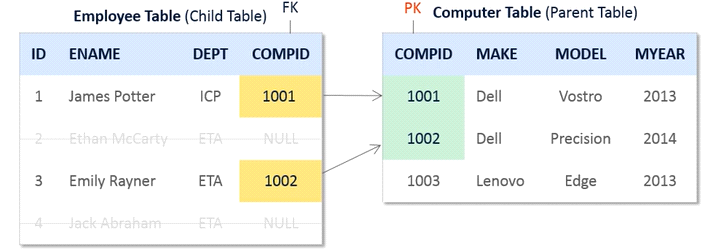






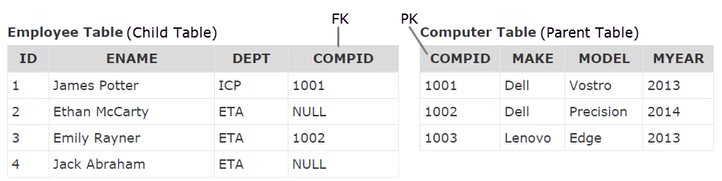


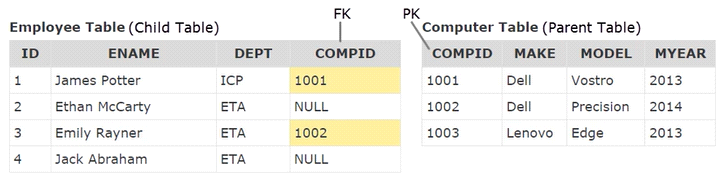


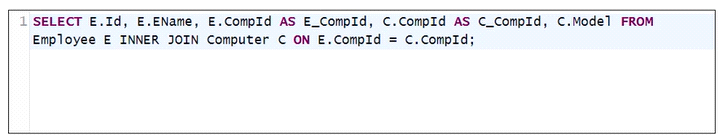


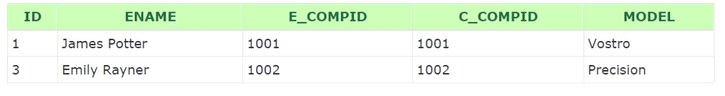


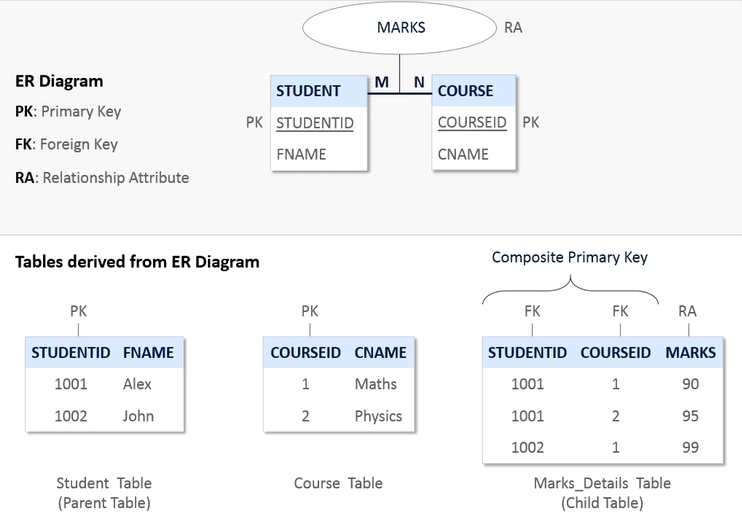


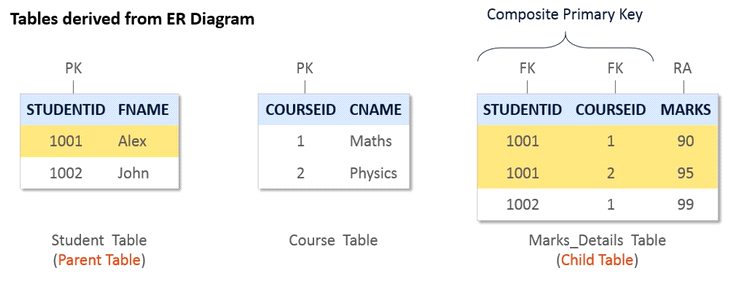


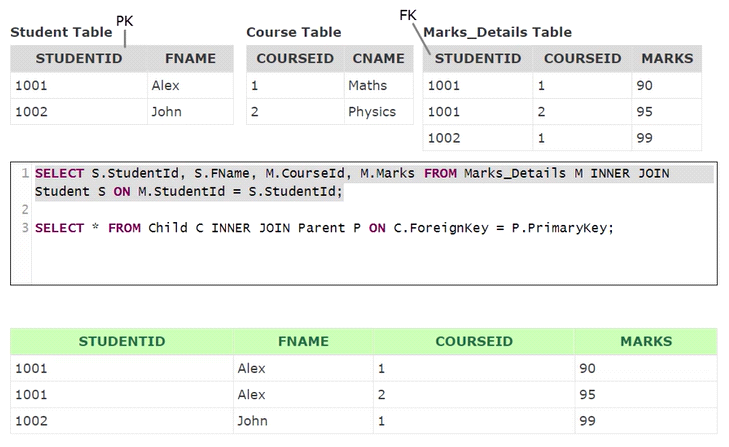


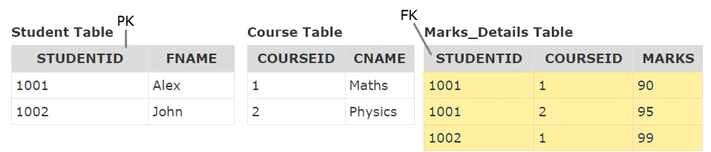


