Database Query Writing 1

March-25/ DBT Query Writing/1

MySQL

Diploma in Advance Computing

March 2025

|  |
| --- |
| 1. Write a SQL query to retrieve the most frequently ordered item(s) for each date from a given ord table. If multiple items have the highest order count on a particular date, include all such items in the result. |
| 1. select \* from (select order\_date, item, count(\*) 'Order Count', dense\_rank() over(partition by orderdate order by count(\*) desc) "Rank Item" from order\_items group by orderdate, item order by orderdate) T1 where `Rank Item`=1; |
| 1. with a as (select orderdate, item, count(\*) "order count", dense\_rank() over(partition by orderdate order by count(\*) desc) "Order Rank" from order\_items group by orderdate, item) select \* from a where `Order Rank`=1; |
|  |
| 1. Write a SQL query to retrieve the first and last order for each customer from the ORD table |
| 1. select custid, orderdate, total from (select custid, orderdate, total, dense\_rank() over(partition by custid order by orderdate) 'First Order', dense\_rank() over(partition by custid order by orderdate desc) 'Last Order' from ord order by custid, orderdate) T1 where `First Order` = 1 or `Last Order` = 1; |
| 1. select custid, orderdate, total, case when `First Order` = 1 then "First Order" when `Last Order`=1 then "Last Order" end "Order Type" from (select custid, orderdate, total, dense\_rank() over(partition by custid order by orderdate) 'First Order', dense\_rank() over(partition by custid order by orderdate desc) 'Last Order' from ord order by custid, orderdate) T1 where `First Order` = 1 or `Last Order` = 1; |
|  |
| 1. Write a SQL query to retrieve the first and last employee for each job from the orders table |
| 1. select \* from (select job, hiredate, dense\_rank() over(partition by job order by hiredate) as "First Employee", dense\_rank() over(partition by job order by hiredate desc) as "Last Employee" from emp group by job, hiredate order by job, hiredate) t1 where `First Employee`=1 or `Last Employee`=1; |
|  |
| 1. Given a Teams table with columns TeamID (integer) and Members (comma-separated string of names), write a query to calculate and display the total number of members in each team." |
| 1. select TeamID, members, length(members) - length(replace(members,",", "")) + 1 `Total Members` from teams; |
|  |
|  |
| 1. How many cases have reached each stage of completion (Stage 1 to Stage 5) for each centre? |
| 1. select centerID, count(COALESCE(Stage1, Stage2, Stage3, Stage4, Stage5)) Stage1, count(COALESCE(Stage2, Stage3, Stage4, Stage5)) Stage2, count(COALESCE(Stage3, Stage4, Stage5)) Stage3, count(COALESCE(Stage4, Stage5)) Stage4, count(COALESCE(Stage5)) Stage5 from stages group by centerID; |
|  |
| 1. Write SQL query to display the name of the manager where more than 4 employees are reporting to them.   Output: - 6 employees are reporting to BLAKE  8 employees are reporting to GRASS |
| 1. select concat(count(\*), " employees are reporting to ", m.ename) "R1" from emp e, emp m where e.mgr = m.empno group by m.ename having count(\*) > 4; |
|  |
| 1. Write SQL query to display all department number and employee name where either ‘M’ or ‘F’ employees are working.   Output: - +-----------+-----------+-------------------------------------------------------------+  | deptno | gender | Employee Names |  +-----------+-----------+-------------------------------------------------------------+  | 10 | M | AARAV,THOMAS,CLARK,KING,MILLER |  | 50 | F | VRUSHALI,SANGITA,SUPRIYA |  +------------+----------+--------------------------------------------------------------+ |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |