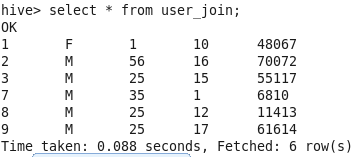
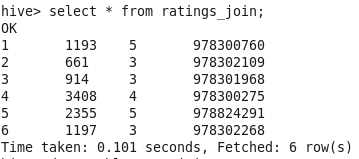
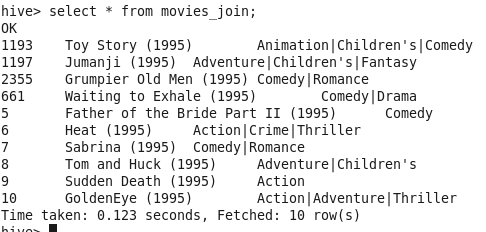
**User:**



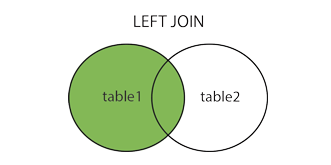
**Rating:**



**Movies:**

****

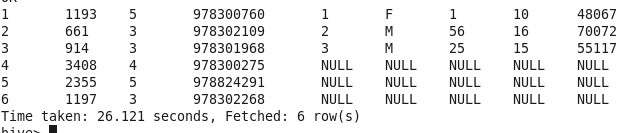
**Left-Outer Join:**

****

In this join, all the records from the left-hand table that match the WHERE clause are returned. If the right-hand table doesn’t have a record that matches the ON criteria, NULL is used for each column selected from the right-hand table.

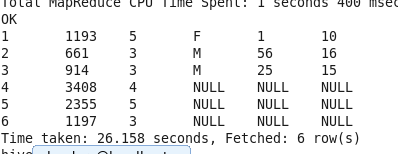
**Left-Outer Join:**

SELECT \* FROM ratings\_join r left OUTER JOIN user\_join u ON r.userid = u.userid;



**Left-Outer Join without where clause:**

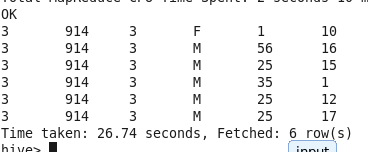
SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid;



**Left-Outer Join without on clause:**

Performs Cartesian Product, just performs cross product of two tables and implements where clause.

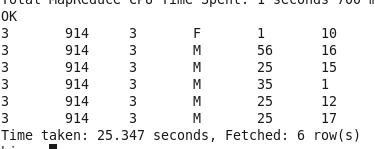
SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u where r.userid = 3;



**Cartesian Join when used on specific user-id value returns same result:**

select r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation from ratings\_join r join user\_join u where r.userid = 3;

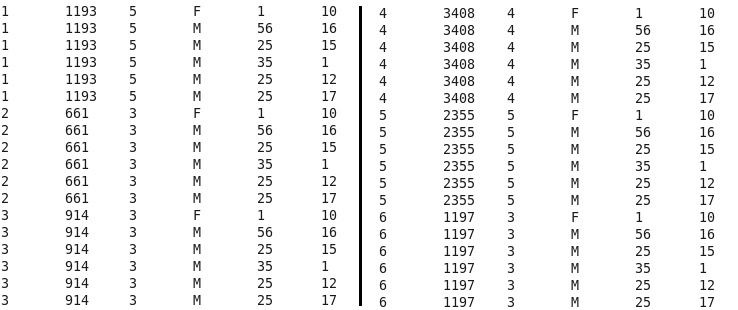
cartesian-join is different from inner-join, the latter join uses "ON" clause.



**Left-Outer Join without on clause and where clause:**

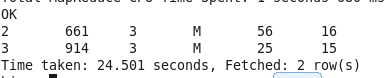
Performs Cartesian join because on clause is not present.

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u;



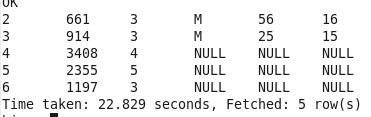
**Left-Outer Join with where clause in Gender:**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid where u.Gender = "M";



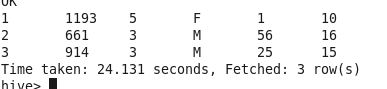
**Left-Outer Join with where clause in Gender and check NULL:**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid where (u.Gender = "M" or u.Gender IS NULL);



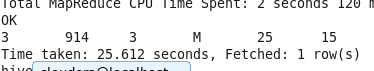
**Left-Outer Join with where clause in Gender and check NOT NULL :**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid where (u.Gender = "M" or u.Gender IS NOT NULL);

****

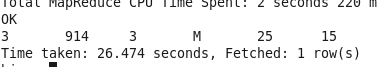
**Left-outer Join with ON:**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid where u.userid = 3;



**Inner Join when used on specific user-id value returns same result:**

select r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation from ratings\_join r join user\_join u ON r.userid = u.userid where r.userid = 3;



**MISC:**

**Performs left outer join(Left-outer-join, ON) and apply WHERE clause:**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u ON r.userid = u.userid where r.userid = 6; 1 RECORD WITH NULL

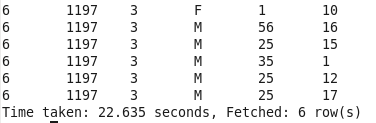


**Performs inner join(Join, ON) and apply WHERE clause:**

select r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation from ratings\_join r join user\_join u ON r.userid = u.userid where r.userid = 6; NO OUTPUT

**Performs Cartesian Join(Join, No ON CLAUSE) and apply WHERE clause:**

select r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation from ratings\_join r join user\_join u where r.userid = 6; MANY 6ES



**Performs Cartesian Join(Left-outer-join, No ON CLAUSE) and apply WHERE clause:**

SELECT r.userid, r.movieid, r.rating, u.Gender, u.Age, u.Occupation FROM ratings\_join r LEFT OUTER JOIN user\_join u where r.userid = 6; Many 6es

