Comparison Chart

Feature	SQL	HiveQL
Updates	UPDATE, DELETE INSERT,	UPDATE, DELETE INSERT,
Transaction	Supported	Limited Support Supported
Indexes	Supported	Supported
Data Types	Integral, floating-point, fixed-point, text and binary strings, temporal	Boolean, integral, floating-point, fixed-point, text and binary strings, temporal, array, map, struct
Functions	Hundreds of built-in functions	Hundreds of built-in functions
Multitable inserts	Not supported	Supported
Select	Supported	Supported with SORT BY clause for partial ordering and LIMIT to restrict number of rows returned
Joins	Supported	Inner joins, outer joins, semi join, map joins, cross joins
Subqueries	Used in any clause	Used in FROM, WHERE, or HAVING clauses
Views	Updatable	Read-only

Project 3 Queries:

1. **Calculate male to female ratio.**

```
SELECT gender, Count(*) / (select count(*) FROM Passenger) AS Sex_Ratio
FROM passenger
GROUP BY gender ;
```

IN HIVE

```
SELECT gender, Count(*)/ (select count(*) FROM Passenger) AS Sex_Ratio
FROM passenger
GROUP BY gender ;
```

2. **Distinct Female Passenger (3 Way Join)**

```
SELECT DISTINCT(passenger_name), route_point, train_name FROM
passenger p
INNER JOIN train t
ON t.train_id = p.train_id
INNER JOIN route r
ON t.train_id = r.train_id
WHERE gender="F"
GROUP BY Passenger_Name
ORDER BY Passenger_Name ;
```

IN HIVE

mapjoin is useful to cache small tables in memory.

```
select /*+ MAPJOIN(dept) */ passenger.passenger_name,train.train_name
from passenger join train on t.train_id = p.train_id;
```

3. **Count number of Train Stops for each train**

```
SELECT DISTINCT t.train_id, train_name, count(*) AS TrainStops,Arrival_Time
FROM train t
INNER JOIN route r ON t.train_id = r.train_id
GROUP BY t.train_name, DATE(arrival_time)
ORDER BY COUNT(*);
```

4. Create table as select

In SQL:

```
CREATE TABLE mytraininfo AS
SELECT ticket_id, train_id, gender, passenger_name, age ,destination
FROM passenger p
LEFT JOIN Train t on p.train_id = t.train_id
WHERE ( p.gender= ‘M’ and age >21) ;
```

In Hive:

```
CREATE TABLE mytraininfo AS
SELECT ticket_id, train_id, gender, passenger_name, age ,destination
```

```
FROM passenger p
LEFT JOIN Train t on p.train_id = t.train_id
WHERE ( p.gender= ‘M’ and age >21) ;
```

5. **Multiple Table insert**

```
hive> from sample_view
> insert overwrite table Train
> select train_id,train_name
>insert overwrite table Route
>select train_id, arrival_time,route_point
>insert overwrite table Passenger
>select train_id, ticket_id;
```

6. 4 Way Join

```
SELECT DISTINCT(Passenger_name), Train_name, Ticket_price, ti.Ticket_id,
Ticket_status, Status_date  FROM Passenger p
INNER JOIN Train t ON t.train_id = p.train_id
INNER JOIN Ticket ti ON p.ticket_id = p.ticket_id
INNER JOIN TrainStatus ts ON t.train_id = ts.train_id
WHERE Ticket_price = 35;
```

7. Nested Queries

```
SELECT train_name FROM train
WHERE train_id IN
(SELECT route.train_id FROM route
WHERE route.route_point="San Francisco");
```

Hive doesn’t support sub queries anywhere only than the from clause. So you can't use subquery in where clause you have to create a temp table in from clause and you can use that table. Now if you create a temp table and then you are using it in your where clause than to refer that temp table it has to again run the fetching query so again it will not support.

8. VIEW

```
CREATE VIEW TrainStops AS
SELECT t.train_id, train_name, group_concat(distinct route_point) AS Routes
FROM route r
INNER JOIN Train t ON t.train_id = r.train_id
GROUP BY train_id;
```