SQL Assignment

TASK -1 : Create Database: Practices

Note: Use Create SQL statement for creating table. tblProduct:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraint | Description |
| Prd\_ID | Int | Primary Key | Auto Increment |
| Prd\_Name | Varchar(25) |  |  |
| Recommended\_Price | Varchar(25) |  |  |
| Category | Varchar(25) |  |  |

tblCustomer:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraint | Description |
| CustomerID | Int | Primary Key | Auto Increment |
| FirstName | Varchar(25) |  |  |
| LastName | Varchar(25) |  |  |
| City | Varchar(25) |  |  |
| State | Varchar(25) |  |  |
| Zip | Varchar(25) |  |  |

tblSales

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraint | Description |
| Sales\_ID | Int | Primary Key | Auto Increment |
| Prd\_ID | Int | Foreign Key |  |
| Cust\_Id | Int | Foreign Key |  |
| SalesPrice | Varchar(25) |  |  |
| SalesDate | Date |  |  |

CREATE DATABASE practices

CREATE TABLE product

(

prd\_id int PRIMARY KEY AUTO\_INCREMENT,

prd\_name varchar(25),

recommended\_price varchar(25),

category varchar(25)

)

CREATE TABLE costomer

(

cst\_id int PRIMARY KEY AUTO\_INCREMENT,

firstname varchar(25),

lastname varchar(25),

city varchar(25),

state varchar(25),

zip varchar(25)

)

CREATE TABLE sales

(

sales\_id int PRIMARY KEY AUTO\_INCREMENT,

salesprice varchar(25),

salesdate date,

prd\_id\_fk int,

FOREIGN KEY(prd\_id\_fk) REFERENCES product(prd\_id),

cst\_id\_fk int,

FOREIGN KEY(cst\_id\_fk) REFERENCES costomer(cst\_id)

)

Queries :

**Return the FirstName, LastName , ProductName and Sales Price for all products sold in month of October 2022**

SELECT \* FROM sales WHERE salesdate=2022-10-23

SELECT \* FROM sales WHERE salesprice=NULL

**Return the CustomerID, FirstName, and LastName of those individuals in the Customer table who have made no Sales purchases.**

ALTER TABLE sales ADD recsalesprice varchar(30);

ALTER TABLE sales ADD salesperson varchar(30);

**Return the FirstName, LastName, SalePrice, Recommended SalePrice, and the difference between the SalePrice and Recommended SalePrice for all Sales. The difference must be returned as a positive number.**

SELECT recsalesprice,salesprice FROM sales ORDER BY ABS(recsalesprice-salesprice) DESC

**Add the following Customer and Sale information to the database. (using store procedure)**

**FirstName : Mansi**

**LastName: Joshi**

**City: Mumbai**

**State:MH**

**Zip:400001**

**ProductID:3**

**Sale Price:205**

**SaleDate:05/12/2022**

DELIMITER $$

CREATE PROCEDURE insert\_data2()

BEGIN

INSERT INTO costomer(firstname,lastname,city,state,zip) VALUES("Mansi","Joshi","Mumbai","MH",40001);

INSERT INTO sales(prd\_id,salesprice) VALUES(3,205);

END

**Return the Product Category and the average Sale Price for those customers who have purchased two or more products.**

SELECT AVG(salesprice) FROM sales

SELECT AVG (salesprice) FROM sales WHERE sales\_id IN (SELECT sales\_id FROM sales WHERE cst\_id\_fk=2)

**Update the Sale Price to the Recommended Sale Price of those Sales occurring between 6/10/2022and6/20/2022.**

UPDATE sales SET salesprice = recommended\_price FROM sales INNER JOIN product ON sales.prd\_id\_fk = product.prd\_id WHERE salesdate BETWEEN'2022-06-10' AND '2022-06-20';

**Number of Sales by Product Category where the average Recommended Priceis10 or more dollars greater than the average Sale Price.**

SELECT product.Category, COUNT(\*) AS Numsales

FROM sales

INNER JOIN product ON sales.prd\_id\_fk = product.prd\_id

GROUP BY product.Category

HAVING AVG(CAST(recsalesprice AS FLOAT)) - AVG(CAST(salesprice AS FLOAT)) >= 10;

