**Q1. Write a Query to find out minimum total valuation of the product. (USE CTE)**

WITH emptotals AS (

SELECT em.employee\_id , em.employee\_name , SUM(im.item\_valuation) as total

FROM table1 JOIN table2 ON a.x = b.x

GROUP BY em.employee\_id , em.employee\_name

)

SELECT \* FROM emptotals

WHERE total = (SELECT MIN(total) FROM emptotals);

AND JUST REPLACE THIS CONCEPT FOR MAXIMUM also.

Fact1 : WHEN USING CASE do not apply commas AFTER each when..

Fact2 : If said in question if contact number not given then replace it with address , so we can use CASE to replace

CASE WHEN contact\_no is null then address ELSE TO\_CHAR(contact\_no) END AS CONTACT\_TYPE.

Fact3 : Customers irrespective of holding a library card . : SO HERE is the intuition of LEFT JOIN

Fact5 : Wheneveer you here the sentence saying like you have X but don’t have Y , X not having Y . From these you should conclude to use LEFT JOIN. AND AFTER APPLYING LEFT JOIN ALWAYS MAKE SOMETHING NULL.

**Q2. Employee who has not issued any item in the year 2013.**

SELECT Distinct em.employee\_id , em.employee\_name,

FROM table1 JOIN table2 ON …..

AND EXTRACT(YEAR FROM issue\_date) = 2013

WHERE employee\_id is null and employee\_id IN (select employee\_id FROM employee\_issue\_details).

ORDER BY em.employee\_id ASC;

**Q2. Branch having maximum customers.**

**WITH BranchAccountCounts AS (**

**SELECT**

**branch\_id,**

**COUNT(account\_number) AS customer\_count**

**FROM**

**account\_master**

**GROUP BY**

**branch\_id**

**)**

**SELECT**

**bm.branch\_name,**

**bm.branch\_city**

**FROM**

**branch\_master bm**

**JOIN**

**BranchAccountCounts bac ON bm.branch\_id = bac.branch\_id**

**WHERE**

**bac.customer\_count = (**

**SELECT MAX(customer\_count)**

**FROM BranchAccountCountsim**

**)**

**ORDER BY**

**bm.branch\_name ASC;**

**# FACT : If date thing is given In varchar and you want to compare time like between 6 am and 6 pm so do it like 06:00:00 and 18:00:00**

**# FACT : If the contact number is not availaiable then display his address .**

**SO , USE , CASE WHEN contact\_no is null THEN address ELSE to\_char(contact\_no) END AS**

**Q2. Total Average Ticket Cost**

**SELECT af.flight\_id , af.from\_location , af.to\_location , count(af.flight\_id)**

**FROM table1 JOIN table2 ON a.x = b.x**

**GROUP by flight\_id , from\_location , to\_location**

**HAVING afd.price>(SELECT AVG(Price) FROM afd)**

**ORDER BY ..**

**Q3. Accounts with withdrawal amount more than deposit amount.**

**SELECT account\_number,**

**SUM(CASE WHEN account\_type = Withdrawal THEN transaction\_amount ELSE 0) as Withdrawal,**

**Openining\_balance + SUM(CASE WHEN account\_type = Deposit THEN Transaction\_amount ELSE 0) as Deposit**

**GROUP BY ..**

**HAVING Withdrawal > Deposit**

**Q4. Delay in Flights**

**SELECT profile\_id,first\_name ,last\_name,flight\_id,flight\_departure\_date,actual\_dep\_time,**

**Actual\_arrival\_time,(departure\_time + INTERVAL ‘1’ HOUR ) as delayed\_departure\_time,**

**(arrival\_time + INTERVAL ‘1’ HOUR) AS**

**Q5. Write a query to display the flight details in which more than 10% of seats havee been booked**

**SELECT \* FROM table**

**WHERE (booked\_seats/total\_seats)>10’**

**Q6. Actors who acted in maximum number of movies.**

**WITH CountMovies AS (**

**SELECT lead\_role\_1, COUNT(\*) AS total\_movies**

**FROM movie\_master**

**GROUP BY lead\_role\_1**

**)**

**SELECT mm.movie\_id, mm.movie\_name, mm.release\_date, mm.director**

**FROM movie\_master mm**

**JOIN CountMovies cm**

**ON mm.lead\_role\_1 = cm.lead\_role\_1**

**WHERE cm.total\_movies = (**

**SELECT MAX(total\_movies) FROM CountMovies**

**)**

**ORDER BY mm.movie\_name;**

**Q7. Passenger who booked maximum number of tickets.**

**WITH CountTickets AS (**

**SELECT profile\_id , COUNT(\*) as Count**

**FROM air\_ticket\_info**

**GROUP BY profile\_id**

**)**

**SELECT profile\_id**

**FROM CountTickets**

**WHERE Count = (SELECT MAX(COUNT) FROM CountTickets)**

**ORDER BY profile\_id;**

**Q8. Customer issued with No movies then display 0.**

**SELECT c.customer\_name, COUNT(m.movie\_id) AS MOVIE\_COUNT**

**FROM customers c**

**LEFT JOIN rentals r ON c.customer\_id = r.customer\_id**

**LEFT JOIN movies m ON r.movie\_id = m.movie\_id**

**GROUP BY c.customer\_name**

**ORDER BY c.customer\_name ASC;**

**HERE COUNT(NULL) WILL BE 0 so LEFT JOIN enough.**

**OR USE COALESCE(COUNT(MOVIE\_ID),0)**

**Q9. Customer with min/Max movies**

**WITH MovieCount AS (**

**SELECT**

**c.customer\_id,**

**c.customer\_name,**

**COUNT(\*) AS total\_movies**

**FROM customer c**

**JOIN issue i**

**ON c.customer\_id = i.customer\_id**

**GROUP BY c.customer\_id, c.customer\_name**

**HAVING COUNT(\*) >= 1**

**)**

**SELECT customer\_id, customer\_name, total\_movies**

**FROM MovieCount**

**WHERE total\_movies = (SELECT MAX(total\_movies) FROM MovieCount)**

**OR total\_movies = (SELECT MIN(total\_movies) FROM MovieCount)**

**ORDER BY**

**CASE**

**WHEN total\_movies = (SELECT MAX(total\_movies) FROM MovieCount) THEN 1**

**ELSE 2**

**END,**

**customer\_name;**