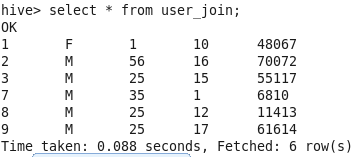
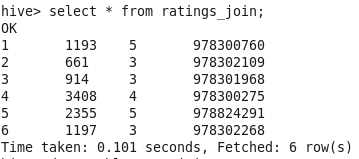
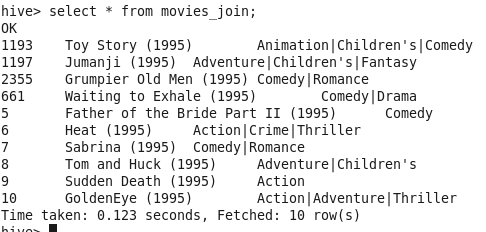
**User:**



**Rating:**

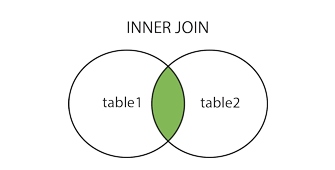


**Movies:**

****

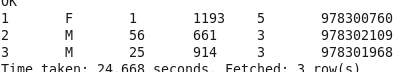
**Inner-Join:**

An inner join requires each record in the two joined tables to have matching records, and is a commonly used join operation in applications but should not be assumed to be the best choice in all situations. **Inner Join without ON clause acts as Cartesian join.**



**Inner Join - ON Clause:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid;



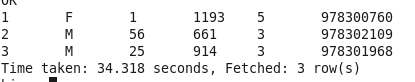
**Inner Join - ON, WHERE Clause:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid where u.userid = 1;

2.PNG

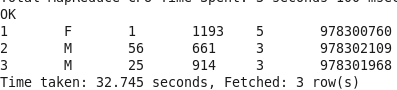
**Inner Join - SORTBY Clause:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid SORT BY u.userid;



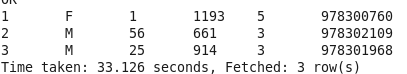
**Inner Join - SORTBY Clause (Two Columns):**

select u.userid as uid,u.Gender as gen, u.age,r.movieid, r.rating, r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid SORT BY gen,uid; #dual sorting by two columns



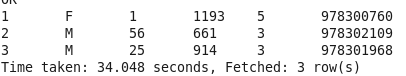
**Inner Join - ORDERBY Clause (Two Columns):**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid ORDER BY u.userid;



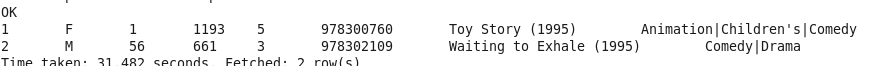
**Inner Join - ON (CONDITION) - ORDERBY Clause:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on ( u.userid = r.userid AND u.userid = r.userid ) ORDER BY u.userid;



**Inner Join - Three Table:**

select u.userid,u.Gender,u.age,r.movieid, r.rating, r.timestamp\_rating, m.title, m.genres from user\_join u join ratings\_join r on u.userid = r.userid join movies\_join m on r.movieid = m.movieid ORDER BY u.userid,m.genres,r.movieid; # Two join clause which joins all three table



**Illegal:**

**No Less-Equal, Greater than Equal in ON Clause:**

select u.userid as uid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid <= r.userid;

Returns an error because it is difficult to implement these kinds of joins in MapReduce. It turns out that Pig offers a **cross product feature** that makes it possible to implement this join, even though Pig’s native join feature doesn’t support it, either.

**Hive does not currently support using OR between predicates in ON clauses:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on ( u.userid = r.userid OR u.userid = r.userid ) ORDER BY u.userid;

**GroupBY is not working here because no aggregate function is used:**

select u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating from user\_join u join ratings\_join r on u.userid = r.userid Group BY u.userid;

**Optimization:**

**Joining two or more tables:**

select u.userid,u.Gender,u.age,r.movieid, r.rating, r.timestamp\_rating, m.title, m.genres from users\_limited u join ratings\_limited r on u.userid = r.userid

join movies\_limited m on r.movieid = m.movieid ORDER BY u.userid, m.genres, r.movieid;

When joining three or more tables, if **every ON clause uses the same join key, a single MapReduce job will be used**. In the above example its different. Single MapReduce job is used if every ON clause uses u.userid/r.userid/r.movieid or m.movieid. **Hive also assumes that the last table in the query is the largest. Therefore, One should structure your join queries so the largest table is last.**

**Hinting to hive regarding the larger table:**

Hive will attempt to stream the rating table r, even though it’s not the last table in the query.

select **/\*+ STREAMTABLE(r) \*/** u.userid,u.Gender,u.age,r.movieid, r.rating,r.timestamp\_rating,m.title,m.genres from users\_limited u join ratings\_limited r on u.userid = r.userid join movies\_limited m on r.movieid = m.movieid ORDER BY u.userid,m.genres,r.movieid;