Assignment – 3

1. Import Job\_History.csv Departments.csv Employee\_A3.csv

After Importing do the following :

a. Write a nested query in SQL to display the details of those departments in which max salary is 7000 or above

for those employees who have already done one or more jobs.

ANSWER

SELECT EMPLOYEE\_ID,FIRST\_NAME,JOB\_ID,SALARY,DEPARTMENT\_ID

FROM Employee

WHERE SALARY>=7000

ORDER BY SALARY;

b. Write a nested query to display all the information of the employees who do not work in those departments

where manager id within the range 100 and 200.

ANSWER

SELECT EMPLOYEE\_ID,FIRST\_NAME,JOB\_ID,SALARY,MANAGER\_ID,DEPARTMENT\_ID

FROM Employee

WHERE MANAGER\_ID BETWEEN 100 AND 200;

c. Write a nested query to identify all the employees who earn more than the average and who work in any of the IT departments

ANSWER

SELECT EMPLOYEE\_ID,FIRST\_NAME,JOB\_ID,SALARY,MANAGER\_ID,DEPARTMENT\_ID

FROM Employee

WHERE JOB\_ID LIKE 'IT%' AND SALARY > (SELECT AVG(SALARY) FROM Employee)

d. Write a nested query in SQL to display all the information of

those employees who did not have any job in the past

SELECT\*FROM Job\_History;

SELECT EMPLOYEE\_ID,YEAR(START\_DATE) AS START\_DATE

INTO START\_DATE

FROM Job\_History

SELECT EMPLOYEE\_ID,YEAR(HIRE\_DATE) AS HIRE\_DATE

INTO HIRE\_DATE

FROM EMPLOYEE;

SELECT START\_DATE.EMPLOYEE\_ID,START\_DATE,HIRE\_DATE

FROM START\_DATE

INNER JOIN HIRE\_DATE

ON START\_DATE.EMPLOYEE\_ID=HIRE\_DATE.EMPLOYEE\_ID

WHERE START\_DATE.START\_DATE=HIRE\_DATE.HIRE\_DATE

2. Analyse 3 tables below(Customer sales ,store and product data) and write a nested query to

find out the name of the most popular item in each of the 2 cities : -

GS SQL Test Ques

SELECT Book1.Cutomer\_Name,Book1.Item\_Code,Book2.Store\_id,Book2.Store\_Location

/\*INTO BOOK\_12\*/

FROM Book1

INNER JOIN Book2

ON Book1.Store\_id=Book2.Store\_id;

SELECT BOOK\_12.Cutomer\_Name,BOOK\_12.Item\_Code,Book3.Item\_Name,BOOK\_12.Store\_id,BOOK\_12.Store\_Location

INTO BOOK123

FROM BOOK\_12

INNER JOIN Book3

ON BOOK\_12.Item\_Code=Book3.Item\_Code;

Select

Product

,Case when Product in ('Coca Cola') then 1

when Product in ('Deodrant') then 2

when Product in ('Cooker') then 3

else 0 end as ind\_misc

from Data\_Set;

SELECT Cutomer\_Name,Item\_Name,Store\_Location,

CASE WHEN Store\_Location in ('Delhi') THEN 1

WHEN Store\_Location IN ('Bangalore') THEN 2

ELSE 0 END AS CITY

INTO CITY

FROM BOOK123;

SELECT Item\_Name,COUNT(\*) AS DELHI

FROM CITY

WHERE CITY =1

GROUP BY Item\_Name

SELECT Item\_Name,COUNT(\*) AS BANGLORE

FROM CITY

WHERE CITY =2

GROUP BY Item\_Name;

3. Do Assignment 2 - Question 4 using Partitions instead of using self-join

SELECT Customer\_ID,MONTH,sales,SUM(Sales) OVER(ORDER BY MONTH) AS TOTAL\_SALES\_BY\_MONTH

FROM QUESTION4

4. Use the 4 datasets created in Assignment 2 – Question 3 and calculate the following :

Name of the 2nd most popular product for each country in terms of Quantity Consumed

SELECT Orders.OrderID,Orders.ProductID,Orders.Quantity,Order\_details.CustomerID

INTO order\_order\_detail

FROM Orders

LEFT JOIN Order\_details

ON Orders.OrderID=Order\_details.OrderID

SELECT\*FROM order\_order\_detail

SELECT \*FROM Customers

SELECT order\_order\_detail.OrderID,order\_order\_detail.ProductID,order\_order\_detail.Quantity,order\_order\_detail.CustomerID,Customers.CustomerName,Customers.City,Customers.Country

/\*INTO Customer\_Order\*/

FROM order\_order\_detail

LEFT JOIN Customers

ON order\_order\_detail.CustomerID=Customers.CustomerID

SELECT Customer\_Order.OrderID,Customer\_Order.ProductID,Customer\_Order.Quantity,Customer\_Order.CustomerID,Product\_Details.ProductName,Customer\_Order.City

INTO Customer\_product\_Order1

FROM Customer\_Order

LEFT JOIN Product\_Details

ON Customer\_Order.ProductID =Product\_Details.ProductID;

SELECT \* FROM Customer\_product\_Order1;

SELECT City,COUNT(\*) AS COUNT\_CITY

FROM Customer\_product\_Order1

GROUP BY City

5. Create a function/Procedure to filter the orders from Orders table where sum of price of all products < @x where @x is the input provided by the user.

SELECT SUM(Price)

FROM Product\_Details