**Largest order taken by each salesperson, date wise.**

SELECT sales.salesdate, costomer.firstname, product.prd\_name, MAX(CAST(sales.salesprice AS FLOAT)) AS LargestOrder

FROM sales

INNER JOIN product ON sales.prd\_id\_fk = product.prd\_id

INNER JOIN costomer ON sales.cst\_id\_fk = costomer.cst\_id

GROUP BY sales.salesdate, costomer.firstname, product.prd\_name;

**Without using a declared iterative construct, return Sale Date and the running total for all sales, ordered by the Sale Date in Ascending Order.**

SELECT salesdate, SUM(CAST(salesprice AS FLOAT)) OVER (ORDER BY salesdate ASC) AS RunningTotal FROM sales ORDER BY salesdate ASC;

TASK 2:

Table Name:- Employee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Empid | EmpName | Department | ContactNo | EmailId | EmpHeadId |
| 101 | Isha | E-101 | 1234567890 | [isha@gmail.com](mailto:isha@gmail.com) | 105 |
| 102 | Priya | E-104 | 1234567890 | [priya@yahoo.com](mailto:priya@yahoo.com) | 103 |
| 103 | Neha | E-101 | 1234567890 | [neha@gmail.com](mailto:neha@gmail.com) | 101 |
| 104 | Rahul | E-102 | 1234567890 | [rahul@yahoo.com](mailto:rahul@yahoo.com) | 105 |
| 105 | Abhishek | E-101 | 1234567890 | [abhishek@gmail.com](mailto:abhishek@gmail.com) | 102 |

Table Name:- EmpDept

|  |  |  |  |
| --- | --- | --- | --- |
| DeptId | DeptName | Dept\_off | DeptHead |
| E-101 | HR | Monday | 105 |
| E-102 | Development | Tuesday | 101 |
| E-103 | Hous Keeping | Saturday | 103 |
| E-104 | Sales | Sunday | 104 |
| E-105 | Purchage | Tuesday | 104 |

Table :- EmpSalary

|  |  |  |
| --- | --- | --- |
| EmpId | Salary | IsPermanent |
| 101 | 2000 | Yes |
| 102 | 10000 | Yes |
| 103 | 5000 | No |
| 104 | 1900 | Yes |
| 105 | 2300 | Yes |

Table :- Project

|  |  |
| --- | --- |
| ProjectId | Duration |
| p-1 | 23 |
| p-2 | 15 |
| p-3 | 45 |
| p-4 | 2 |
| p-5 | 30 |

Table :- Country

|  |  |
| --- | --- |
| cid | cname |
| c-1 | India |
| c-2 | USA |
| c-3 | China |
| c-4 | Pakistan |
| c-5 | Russia |

Table :- ClientTable

|  |  |  |
| --- | --- | --- |
| ClientId | ClientName | cid |
| cl-1 | ABC Group | c-1 |
| cl-2 | PQR | c-1 |
| cl-3 | XYZ | c-2 |
| cl-4 | tech altum | c-3 |
| cl-5 | mnp | c-5 |

Table :- EmpProject

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EmpId | ProjectId | ClientID | StartYear | EndYear |
| 101 | p-1 | Cl-1 | 2010 | 2010 |
| 102 | p-2 | Cl-2 | 2010 | 2012 |
| 103 | p-1 | Cl-3 | 2013 |  |
| 104 | p-4 | Cl-1 | 2014 | 2015 |
| 105 | p-4 | Cl-5 | 2015 |  |

CREATE DATABASE assignmenttask2

CREATE TABLE Employee

(

Empid int PRIMARY KEY AUTO\_INCREMENT,

EmpName varchar(25),

Department varchar(25),

ContactNo int,

EmailId varchar(25),

EmpHeadId int

)

CREATE TABLE EmpDept

(

DeptId int PRIMARY KEY,

DeptName varchar(25),

Dept\_off varchar(25),

DeptHead int

)

CREATE TABLE EmpSalary

(

EmpId int,

Salary int,

IsPermanent varchar(25),

FOREIGN KEY(EmpId) REFERENCES employee(Empid)

)

CREATE TABLE Project

(

ProjectId varchar(25),

Duration int

)