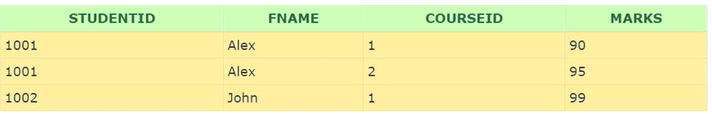


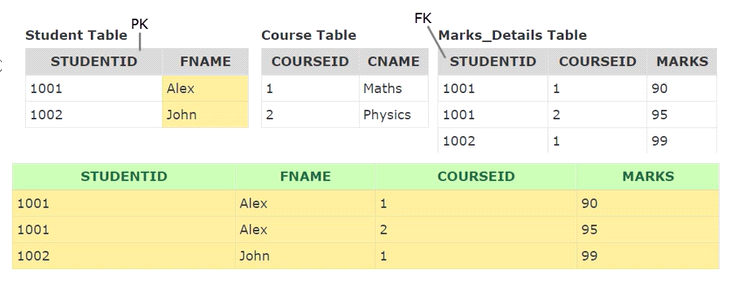


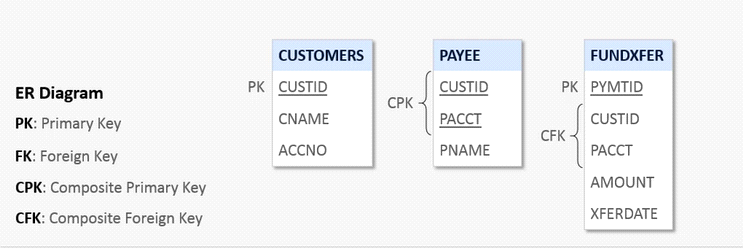


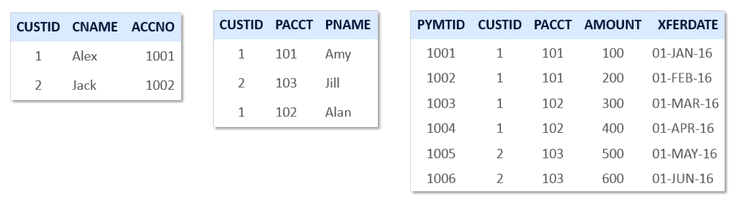


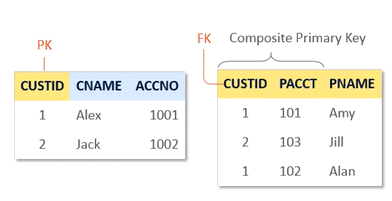


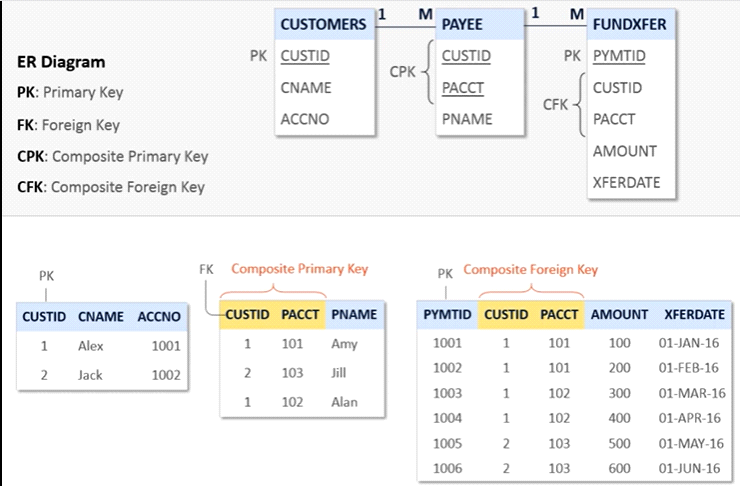


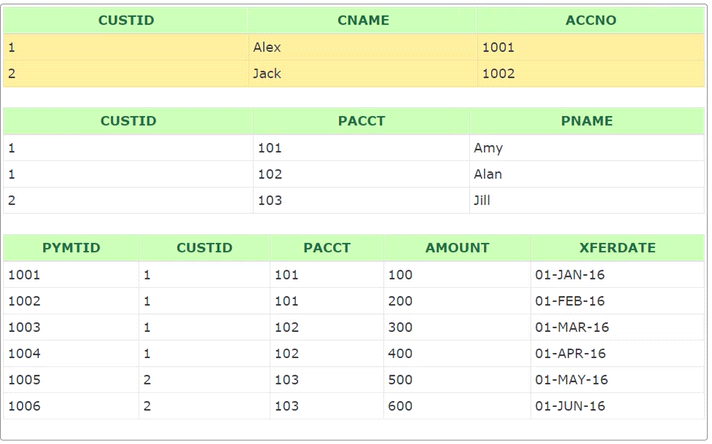


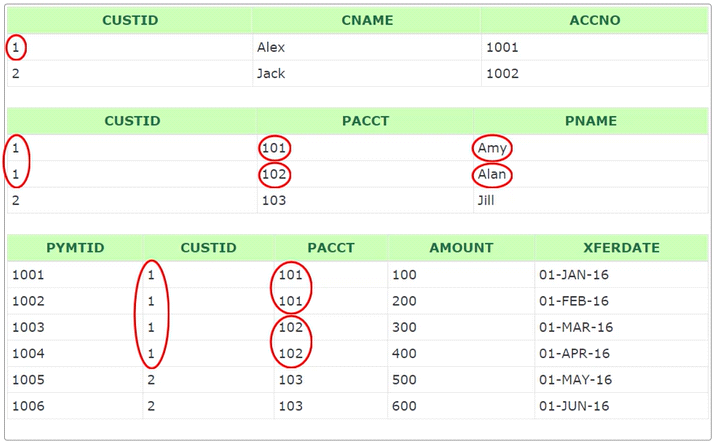


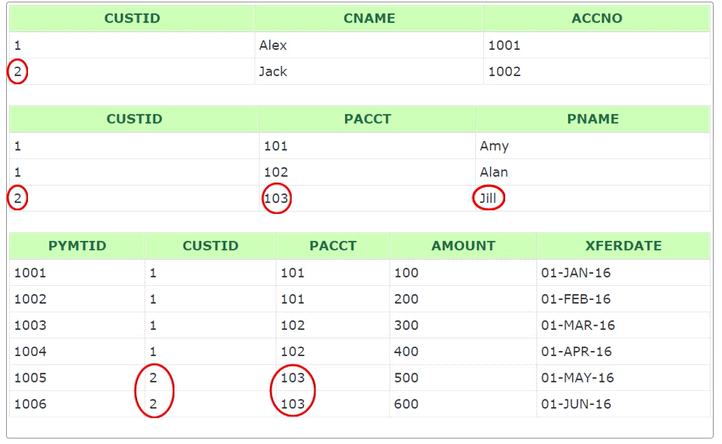


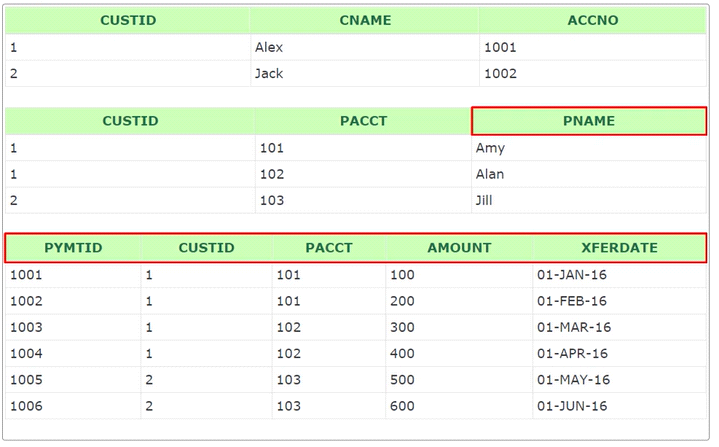




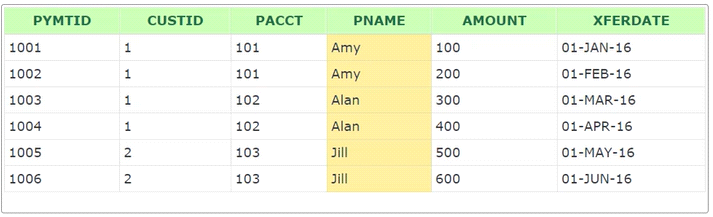




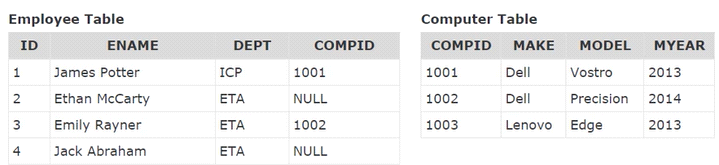


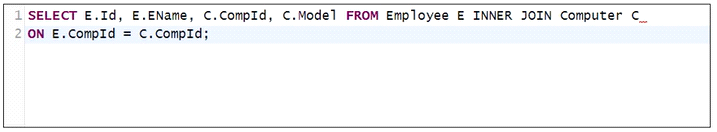


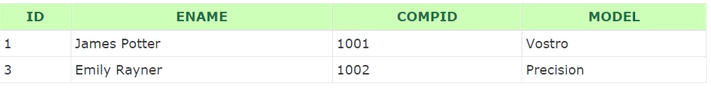


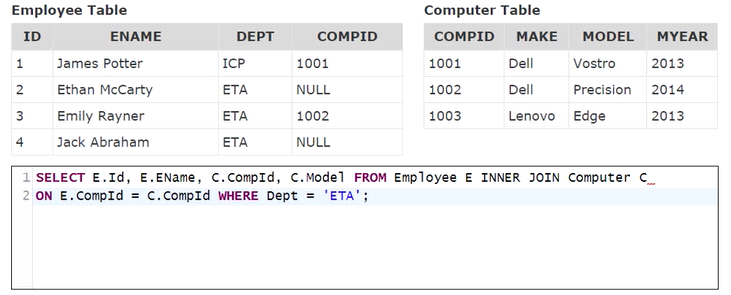




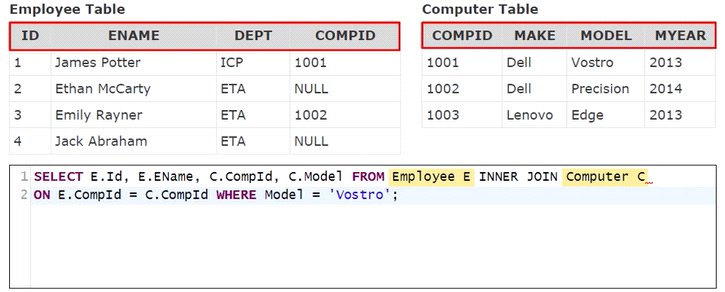




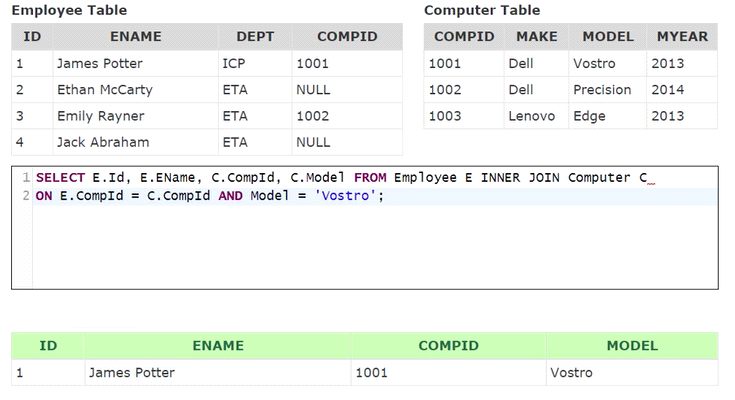


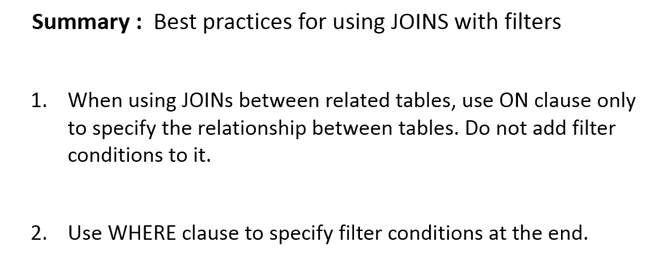


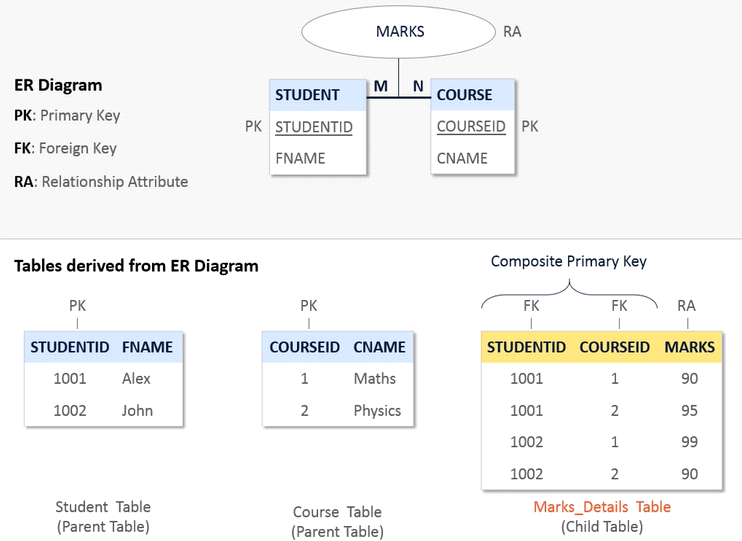


