

LAB ASSIGNMENT - 5

1. SELECT

```
    UPPER ('sumit sagar') AS UPPER_CASE,  
    LOWER ('SUMIT SAGAR') AS LOWER_CASE,  
    LENGTH ('sumit sagar') AS LENGTH,  
    SUBSTR ('sumit sagar', 1, 6) AS SUBSTRING,  
    REPLACE ('sumit sagar', 'sumit', 'harsh sagar')  
        AS REPLACE_NAME,  
    TRIM (' sumit sagar ') AS TRIMSTR,  
    CONCAT ('sumit', 'sagar') AS CONCAT_STR  
FROM dual;
```

2. CREATE TABLE students5 (

```
    Roll NUMBER,  
    F_Name VARCHAR2(50),  
    M_Name VARCHAR2(50),  
    L_Name VARCHAR2(50),  
    Sec CHAR(1),  
    City VARCHAR2(50),  
    Area VARCHAR2(50),  
    HouseNo VARCHAR2(50),  
    Div Number
```

```
);
```

- (i) ~~SELECT Roll, F_Name || ' ' || M_Name || ' ' ||~~
~~L_Name AS Full_Name FROM students5;~~
- (ii) ~~SELECT Roll, F_Name || ' ' || M_Name || ' ' ||~~
~~L_Name AS Full_Name, Area || ', ' || HouseNo ||~~
~~' , ' || city AS Address FROM students5;~~

- (iii) SELECT Roll, SUBSTR (F-Name, 1, 1) || '.' ||
SUBSTR (M-Name, 1, 1) || '.' || L-Name AS Name
FROM Students5;
- (iv) SELECT *
FROM Students5
ORDER BY Sec; ✓
- (v) SELECT *
FROM Students5
ORDER BY Sec Desc; ✓
- (vi) SELECT *
FROM Students5
ORDER BY Sec, F-Name; ✓
- (vii) SELECT Roll, F-Name || ' ' || M-Name || ' ' ||
L-Name AS Name, LPAD (City, LENGTH (City)
+ 5, '*') AS City FROM Students5;
- (viii) SELECT *
FROM Students5
WHERE Div IS NULL; ✓
- (ix) SELECT DISTINCT City
FROM Students5; ✓
- (x) SELECT Roll, F-Name,
CASE
. When Div = 1 THEN 'FIRST'
. When Div = 2 THEN 'SECOND'
. When Div = 3 THEN 'THIRD'
. When Div = 0 THEN 'FAIL'
ELSE 'NOT AWARDED'
END AS Division
FROM Students5;

(xi) ~~NOT~~ SELECT Roll, F_Name ,

CASE

WHEN Sec = 'A' THEN 1

WHEN Sec = 'B' THEN 2

END AS Section

FROM students;

3) SELECT Empcode ,

TRIM (TRANSLATE (Empname, '0123456789',

'')) AS Empname

FROM employees;

4) SELECT SUBSTR (First_Name, 1, 1) || '.' ||

SUBSTR (Middle_Name, 1, 1) || '.' ||

Last_Name AS Name

FROM names;

5) SELECT Sname ,

CASE

WHEN Marks >= 35 THEN 'Pass'

ELSE 'Fail'

END AS Result

FROM studentsmarks;

Dilip
10/6/2024

Lab Assignment 6

(i) CREATE TABLE CLIENT-MASTER (

Client-No VARCHAR2(6) PRIMARY KEY CHECK
(Client-No LIKE 'C%'),

Name VARCHAR2(20) NOT NULL,

City VARCHAR2(20) CHECK (CITY IN ('Delhi',
'Mumbai', 'Chennai')),

Pincode NUMBER(6),

State VARCHAR2(20),

BalDue NUMBER(10,2),

Email VARCHAR2(30) UNIQUE

);

(ii) CREATE TABLE PRODUCT-MASTER (

Product-No VARCHAR2(6) PRIMARY KEY

Product-No CHECK (Product-No LIKE 'P%'),

Name VARCHAR2(20) NOT NULL,

UnitMeasure VARCHAR2(10) NOT NULL,

QtyOnHand NUMBER(8) NOT NULL CHECK
(QtyOnHand >= ReorderLevel),

ReorderLevel NUMBER(8) NOT NULL,

SellPrice NUMBER(8,2) NOT NULL CHECK
(SellPrice >0),

CostPrice NUMBER(8,2) NOT NULL CHECK
(CostPrice >0)

