**Assignment – 6**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Create following tables for given schema. Apply constraints as mentioned in the shema.  trainer(tid PK, tname not null , cid fk, salary default 20000,experience in years check >=10 )  course(cid pk, cname unique not null, fees default 5000)   **trainer** Table   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **tid** | **tname** | **cid** | **salary** | **experience** | | t1 | Amit | c2 | Your data | Your data | | t2 | Rajan | c1 | Your data | Your data | | t3 | Shruti | c2 | Your data | Your data | | t4 | Arti | c2 | Your data | Your data | | t5 | Sameer | c3 | Your data | Your data | | t6 | Sanjay | null | Your data | Your data | | t7 | Soha | null | Your data | Your data |     **course** Table   |  |  |  | | --- | --- | --- | | **cid** | **cname** | **fees** | | c1 | Coe Java | 7000 | | c2 | Linux | 8000 | | c3 | Data Structure | 10000 | | c4 | Python | 12000 | | c5 | Dot Net | null | |
| * Check data in both the tables.   1.Try adding following data in the above table   trainer table     (t2,'Mahesh',null,16000,12)     (t8,'Meena',c3,25000,11)     (t9,null,c2,30000,10)     (t10,'Soham',c2,35000,8)     (t11,'Anil',c3,null,11)  2. Delete course c3 from course table  3. Update fees of course C5 to 10000 | |
|  | |

**Use partition by clause**

|  |  |
| --- | --- |
| 1 | Display all empno,ename,deptno,sal,avg(sal),min(sal), max(sal) |
| 2 | Write a query to display maximum salary, minimum salary and average sal along with emp name and deptno  and dname for all departments. |
| 3 | Find sum of salary for all employees whose sal > 1000. Query should display ename,sal,Sum of salary, should be displayed according to the job. |
| 4 | Find number of clerks working in each department. Display ename,job and count of that job. |
| 5 | Find how many employees are working under same manager display mgr and the count |
| 6 | Calculate how many employees earn sal greater than 1500 in each dept. Query should display ename, sal, count of employees earning sal >1500. |
| 7 | Calculate the difference between max salary and minimum salary for depatno 20.  Query should display ename,sal, diff between max sal and employee's sal, diff between min sal and max sal of the dept. |

**Mixed Queries**

|  |  |
| --- | --- |
| 1 | Display all employees who earn sal more that Smith earns. |
| 2 | Display all employees who are working in Smith's department. |
| 3 | Display all employees who earn salary < Scott and salary > Adams sal. |
| 4 | Display all employees with salary > either Ward's  salary or Blake's sal. |
| 5 | Display all employees who earn more than average sal of dept 10. |
| 6 | Display all employees who are working in research department. |
| 7 | Display   sum of salary  and min(sal) for all employees who are managers. |
| 8 | Create following tables and solve the  queries |
|  | 1. create table category(        cid int primary key,        cname varchar (15) unique not null,        description varchar (30)           );   |  |  |  | | --- | --- | --- | | **cid** | **cname** | **description** | | 1 | chips | very crunchy | | 2 | chocolate | very chocolaty | | 3 | snacks | yummy | | 4 | cold drinks | thanda thanda cool cool | |
| 2. create table Product(    pid  int primary key,    pname varchar (20),    price float (6,2) default 20.00,    qty int check(qty>0),    cid int  ,    sid int,    constraint fk\_cid foreign key(cid) references category(cid)     constraint fk\_sid foreign key(sid) references salesman(sid)      );   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **prodid** | **pname** | **qty** | **price** | **catid** | **sid** | | 123 | lays | 30 | 30.00 | 1 | 12 | | 111 | pepsi | 40 | 50.00 | 4 | 11 | | 134 | nachos | 50 | 50.00 | 1 | 12 | | 124 | dairy milk | 40 | 60.00 | 2 | 14 | | 125 | pringles | 40 | 60.00 | 1 | 14 | | 213 | Sketch Pens | 56 | 150.00 | null | 11 | | 167 | eraser | 90 | 5.00 | null | null | |
| 3. create table salesman (    sid int primary key,    sname varchar (15) unique not null,    city varchar (15) );   |  |  |  | | --- | --- | --- | | **sid** | **sname** | **city** | | 11 | Rahul | Pune | | 12 | Kirti | Mumbai | | 13 | Prasad | Nashik | | 14 | Arnav | Amravati | |
|  | 1. List all pid,pname,cid,cname  with category chips |
|  | 2. Display all products sold by kirti |
|  | 3. List all products which belong to the category of lays and sold by Arnav. Consider product table |
|  | 4. Display the count of employees working for the company. |
|  | 5. Display the count of products having price > 50 |
|  | 6. List all salesmen who stays in city where  name starts with P or N |
|  | 7. List prodid,pname,price,qty,and salesman for salesman working in Mumbai location. |