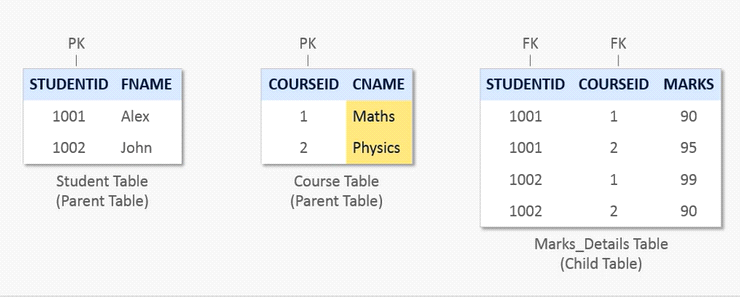


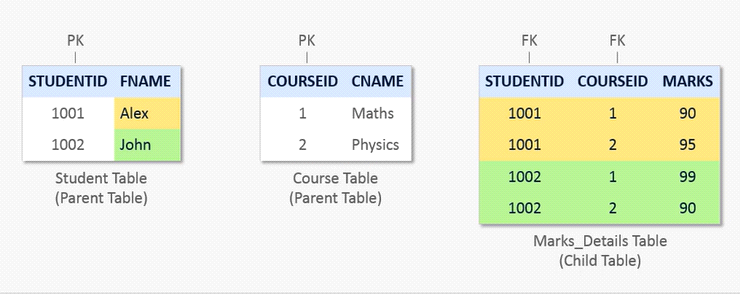


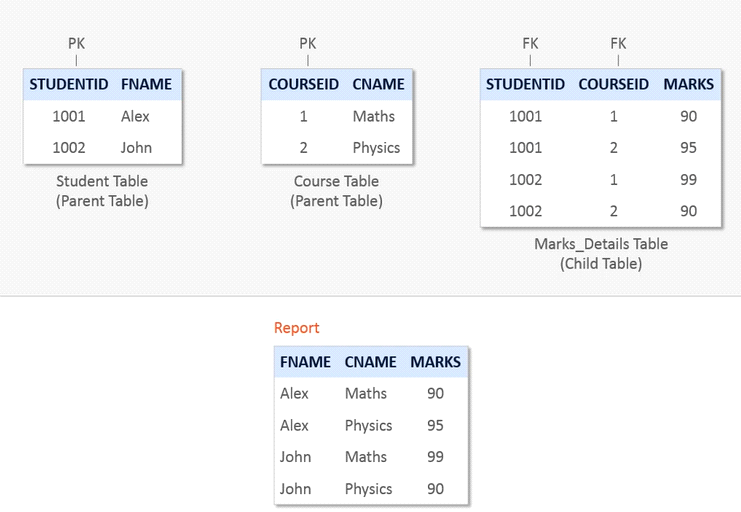


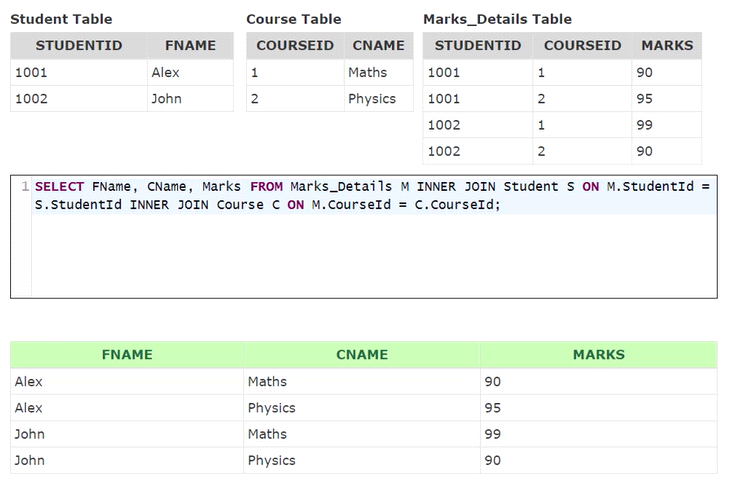


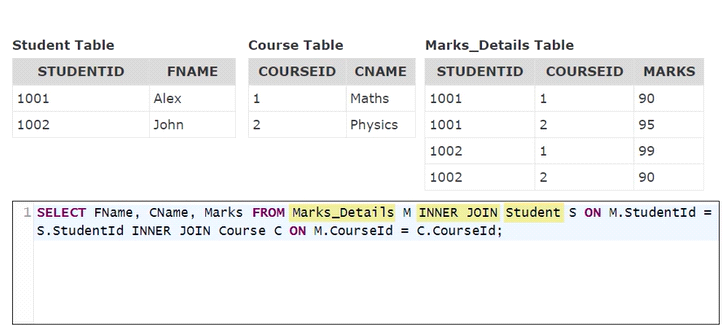


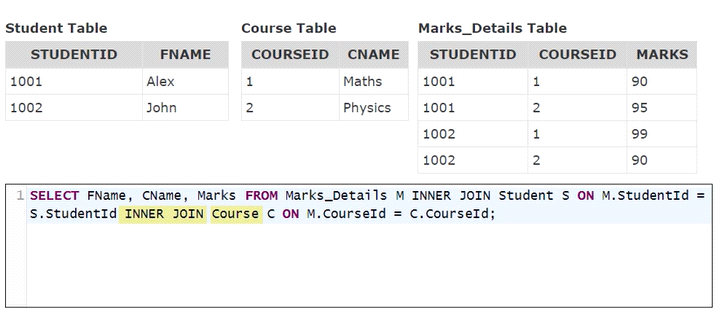


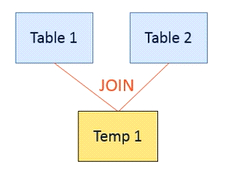


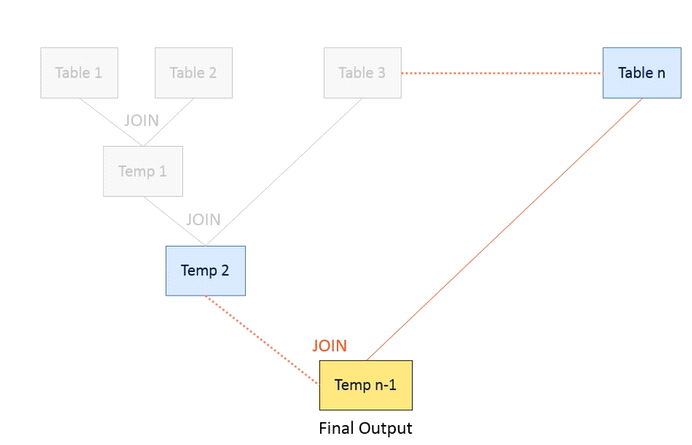


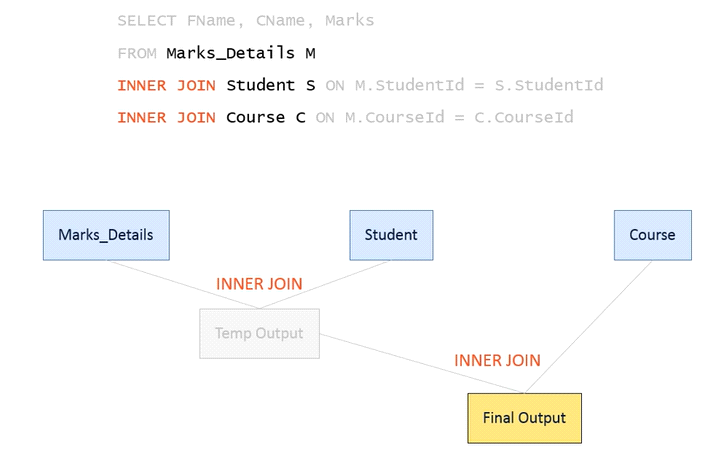


