**Practice**: [https://www.w3resource.com/sql-exercises/sql-joins-exercises.php#SQLEDITOR](https://www.w3resource.com/sql-exercises/sql-joins-exercises.php%23SQLEDITOR)

**Explanation**: https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins/

A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:

**INNER JOIN**

**LEFT JOIN** also known as Left Outer Join

**RIGHT JOIN** also known as right outer Join

**FULL JOIN** also known as full outer join – It is a combination of both left join and right join

**SELF JOIN**: join between two copies of the same table,

It's basically used where there is any relationship between rows stored in the same table

**CARTESIAN JOIN**: The CARTESIAN JOIN is also known as **CROSS JOIN**. In a CARTESIAN JOIN there is a join for each row of one table to every row of another table. This usually happens when the matching column or WHERE condition is not specified.

* In the absence of a WHERE condition the CARTESIAN JOIN will behave like a CARTESIAN PRODUCT . i.e., the number of rows in the result-set is the product of the number of rows of the two tables.
* In the presence of WHERE condition this JOIN will function like a INNER JOIN.
* Generally speaking, Cross join is similar to an inner join where the join-condition will always evaluate **to True**

**NATURAL JOIN**: Natural Join joins two tables based on same attribute name and datatypes. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.

salesman:

CREATE TABLE salesman

(

salesman\_id INT,

name nvarchar(40),

city nvarchar(40),

commission FLOAT(8,4)

);

Insert into salesman values (5001 ,'James Hoog','New York',0.15);

Insert into salesman values (5002 ,'Nail Knite','Paris',0.13);

Insert into salesman values (5005 ,'Pit Alex','London',0.11);

Insert into salesman values (5006 ,'Mc Lyon','Paris',0.14);

Insert into salesman values (5007 ,'Paul Adam','Rome',0.13);

Insert into salesman values (5003 ,'Lauson Hen','San Jose',0.12);

Select \* from salesman;

customer:

CREATE TABLE customer

(

customer\_id INT,

cust\_name nvarchar(40),

city nvarchar(40),

grade INT,

salesman\_id INT

);

Insert into customer values (3002,'Nick Rimando','New York',100,5001);

Insert into customer values (3007,'Brad Davis','New York',200,5001);

Insert into customer values (3005,'Graham Zusi','California',100,5002);

Insert into customer values (3008,'Julian Green','London',300,5002);

Insert into customer values (3004,'Fabian Johnson','Paris',300,5006);

Insert into customer values (3009,'Geoff Cameron','Berlin',100,5003);

Insert into customer values (3003,'Jozy Altidor','Moscow',200,5007);

Insert into customer values (3001,'Brad Guzan','London',null,5005);

Select \* from customer;

CREATE TABLE orders

(

ord\_no INT,

purch\_amt FLOAT(8,4),

ord\_date date,

customer\_id INT,

salesman\_id INT

);

Insert into Orders values (70001,150.5,'2012-10-05',3005,5002);

Insert into Orders values (70009,270.65,'2012-09-10',3001,5005);

Insert into Orders values (70002,65.26,'2012-10-05',3002,5001);

Insert into Orders values (70004,110.5,'2012-08-17',3009,5003);

Insert into Orders values (70007,948.5,'2012-09-10',3005,5002);

Insert into Orders values (70005,2400.6,'2012-07-27',3007,5001);

Insert into Orders values (70008,5760,'2012-09-10',3002,5001);

Insert into Orders values (70010,1983.43,'2012-10-10',3004,5006);

Insert into Orders values (70003,2480.4,'2012-10-10',3009,5003);

Insert into Orders values (70012,250.45,'2012-06-27',3008,5002);

Insert into Orders values (70011,75.29,'2012-08-17',3003,5007);