);

(iii) CREATE TABLE SALESMAN-MASTER (

Salesman-No VARCHAR2(6) PRIMARY KEY
CHECK (Salesman-No LIKE 'S%.'), ✓

Sal-name VARCHAR2(20), ✓

Address VARCHAR2(20), ✓

City VARCHAR2(20), ✓

State VARCHAR2(20), ✓

Sal-amount NUMBER(8,2) NOT NULL CHECK
(Sal-amount > 0), ✓

Tgt-to-get NUMBER(8,2) NOT NULL
CHECK (Tgt-to-get > 0), ✓

Remarks VARCHAR2(30) ✓

);

(iv) CREATE TABLE SALES-ORDER (

Order-No VARCHAR2(6) PRIMARY KEY
CHECK (Order-No Like 'O%.'), ✓

Order-Date Date NOT NULL, ✓

Client-No VARCHAR2(6) REFERENCES
CLIENT-MASTER (Client-No), ✓

Dely-add VARCHAR2(25), ✓

Salesman-No VARCHAR2(6) REFERENCES
SALESMAN-MASTER (Salesman-No), ✓

Dely-type CHAR(1) DEFAULT 'F' CHECK
(Dely-type IN ('P', 'F')), ✓

Dely-Date DATE CHECK (Dely-Date >= Order-
Date)); ✓

(v) CREATE TABLE SALES-ORDER-DETAILS(
✓ Order-No VARCHAR2(6) PRIMARY KEY REFERENCES
SALES-ORDER (order-No),
Product-No VARCHAR2(6) PRIMARY KEY REFERENCES
PRODUCT-MASTER (Product-No),"
Qty_Order NUMBER(8),
Qty_Dispatch NUMBER(8)
);

Rahilans
10/6/24

LAB ASSIGNMENT- 7

1. • SELECT SYSDATE FROM dual;
• SELECT ADD_MONTHS(SYSDATE, 3) AS "MONTH"
FROM dual;
• SELECT LAST_DAY(SYSDATE) FROM dual;
• SELECT MONTHS_BETWEEN(SYSDATE, TO_DATE
('2023-01-01', 'YYYY-MM-DD')) AS "MONTHS"
FROM dual;
• SELECT ROUND(SYSDATE, 'MONTH') AS
"ROUNDED Date" FROM dual;
2. CREATE TABLE EMPLOYEE (
EmpId VARCHAR2(3) PRIMARY KEY CHECK
(EmpId LIKE 'E%'),
Ename VARCHAR2(25) NOT NULL,
DOB DATE NOT NULL,
DOJ DATE NOT NULL CHECK (DOJ > DOB),
Salary NUMBER(7) NOT NULL CHECK
(Salary > 0),
Adhar NUMBER(12) UNIQUE CHECK LENGTH
(Adhar) = 12);
2.1) SELECT * FROM EMPLOYEE WHERE DOJ =
TO_DATE('2015-03-15', 'YYYY-MM-DD');
2.2) SELECT Ename, TO_CHAR(DOJ, 'Month DD, YYYY')
AS Hiredate FROM EMPLOYEE;
2.3) SELECT * FROM EMPLOYEE WHERE DOJ <
TO_DATE('2018-01-01', 'YYYY-MM-DD');

- 2.4) SELECT * FROM EMPLOYEE WHERE TO-CHAR(DOJ, 'MM') = '01'; ✓
- 2.5) SELECT * FROM EMPLOYEE WHERE TO-CHAR(DOJ, 'YYYY') = '2018'; ✓
- 2.6) SELECT * FROM EMPLOYEE ORDER BY DOJ DESC; ✓
- 2.7) SELECT Ename, DOJ, ROUND(MONTHS_BETWEEN(SYSDATE, DOJ)/12, 1) AS Experience FROM EMPLOYEE; ✓
- 2.8) SELECT * FROM EMPLOYEE WHERE MOD(Salary, 2) = 1; ✓
- 2.9) SELECT * FROM EMPLOYEE WHERE (EmpId = 'E01' OR EmpId = 'E10') AND TO-CHAR(DOJ, 'YYYY') = '1991'; ✓
- 2.10) SELECT * FROM EMPLOYEE WHERE TO-CHAR(DOJ, 'MM') <> '02'; ✓
- 2.11) SELECT 'Happy Birthday' || Ename AS MESSAGE FROM EMPLOYEE WHERE TO-CHAR(DOB, 'MM-DD') = TO-CHAR(SYSDATE, 'MM-DD'); ✓
- 2.12) SELECT EmpId, Ename, TO-CHAR(DOB, 'DDth Month YYYY') AS Date-of-Birth FROM EMPLOYEE; ✓