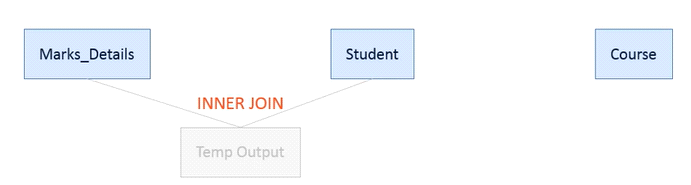


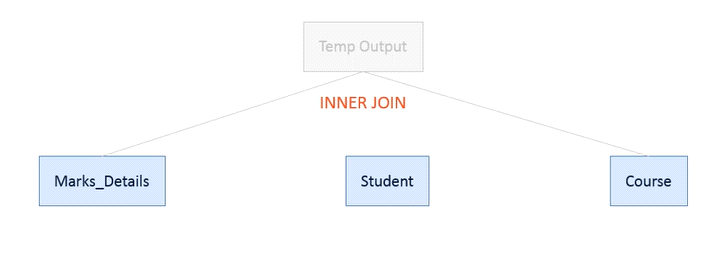


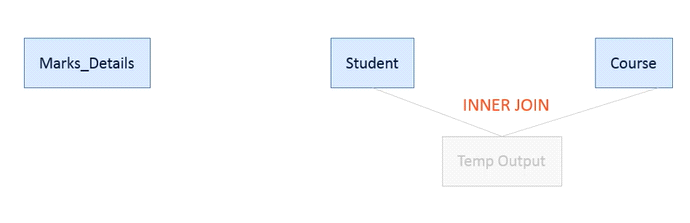


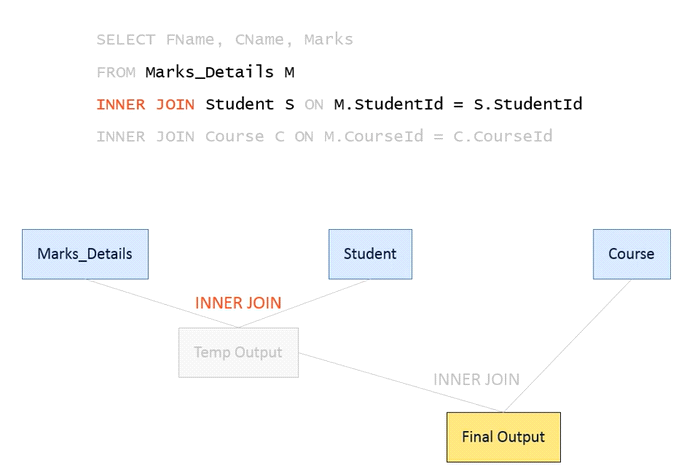


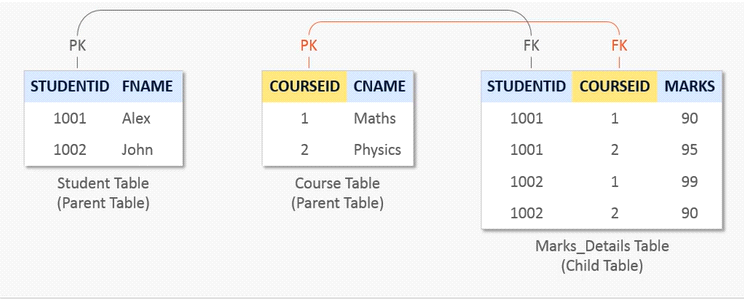


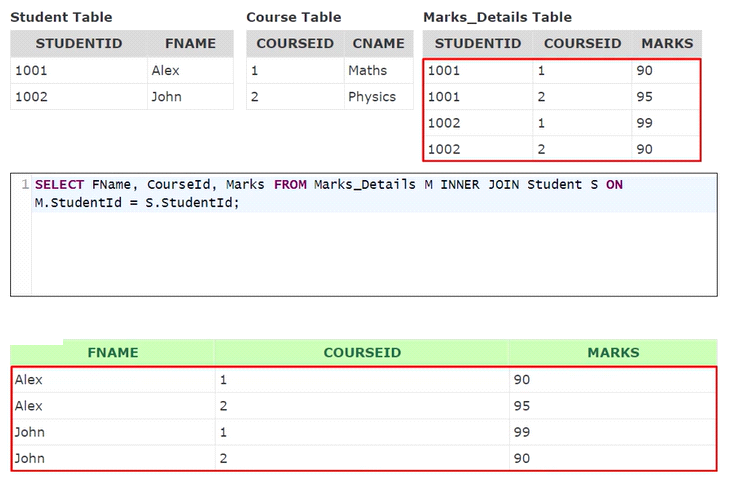


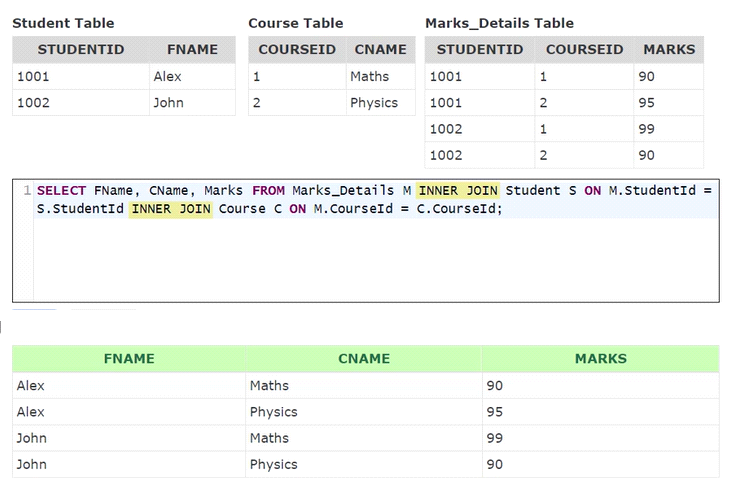




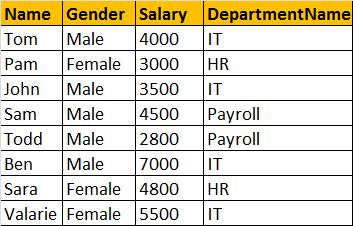








**Write a query, to retrieve Name, Gender, Salary and DepartmentName from Employees and Departments table. The output of the query should be as shown below.**



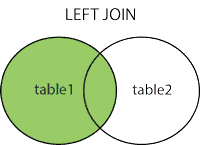
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
INNER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id

**Note:** JOIN or INNER JOIN means the same. It's always better to use INNER JOIN, as this explicitly specifies your intention.

If you look at the output, we got only 8 rows, but in the Employees table, we have 10 rows. We didn't get JAMES and RUSSELL records. This is because the DEPARTMENTID, in Employees table is NULL for these two employees and doesn't match with ID column in Departments table.  
  
So, in summary, INNER JOIN, returns only the matching rows between both the tables. Non matching rows are eliminated.

