**WORKSHEET 4 SQL**

1. Write a SQL query to show average number of orders shipped in a day (use Orders table).

ANS: SELECT AVG (orders\_shipped)

FROM orders;

2. Write a SQL query to show average number of orders placed in a day.

ANS: SELECT date(order\_placed\_date), COUNT(order\_id) AS num\_orders,

SUM(order\_total) AS daily\_total

FROM orders

WHERE order\_placed\_date>=date\_sub(current\_date, INTERVAL 31 DAY)

GROUP BY date(order\_placed\_date)

SELECT date(order\_placed\_date)

, COUNT(id) AS num\_orders

, SUM(order\_total) AS daily\_total

FROM [Table]

GROUP BY date(order\_placed\_date)

SELECT date(order\_placed\_date)

, COUNT(id) AS num\_orders

, SUM(order\_total) AS daily\_total

FROM orders

GROUP BY 1

SELECT order\_placed\_date, SUM(order\_total)

FROM orders

GROUP BY order\_placed\_date

SELECT order\_placed\_date, SUM(order\_total)

FROM orders

GROUP BY order\_placed\_date

3. Write a SQL query to show the product name with minimum MSRP (use Products table).

ANS: select p.name, min (c.price)

from product p, catalog c, producer prc

where c.pp\_product\_id = p.product\_id and prc.producer\_id =

c.pp\_producer\_id

group by p.name;

select \*

from (select p.name as product\_name, p.type, prod.name as producer\_name,

price,

ROW\_NUMBER() over (partition by p.name order by price ) as seqnum

from catalog c join

product p

on c.pp\_product\_id = p.product\_id join

producer prod

on c.pp\_producer\_id = prod.producer\_id

) t

where seqnum = 1;

select \*

from (select p.name as product\_name, p.type, prod.name as producer\_name,

price,

min(price) over (partition by p.name) as minprice

from catalog c join

product p

on c.pp\_product\_id = p.product\_id join

producer prod

on c.pp\_producer\_id = prod.producer\_id

) t

where price = minprice;

SELECT p.name, pr.name, c2.minprice

FROM product p, producer pr, catalog c,

(SELECT pp\_product\_id, MIN(price) AS minprice

FROM catalog c1

GROUP BY pp\_product\_id

) c2

WHERE p.product\_id=c.pp\_product\_id

AND pr.producer\_id=c.pp\_producer\_id

AND p.product\_id =c2.pp\_product\_id

AND c.price =c2.minprice;

4. Write a SQL query to show the product name with maximum value of stockQuantity.

ANS: SELECT MAX(product\_quantity) FROM

(SELECT COUNT(quantity) as product\_quantity FROM

sales\_products) as derived;

SELECT product\_id, MAX(product\_quantity) FROM

(SELECT COUNT(quantity) as product\_quantity FROM

sales\_products) as derived;

SELECT product\_id, COUNT(quantity) as product\_quantity

FROM sales\_products

GROUP BY product\_id

HAVING COUNT(quantity) >= ALL(

SELECT COUNT(quantity)

FROM sales\_products

GROUP BY product\_id

)

SELECT product\_id, SUM(quantity) as product\_quantity

FROM sales\_products

GROUP BY product\_id

HAVING SUM(quantity) >= ALL(

SELECT SUM(quantity)

FROM sales\_products

GROUP BY product\_id

)

SELECT product\_id, SUM(quantity) as product\_quantity

FROM sales\_products

GROUP BY product\_id

HAVING SUM(quantity) >= (

SELECT MIN(quantity\_sum)

FROM

(

SELECT SUM(quantity) quantity\_sum

FROM sales\_products

GROUP BY product\_id

ORDER BY SUM(quantity) desc

LIMIT 5

) t

)

SELECT product\_id, SUM(quantity) as product\_quantity

FROM sales\_products

GROUP BY product\_id

ORDER BY SUM(quantity) desc

LIMIT 5

5. Write a query to show the most ordered product Name (the product with maximum number of orders).

ANS: SELECT p.`product\_id`, p.`name`, SUM(o.`quantity`) AS quantity

FROM `Order\_Detail` AS o

INNER JOIN `Product` AS p

ON o.`product\_id` = p.`product\_id`

GROUP BY o.`product\_id`

ORDER BY SUM(o.`quantity`) DESC, p.`name` ASC

LIMIT 3

6. Write a SQL query to show the highest paying customer Name.

ANS: Select name

from table

where salary = (select max(salary) from table)