LAB ASSIGNMENT - 8

1. Select s.name, c.cust-name, s.city FROM
Salesman s JOIN customer c ON s.city =
c.city AND s.salesman-id = c.salesman-id;
2. Select o.ord-no, o.purch-amt, c.cust-name,
c.city FROM order o JOIN customer c
ON o.customer-id = c.customer-id WHERE
o.purch-amt BETWEEN 500 AND 2000;
3. Select s.name AS salesman-name, c.cust-name
FROM Salesman s JOIN customer c ON
s.salesman-id = c.salesman-id;
4. Select c.cust-name FROM customer c
JOIN Salesman s ON c.salesman-id = s.salesman
id WHERE s.comission > 0.12;
5. Select c.cust-name FROM customer c
JOIN Salesman s ON c.salesman-id =
s.salesman-id WHERE s.city <> c.city
AND s.comission > 0.12;
6. Select o.ord-no, o.ord-date, o.purch-amt,
c.cust-name, s.name, s.comission FROM
order o JOIN customer c ON o.customer-id
= c.customer-id JOIN Salesman s ON
o.salesman-id = s.salesman-id;
7. Select c.cust-name FROM customer c
LEFT JOIN order o ON c.customer-id =
o.customer-id ORDER BY c.cust-name ASC;

8. Select c.cust-name FROM customer c
LEFT JOIN orders o ON c.customer-id =
o.customer-id WHERE c.grade < 300
ORDER BY c.cust-name ASC;
9. Select c.cust-name, c.city, o.ord-no,
o.ord-date, o.purch-amt FROM customer
c LEFT JOIN orders o ON c.customer-id
= o.customer-id ORDER BY o.ord-date ASC;
10. select c.cust-name, c.city, o.ord-no,
o.ord-date, o.purch-amt, s.name, s.commission
FROM customer c LEFT JOIN orders o
ON c.customer-id = o.customer-id ORDER
BY o.order-date ASC;
11. Select s.name FROM salesman s
LEFT JOIN customer c ON s.salesman-id
= c.salesmanid ORDER BY s.name ASC;
12. Select s.name, COUNT(c.customer-id),
COUNT(o.ord-no) FROM salesman s
LEFT JOIN customer c ON s.salesman-id =
c.salesmanid LEFT JOIN orders o ON
c.customer-id = o.customer-id GROUP BY
s.name ORDER BY s.name ASC;

13. Select s.name, COUNT(DISTINCT c.customer-id),
COUNT(o.ord-no) FROM salesman s
LEFT JOIN customer c ON s.salesman-id =
c.salesman-id WHERE c.grade IS NOT NULL
OR o.ord-no IS NULL GROUP BY s.name
ORDER BY s.name ASC;
14. Select s.name AS salesman-name, c.cust-
name FROM salesman s CROSS JOIN
customer c;
15. SELECT i.* , c.* FROM item-mast i
JOIN company-mast c ON i.PRO-COM =
c.COM-ID;
16. SELECT i.PRO-NAME, i.PRO-PRICE, c.COM-NAME
FROM item-mast i JOIN company-mast c
ON i.PRO-COM = c.COM-ID;
17. Select c.COM-NAME, AVG(i.PRO-PRICE) AS
avg-price FROM item-mast i JOIN
company-mast c ON i.PRO-COM = c.COM-ID
GROUP BY c.COM-NAME;
18. Select c.COM-NAME FROM item-mast i
JOIN company-mast c ON i.PRO-COM =
c.COM-ID GROUP BY c.COM-NAME
HAVING AVG(i.PRO-PRICE) >= 350;