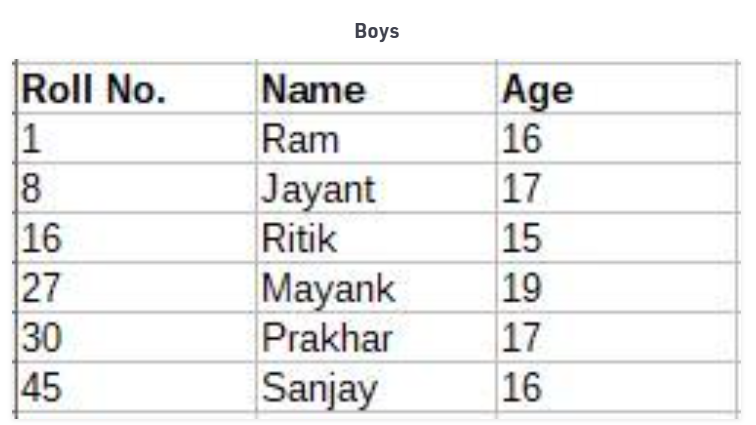
Insert into Orders values (70013,3045.6,'2012-04-25',3002,5001);

Select \* from orders;

**JOIN vs UNION**

* JOIN in SQL is used to combine data from many tables based on a matched condition between them. The data combined using JOIN statement results into new columns.

Consider the two table





*SELECT Boys.Name, Boys.Age, Girls.Address FROM Boys*

*INNER JOIN Girls ON Boys.Rollno = Girls.Rollno;*

Output:



[UNION](https://www.geeksforgeeks.org/sql-union-clause/):  
UNION in SQL is used to combine the result-set of two or more SELECT statements. The data combined using UNION statement is into results into new distinct rows.

**Example:**

SELECT Name

FROM Boys

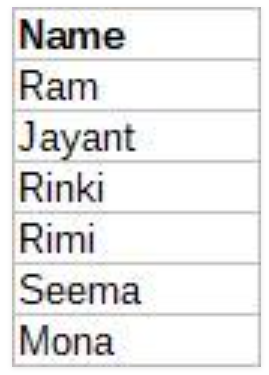
WHERE Rollno < 16

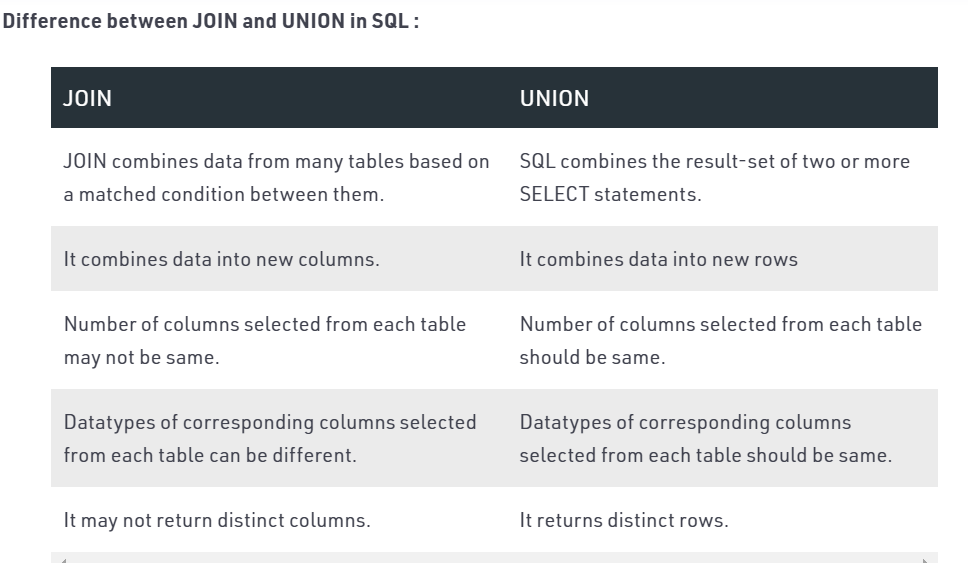
UNION

SELECT Name

FROM Girls

WHERE Rollno > 9





Union gives only distinct values where as Union ALL returns duplicate values