SELECT top 1 Name

FROM tableName

ORDER BY Salary DESC

select name

from table

order by salary desc

fetch first 1 rows only

select name from (

select name, row\_number() over (order by salary desc) as rn

from table

) where rn = 1

SELECT TOP name

FROM yourtable

WHERE salary = (SELECT MAX(salary) FROM yourtable)

select Name, Salary from tb\_name where salary=(select max(salary) from tb\_name);

7. Write a SQL query to show cutomerNumber, customerName of all the customers who are from Melbourne city.

ANS: SELECT o.orderNumber

FROM orders o

LEFT JOIN customers c

ON c.customerNumber = o.customerNumber

WHERE c.customerName LIKE 'N%'

SELECT cust\_name,city,grade

FROM customer

ORDER BY customer\_id;

8. Write a SQL query to show name of all the customers whose name start with “N”

ANS: SELECT o.orderNumber

FROM orders o

LEFT JOIN customers c

ON c.customerNumber = o.customerNumber

WHERE c.customerName LIKE 'N%'

SET @Initial = 'N' --Initial to search

SET @FieldChoice = 1 -- {1,2,3} = {customerName,contactFirstName,contactLastName}

/\*Your query\*/

SELECT O.orderNumber

FROM orders O

INNER JOIN customers C

ON C.customerNumber = O.customerNumber

WHERE (LEFT(C.customerName,1) IN (@Initial) AND @FieldChoice = 1)

OR (LEFT(C.contactFirstName,1) IN (@Initial) AND @FieldChoice = 2)

OR (LEFT(C.contactLastName,1) IN (@Initial) AND @FieldChoice = 3)

DECLARE @Initial varchar(1) --Declaring our Initial

SET @Initial = 'N' --Initial to search

/\*Your query\*/

SELECT O.orderNumber

FROM orders O

INNER JOIN customers C

ON C.customerNumber = O.customerNumber

AND (LEFT(C.contactLastName,1) IN (@Initial)

9. Write a SQL query to show name of all the customers whose phone start with ‘7’ and are from city ‘Las Vegas’.

ANS: SELECT o.orderNumber

FROM orders o

LEFT JOIN customers c

ON c.customerNumber = o.customerNumber

WHERE c.customerName LIKE '7'

SET @Initial = '7' --Initial to search

SET @FieldChoice = 1 -- {1,2,3} = {customerName,contactFirstName,contactLastName}

/\*Your query\*/

SELECT O.orderNumber

FROM orders O

INNER JOIN customers C

ON C.customerNumber = O.customerNumber

WHERE (LEFT(C.cityName,1) IN (@Las Vegas) AND @FieldChoice = 1)

OR (LEFT(C.cityFirstName,1) IN (@Initial) AND @FieldChoice = 2)

OR (LEFT(C.cityName,1) IN (@Initial) AND @FieldChoice = 3)

DECLARE @Initial varchar(1) --Declaring our Initial

SET @Initial = '7' --Initial to search

/\*Your query\*/

SELECT O.orderNumber

FROM orders O

INNER JOIN customers C

ON C.customerNumber = O.customerNumber

AND (LEFT(C.contactLastName,1) IN (@Initial

10. Write a SQL query to show name of all the customers whose creditLimit < 1000 and city is either “Las Vegas” or ”Nantes” or “Stavern”.

ANS: SQL code used for analysis:

i. Group 1

SELECT

b.stars,

b.city,

substr(h.hours,-11) hours, -- remove the day from hours column

count(b.id) number\_of\_businesses

FROM business as b

INNER JOIN hours h ON b.id=h.business\_id -- join to fetch data from hours and business table

GROUP BY b.stars,b.city,substr(h.hours,-11)

HAVING city='Las Vegas' AND b.stars BETWEEN 2.0 AND 3.0

ORDER BY b.stars;

Group 2

SELECT

b.stars,

b.city,

substr(h.hours,-11) hours, -- remove the day from hours column

count(b.id) number\_of\_businesses

FROM business as b

INNER JOIN hours h ON b.id=h.business\_id

GROUP BY b.stars,b.city,substr(h.hours,-11)

HAVING city='Las Vegas' AND b.stars BETWEEN 4.0 AND 5.0

ORDER BY b.stars;

ii.