19. Select c.com-name, i.pro-id, MAX(i.pro-price)
FROM item-mast i JOIN company-mast c
ON i.pro-com = c.com.id GROUP BY
c.com-name, i.pro-id ORDER BY
max-price DESC;
20. Select e.* , d.* FROM emp-details e
JOIN emp-department d ON e.emp-dept =
d.dpt-code;
21. Select e.emp-fname, e.emp-lname,
d.dept-name, d.dpt-allotment FROM
emp-details e JOIN emp-departmen d
ON e.EMP-DEPT = d.DEPT-CODE;
22. Select e.EMPNAME, e.emp-lname FROM
emp-details e JOIN emp-departmen d
ON e.EMP-DEPT = d.DEPT-CODE WHERE
d.DPT_ALLOTMENT > 500.00;
23. Select d.DEPT_NAME FROM emp-details e
JOIN emp-departmen d ON e.EMP-DEPT =
d.DPT-CODE GROUP BY d.DPT_NAME
HAVING COUNT(e.EMP_IDNO) > 2;

LAB ASSIGNMENT - 9

1. Select SUM(purch-amt) FROM orders;
2. Select AVG(purch-amt) FROM orders;
3. Select COUNT(DISTINCT salesman-id) FROM orders;
4. Select MAX(purch-amt) FROM orders;
5. Select MIN(purch-amt) FROM orders;
6. Select customer-id, MAX(purch-amt) FROM orders GROUP BY customer-id;
7. Select customer-id, ord-date, MAX(purch-amt) FROM orders GROUP BY customer-id, ord-date;
8. Select salesman-id, MAX(purch-amt) FROM orders WHERE ord-date = '2012-08-17' GROUP BY salesman-id;
9. Select customer-id, ord-date, MAX(purch-amt) FROM orders GROUP BY customer-id, ord-date HAVING MAX(purch-amt) > 2000;
10. Select customer-id, ord-date, MAX(purch-amt) FROM orders WHERE purch-amt BETWEEN 2000 and 6000 GROUP BY customer-id, ord-date;
11. Select customer-id, MAX(purch-amt) FROM orders WHERE customer-id BETWEEN 3002 and 3007 GROUP BY customer-id;

2. select COUNT(*) FROM Orders WHERE
ord-date = '2012-08-17';
3. select COUNT(*) FROM customer;
4. select COUNT(*) FROM customer
WHERE grade IS NOT NULL;
5. select city, MAX(grade) FROM customer
GROUP BY city;
6. select city, COUNT(*) FROM Salesman
GROUP BY city;
7. Select Count(*) FROM item-mast
WHERE PRO PRICE >= 350;
8. Select MAX(DOJ) FROM Employee;
9. Select MIN(DOJ) FROM Employee;
10. select COUNT(DISTINCT DEPT) FROM
employee;
11. select COUNT(*) FROM Employee WHERE
Dept = 'MCA';
12. select Dept, MAX(AVG(Salary)) FROM
Employee GROUP BY Dept;
13. Select COUNT(*) FROM employee;
14. select Dept, COUNT(*) FROM employee
GROUP BY Dept;

LAB ASSIGNMENT - 10

```
CREATE TABLE Employee (
    EmpId NUMBER PRIMARY KEY,
    Name VARCHAR(50),
    DOJ DATE,
    JobID VARCHAR(20),
    Salary NUMBER
);
```

- 1.1)

```
SELECT Name FROM Employee
WHERE Salary > (SELECT Salary FROM
Employee WHERE EmpId = 104);
```
- 1.2)

```
SELECT Name, Salary, JobID FROM
Employee WHERE JobID = (SELECT
JobID FROM Employee WHERE EmpId =
103);
```
- 1.3)

```
SELECT Name, Salary, JobID
FROM Employee
WHERE Salary = (SELECT MIN(Salary)
FROM Employee);
```
- 1.4)

```
SELECT * FROM Employee
WHERE DOJ = (SELECT DOJ FROM
Employee WHERE EmpId = 106);
```

1.5) Select * FROM Employee
WHERE DOJ > (SELECT DOJ FROM
Employee WHERE EmpId = 102);

Rahilans
10/6/2024