CREATE TABLE Country

(

cid varchar(25),

cname varchar(25)

)

CREATE TABLE ClientTable

(

ClientId varchar(25),

ClientName varchar(25),

cid\_fk varchar(25),

FOREIGN KEY(cid\_fk) REFERENCES country(cid)

)

CREATE TABLE EmpProject

(

EmpId\_fk int,

ProjectId\_fk varchar(25),

ClientID varchar(25),

StartYear int,

EndYear int,

FOREIGN KEY(EmpId\_fk) REFERENCES employee(Empid),

FOREIGN KEY(ProjectId\_fk) REFERENCES Project(ProjectId),

FOREIGN KEY(ClientID\_fk) REFERENCES ClientTable(ClientId)

)

CREATE TABLE EmpProject

(

EmpId\_fk int,

ProjectId\_fk varchar(25),

ClientID\_fk varchar(25),

StartYear int,

EndYear int,

FOREIGN KEY(EmpId\_fk) REFERENCES employee(Empid),

FOREIGN KEY(ProjectId\_fk) REFERENCES Project(ProjectId),

FOREIGN KEY(ClientID\_fk) REFERENCES ClientTable(ClientId)

)

**Queries :**

**Select the detail of the employee whose name start with P.**

SELECT \* FROM Employee WHERE EmpName LIKE 'P%';

**How many permanent candidate take salary more than 5000.**

SELECT \* FROM Employee WHERE EmpName LIKE 'P%';

**Select the detail of employee whose emailed is in Gmail.**

SELECT \* FROM Employee WHERE EmailId LIKE '%@gmail.com';

**Select the details of the employee who work either for department E-104 or E-102.**

SELECT \* FROM Employee WHERE Department IN ('E-104', 'E-102');

**What is the department name for DeptID E-102?**

SELECT DeptName FROM EmpDept WHERE DeptId = 'E-102';

**What is total salary that is paid to permanent employees?**

SELECT SUM(Salary) FROM EmpSalary WHERE IsPermanent = 'Yes';

**List name of all employees whose name ends with a.**

SELECT EmpName FROM Employee WHERE EmpName LIKE '%a';

**List the number of department of employees in each project.**

SELECT ProjectId, COUNT(DISTINCT Department) as NumDept FROM EmpProject GROUP BY ProjectId;

**How many project started in year 2010.**

SELECT COUNT(\*) FROM EmpProject WHERE StartYear = 2010;

**How many project started and finished in the same year.**

SELECT COUNT(\*) FROM EmpProject WHERE StartYear = EndYear;

**Select the name of the employee whose name's 3rd character is 'h'.**

SELECT EmpName FROM Employee WHERE SUBSTRING(EmpName, 3, 1) = 'h';

**Select the department name of the company which is assigned to the employee whose employee id is grater 103.**

SELECT DeptName FROM EmpDept WHERE DeptId IN (SELECT Department FROM Employee WHERE Empid > 103);

**Select the name of the employee who is working under Abhishek.**

SELECT EmpName FROM Employee WHERE EmpHeadId = (SELECT Empid FROM Employee WHERE EmpName = 'Abhishek');

**Select the name of the employee who is department head of HR.**

SELECT EmpName FROM Employee WHERE Empid = (SELECT DeptHead FROM EmpDept WHERE DeptName = 'HR');

**Select the name of the employee head who is permanent.**

SELECT EmpName FROM Employee WHERE Empid IN (SELECT DeptHead FROM EmpDept WHERE DeptHead IN (SELECT Empid FROM Employee WHERE IsPermanent = 'Yes'));

**Select the name and email of the Dept Head who is not Permanent.**

SELECT EmpName, EmailId FROM Employee WHERE Empid IN (SELECT DeptHead FROM EmpDept WHERE DeptHead IN (SELECT Empid FROM Employee WHERE IsPermanent = 'No'));

**Select the employee whose department off is Monday**

SELECT \* FROM Employee WHERE Department IN (SELECT DeptId FROM EmpDept WHERE Dept\_off = 'Monday');

**Select the Indian clients details.**

SELECT \* FROM ClientTable WHERE cid = 'c-1';

**Select the details of all employee working in development department.**

SELECT \* FROM Employee WHERE Department = 'E-102';