SQL Exercise

Given the following three tables and their instances, provide the *SQL statements* for questions (a) to (e)

SALESPERSON (Name, Age, Salary)
ORDERS (Number, CustName, SalespersonName, Amount)
CUSTOMER (Name, City, IndustryType)

SALESPERSON

SName	*Age*	*Salary*
Abel	63	120,000
Baker	38	42,000
Jones	26	36,000
Murphy	42	50,000
Zenith	59	118,000
Kobad	27	34,000

CUSTOMER

CName	*City*	*IndustryType*
Perth Construction	Perth	B
Sydney Lumber	Sydney	F
Tri-City Builders	Melbourne	B
Melbourne Housing	Melbourne	C

ORDERS

Number	*CustName*	*SPName*	*Amount*
100	Perth Construction	Zenith	560
200	Perth Construction	Jones	1,800
300	Sydney Lumber	Abel	480
400	Melbourne Housing	Abel	2,500
500	Perth Construction	Murphy	6,000
600	Tri-City Builders	Abel	700
700	Sydney Lumber	Jones	150

(a) Show the name of all salespersons under 30 years old.
(b) Compute the number of orders of each salesperson.
(c) Compute the number of orders for each salesperson, considering only orders for an amount exceeding 500.
(d) Compute the number of different customers who have an order.
(e) Show the age of salespersons who have an order with customer in "Melbourne".

Sample Solution

Query a: Show the name of all salespersons under 30 years old.

SELECT SName

FROM SALESPERSON

WHERE Age < 30;

```
SQL> SELECT SName
  2  FROM SALESPERSON
  3  WHERE Age < 30;

SNAME
---------------
Jones
Kobad
```

Query b: Compute the number of orders of each salesperson.

SELECT SPName, COUNT (*)

FROM ORDERS

GROUP BY SPName;

```
SQL> SELECT SPName, COUNT(*)
  2  FROM ORDERS
  3  GROUP BY SPName;

SPNAME          COUNT(*)
--------------- ----------
Jones                    2
Abel                     3
Murphy                   1
Zenith                   1
```

Query c: Compute the number of orders for each salesperson, considering only orders for an amount exceeding 500.

SELECT SPName, COUNT (*) AS BigOrder

FROM ORDERS

WHERE Amount > 500

GROUP BY SPName;

```
SQL> SELECT SPName, COUNT(*) AS BigOrder
  2  FROM ORDERS
  3  WHERE Amount > 500
  4  GROUP BY SPName;

SPNAME            BIGORDER
--------------- ----------
Jones                    1
Abel                     2
Murphy                   1
Zenith                   1
```

Query d: Compute the number of different customers who have an order.

SELECT Count(*) AS NumberofOrders

FROM CUSTOMER

WHERE CName IN

(SELECT CustName FROM ORDERS);

OR

SELECT Count(DISTINCT CustName)

FROM ORDERS;

```
SQL> SELECT COUNT(*)
  2  FROM CUSTOMER
  3  WHERE CName IN
  4   (SELECT CustName FROM ORDERS);

  COUNT(*)
----------
         4
```

Query e: Show the age of salespersons who have an order with customer in "Melbourne".

SELECT S.Age

FROM SALESPERSON S, ORDERS O, CUSTOMER C

WHERE S.SName = O.SPName

AND O.CustName = C.CName

AND C.City = 'Melbourne'

```
SQL> SELECT S.Age
  2  FROM SALESPERSON S, ORDERS O, CUSTOMER C
  3  WHERE S.SName = O.SPName
  4  AND O.CustName = C.CName
  5  AND C.City = 'Melbourne';

       AGE
----------
        63
        63
```

SQL Exercise 2019 (from past tests)

Question 2 – SQL Query SELECT with IN/NOT IN: Given the following tables

EMPLOYEES (**EmployeeID,** Name, Salary, Age, HighestDegree, *DepartmentID*)

DEPARTMENT (**DepartmentID**, *ManagerID*, Location, DeptDescription)

Assume that managers are also employees, and therefore the domain value of ManagerID is the same as EmployeeID (ie. ManagerID is a FK to EmployeeID). Every department has exactly one manager, and that every manager manages exactly one department.

Choose the query that produces *a list of names of all non-managers who earn more than at least one manager*.

(a)
```
SELECT E.Name
FROM Employees E, Department D
WHERE E.EmployeeID <> D. ManagerID
AND E.Salary >
      (SELECT MIN(E.Salary)
       FROM Employees E, Department D
       WHERE E.EmployeeID = D.ManagerID);
```

(b)
```
SELECT E.Name
FROM Employees E, Department D
WHERE E.EmployeeID <> D. ManagerID
AND E.Salary >
      (SELECT MIN(E.Salary)
       FROM Employees E);
```

(c)
```
SELECT E.Name
FROM Employees E
WHERE E.EmployeeID NOT IN
      (SELECT ManagerID
       FROM Department)
AND E.Salary >
      (SELECT MIN(E1.Salary)
       FROM Employees E1, Department D
       WHERE E1.EmployeeID = D.ManagerID);
```

(d)
```
SELECT E.Name
FROM Employees E
WHERE E.EmployeeID NOT IN
      (SELECT ManagerID
       FROM Department)
AND E.Salary >
      (SELECT MIN(E.Salary)
       FROM Employee E);
```

Question 3 – SQL Query SELECT with SUM: The following tables and instances show a snapshot of an investment portfolio database:

MEMBER

M_No	M_name	Street	Town	Post_code	Email	Balance
1066	Ben King	Kings St	Bundoora	3083	bs@bs.com	45000
13144	Tomy Page	High St	Clayton	3800	tar@tar.com	30000
1776	Harry Shine	Elphin Rd	Hawthorn	3122	hs@hs.com	50000
2001	Karl May	Burke Rd	Kew	3101	km@km.com	20000
2002	Quy Smith	Glen Rd	Kew	3101	qs@qs.com	10000

INVESTMENT

Symbol	Company_Name	Current_Price	Asking_Price	Bid_Price
AMZN	Amazon.com	119.06	119.18	119.06
EBAY	eBay Inc	174.00	174.25	174.00
KLM	KLM Royal Dutch Air	29.93		
MSFT	Microsoft Corp	78.50	78.56	78.50
ORCL	Oracle Corp	23.25	23.25	23.18

TRANSACTION

M_No	Symbol	TDate	TType	QTY	TPrice	Commission
1776	ORCL	20-04-2008	buy	100	23.18	1%
1776	EBAY	20-03-2008	buy	100	174.00	1%
1066	ORCL	09-01-2008	buy	50	23.25	1%
1066	ORCL	14-01-2008	sell	20	23.30	1%

Choose the correct SQL statement to display the following query:

For each member, display the total 'buy' transaction price and the total 'sell' transaction price (note: transaction price = TPrice x QTY)

(a)
```
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
GROUP BY T.M_No, T.TType;
```

(b)
```
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
WHERE T.TType = 'buy'
UNION
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
WHERE T.TType = 'sell';
```

(c)
```
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
WHERE T.TType = 'buy' OR T.TType = 'sell';
```

(d)
```
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
```

```
WHERE T.TType = 'buy'
GROUP BY T.TType
UNION
SELECT T.M_No, T.TType, sum(T.TPrice*T.Qty) as TotalTrans
FROM Transaction T
WHERE T.TType = 'sell'
GROUP BY TType;
```