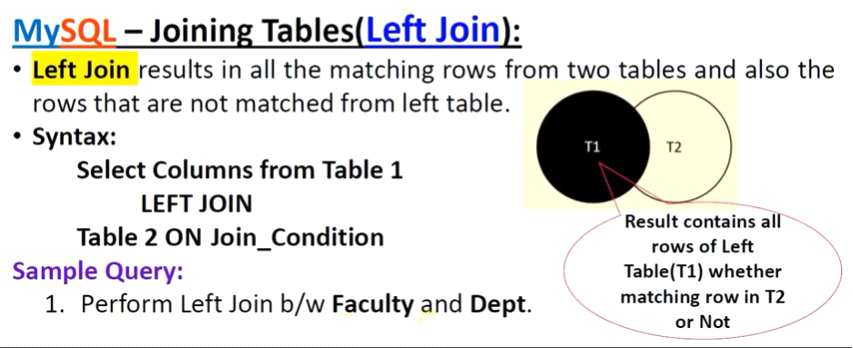
**LEFT JOIN or LEFT OUTER JOIN**

* A LEFT JOIN gives all rows in A, plus any common rows in B. If a record in A doesn’t exist in B, it will return NULL for that row.

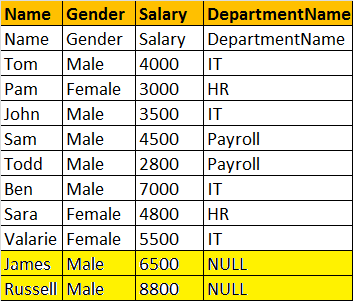


SELECT \* FROM a

*LEFT JOIN* b on a.id = b.id;



* Now, let's say, I want all the rows from the Employees table, including JAMES and RUSSELL records. I want the output, as shown below.



SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
LEFT OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
LEFT JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**Note:** You can use, LEFT JOIN or LEFT OUTER JOIN. OUTER keyowrd is optional  
  
**LEFT JOIN**, returns all the matching rows + non matching rows from the left table. In reality, INNER JOIN and LEFT JOIN are extensively used.

RIGHT JOIN or RIGHT OUTER JOIN

RIGHT JOIN

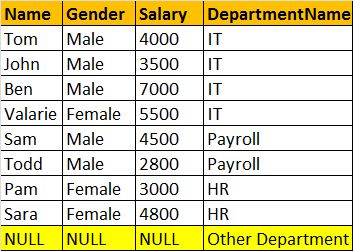
A RIGHT JOIN gives all rows in table B, plus any common rows in A. If a record in B doesn’t exist in A, it will return NULL for that row.



SELECT \* FROM a

RIGHT JOIN b on a.id = b.id;

I want, all the rows from the right table. The query output should be, as shown below.



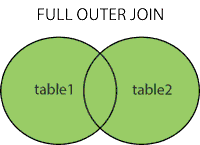
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
RIGHT OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
RIGHT JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**Note:** You can use, RIGHT JOIN or RIGHT OUTER JOIN. OUTER keyowrd is optional  
  
**RIGHT JOIN**, returns all the matching rows + non matching rows from the right table.

RIGHT JOIN, returns all the matching rows + non matching rows from the right table.

**FULL JOIN or FULL OUTER JOIN**

**FULL JOIN**

* MySQL does not support FULL JOIN, so you have to combine JOIN, UNION and LEFT JOIN to get an equivalent. It gives the results of A union B. It returns all records from both tables. Those columns which exist in only one table will contain NULL in the opposite table.



SELECT \* FROM a

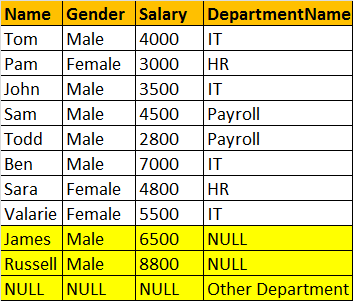
LEFT JOIN b ON a.id = b.id

UNION

SELECT \* FROM a

RIGHT JOIN b ON a.id = b.id

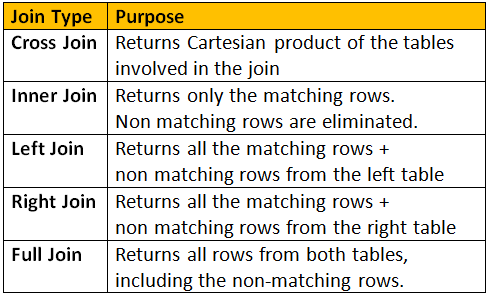
I want all the rows from both the tables involved in the join. The query output should be, as shown below.



select name,gender,salary,departmentname from tblemployee e left outer join tbldepartment d on e.departmentid=d.id union select name,gender,salary,departmentname from tblemployee e right outer join tbldepartment d on e.departmentid=d.id;;

Note:- SQL Server and Oracle Full join support

SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
FULL OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
OR  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
FULL JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id



**CROSS JOIN**  
CROSS JOIN, produces the cartesian product of the 2 tables involved in the join. For example, in the Employees table we have 10 rows and in the Departments table we have 4 rows. So, a cross join between these 2 tables produces 40 rows. Cross Join shouldn't have ON clause.

**CROSS JOIN Query:**  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
CROSS JOIN tblDepartment