Group 1

SELECT stars,

city,

sum(review\_count) as n\_reviews

FROM business

GROUP BY stars,city

HAVING city='Las Vegas' AND stars BETWEEN 2.0 AND 3.0

Group 2:

SELECT stars,

city,

sum(review\_count) as n\_reviews

FROM business

GROUP BY stars,city

HAVING city='Las Vegas' AND stars BETWEEN 4.0 AND 5.0

11. Write a SQL query to show all the orderNumber in which quantity ordered <10.

Ans: SELECT CID, Count(Order.OrderID) AS TotalOrders

FROM [Order]

Where CID = CID

GROUP BY CID

Order BY Count(Order.OrderID) DESC;

SELECT CID, Sum(OrderItem.Quantity\*OrderItem.SalePrice) AS

TotalDollarAmount

FROM OrderItem, [Order]

WHERE OrderItem.OrderID = [Order].OrderID

GROUP BY CID

SELECT CID,

Count(DISTINCT O.OrderID) AS TotalOrders,

Sum(OI.Quantity\*OI.SalePrice) AS TotalDollarAmount

FROM [Order] O

INNER JOIN [OrderItem] OI

ON O.OrderID = OI.OrderID

GROUP BY CID

Order BY Count(DISTINCT O.OrderID) DESC

SELECT CID,

COUNT(Orders.OrderID) AS TotalOrders,

SUM(OrderAmounts.DollarAmount) AS TotalDollarAmount

FROM [Orders]

INNER JOIN (SELECT OrderID, Sum(Quantity\*SalePrice) AS

DollarAmount

FROM OrderItems GROUP BY OrderID) AS OrderAmounts

ON Orders.OrderID = OrderAmounts.OrderID

GROUP BY CID

ORDER BY Count(Orders.OrderID) DESC

12. Write a SQL query to show all the orderNumber whose customer Name start with letter ‘N’.

ANS: /\*Declares your variables\*/

DECLARE @Initial varchar(1) --Declaring our Initial

SET @Initial = 'N' --Initial to search

/\*Your query\*/

SELECT O.orderNumber

FROM orders O

INNER JOIN customers C

ON C.customerNumber = O.customerNumber

AND (LEFT(C.contactLastName,1) IN (@Initial)

13. Write a SQL query to show all the customerName whose orders are “Disputed” in status.

ANS: select customernumber,

max(case when rn = 1 then status end) as last\_order\_status,

sum(status = 'Cancelled') cnt\_cancelled,

sum(status = 'Disputed' ) cnt\_disputed,

sum(status = 'On Hold' ) cnt\_on\_hold

from (

select o.\*,

row\_number() over(partition by customernumber order by orderdate desc) rn

from orders o

) t

group by customernumber

select customernumber,

(select o1.status from orders o1 where o1.customernumber = o.customernumber order by o1.orderdate desc limit 1) as last\_order\_status,

sum(status = 'Cancelled') cnt\_cancelled,

sum(status = 'Disputed' ) cnt\_disputed,

sum(status = 'On Hold' ) cnt\_on\_hold

from orders o

group by customernumber

14. Write a SQL query to show the customerName who made payment through cheque with checkNumber starting with H and made payment on “2004-10-19”.

ANS: foo=> SET TIMEZONE TO 'Japan';

SET

foo=> SELECT '2011-01-01 00:00:00'::TIMESTAMP;

timestamp

2011-01-01 00:00:00

foo=> SELECT '2011-01-01 00:00:00'::TIMESTAMP WITH TIME ZONE;

timestamptz

2011-01-01 00:00:00+09

foo=> SELECT '2011-01-01 00:00:00+03'::TIMESTAMP;

timestamp

2011-01-01 00:00:00

foo=> SELECT '2011-01-01 00:00:00+03'::TIMESTAMP WITH TIME ZONE;

timestamptz

2011-01-01 06:00:00+09

foo=> SET TIMEZONE TO 'Australia/Melbourne';

SET

foo=> SELECT '2011-01-01 00:00:00'::TIMESTAMP;

timestamp

2011-01-01 00:00:00

foo=> SELECT '2011-01-01 00:00:00'::TIMESTAMP WITH TIME ZONE;

timestamptz

2011-01-01 00:00:00+11

foo=> SELECT '2011-01-01 00:00:00+03'::TIMESTAMP;

timestamp

2011-01-01 00:00:00

foo=> SELECT '2011-01-01 00:00:00+03'::TIMESTAMP WITH TIME ZONE;

timestamptz

2011-01-01 08:00:00+11

15. Write a SQL query to show all the checkNumber whose amount > 1000.

ANS: SELECT customer\_id,MAX(purch\_amt)

FROM orders

WHERE customer\_id BETWEEN 3002 and 3007

GROUP BY customer\_id

HAVING MAX(purch\_amt)>1000;