**Final Project Deliverable 2 (100 Points)**

**Student : Aditi Bhujbal.**

Based on the star schema created, answer the following using SQL:

1. Find the seller user id and buy user id such that the buyer has bought at least one item from the seller but the buyer and seller are located in different states.

**Answer:**

**select distinct a.selleruserid, b.state seller\_state, a.buyersuserid, c.state buyer\_state**

**from sales\_Fact a, sellers\_dim b, buyers\_dim c**

**where a.selleruserid = b.userid**

**and a.buyersuserid = c.userid**

**and b.state != c.state;**

**--Alternate way with join**

**select distinct sf.buyersuserid, b.state buyer\_location,**

**sf.selleruserid, s.state seller\_location**

**from sales\_fact sf**

**join sellers\_dim s on (sf.selleruserid = s.userid)**

**join buyers\_dim b on (sf.buyersuserid = b.userid)**

**where b.state != s.state;**

**--using correlated subquery**

**select distinct selleruserid, buyersuserid**

**from sales\_fact a**

**where not exists**

**(select 'X'**

**from sellers\_dim s, buyers\_dim b**

**where b.state = s.state**

**and s.userid = a.selleruserid**

**and b.userid = a.buyersuserid);**

**Result:**

**SELLERUSERID BUYERSUSERID**

**------------ ------------**

**2 992**

**6 41**

**………**

**666 353**

**668 485**

**415 rows selected.**

Table

Description automatically generated

1. Find item name along with the seller id and buyer id such that the seller has sold the item to the buyer.

**Answer:**

**select distinct name ietm\_name, selleruserid as "sold by", buyersuserid as "bought by"**

**from sales\_fact**

**left join items\_dim using (itemid);**

**--or**

**select distinct name item\_name, buyersuserid, selleruserid**

**from sales\_fact sf, items\_dim i**

**where sf.itemid = i.itemid(+);**

Table

Description automatically generated with medium confidence

1. For each seller and each item sold by the seller, find the total amount sold.

**Answer:**

**select sf.selleruserid, sf.itemid, i.name "item\_name", sum(sf.price) "total\_amount\_sold"**

**from sales\_fact sf join items\_dim i on (sf.itemid = i.itemid)**

**group by (sf.selleruserid, sf.itemid, i.name)**

**;**

**Graphical user interface, text, application

Description automatically generated**

1. Find the top seller.

**Answer:**

**select \* from**

**(select sf.selleruserid, s.fname, s.lname, sum(sf.price) "total\_amount\_sold"**

**from sales\_fact sf left join sellers\_dim s on (sf.selleruserid = s.userid)**

**group by sf.selleruserid,s.fname, s.lname**

**order by sum(price) desc)**

**where rownum = 1;**

**Graphical user interface, text, application, email

Description automatically generated**

1. Find the top buyer.

**Answer:**

**select \* from**

**(select sf.buyersuserid, b.fname, b.lname, sum(sf.price) "total\_amount\_bought"**

**from sales\_fact sf left outer join buyers\_dim b on (sf.buyersuserid = b.userid)**

**group by sf.buyersuserid, b.fname, b.lname**

**order by sum(price) desc)**

**where rownum = 1;**

Graphical user interface, text, application, email

Description automatically generated