# DBMS LAB 3 SOLUTION

CREATE TABLE DEPOSIT (

ACTNO INT,

CNAME VARCHAR(50), BNAME VARCHAR(50), AMOUNT DECIMAL(8,2), ADATE DATETIME

);

CREATE TABLE BRANCH (

BNAME VARCHAR(50), CITY VARCHAR(50)

);

CREATE TABLE CUSTOMERS (

CNAME VARCHAR(50), CITY VARCHAR(50)

);

CREATE TABLE BORROW (

LOANNO INT,

CNAME VARCHAR(50), BNAME VARCHAR(50), AMOUNT DECIMAL(8,2)

);

INSERT RECORDS INTO THE TABLE

INSERT INTO DEPOSIT VALUES(101,'ANIL','VRCE',1000.00,'1-MAR-95'); INSERT INTO DEPOSIT VALUES(102,'SUNIL','AJNI',5000.00,'4-JAN-96');

INSERT INTO DEPOSIT VALUES(103,'MEHUL','KAROLBAGH',3500.00,'17-NOV-95'); INSERT INTO DEPOSIT VALUES(104,'MADHURI','CHANDI',1200.00,'17-DEC-95');

INSERT INTO DEPOSIT VALUES(105,'PRMOD','M.G.ROAD',3000.00,'27-MAR-96'); INSERT INTO DEPOSIT VALUES(106,'SANDIP','ANDHERI',2000.00,'31-MAR-96'); INSERT INTO DEPOSIT VALUES(107,'SHIVANI','VIRAR',1000.00,'5-SEP-95'); INSERT INTO DEPOSIT VALUES(108,'KRANTI','NEHRU PLACE',5000.00,'2-JUL-95'); INSERT INTO DEPOSIT VALUES(109,'MINU','POWAI',7000.00,'10-AUG-95');

SELECT \* FROM DEPOSIT;

INSERT INTO BRANCH VALUES('VRCE','NAGPUR'); INSERT INTO BRANCH VALUES('AJNI','NAGPUR'); INSERT INTO BRANCH VALUES('KAROLBAGH','DELHI'); INSERT INTO BRANCH VALUES('CHANDI','DELHI');

INSERT INTO BRANCH VALUES('DHARAMPETH','NAGPUR'); INSERT INTO BRANCH VALUES('M.G.ROAD','BANGLORE'); INSERT INTO BRANCH VALUES('ANDHERI','BOMBAY'); INSERT INTO BRANCH VALUES('VIRAR','BOMBAY');

INSERT INTO BRANCH VALUES('NEHRU PLACE','DELHI'); INSERT INTO BRANCH VALUES('POWAI','BOMBAY');

SELECT \* FROM BRANCH;

INSERT INTO CUSTOMERS VALUES('ANIL','CALCUTTA'); INSERT INTO CUSTOMERS VALUES('SUNIL','DELHI'); INSERT INTO CUSTOMERS VALUES('MEHIL','BARODA'); INSERT INTO CUSTOMERS VALUES('MANDAR','PATNA'); INSERT INTO CUSTOMERS VALUES('MADHURI','NAGPUR'); INSERT INTO CUSTOMERS VALUES('PRAMOD','NAGPUR'); INSERT INTO CUSTOMERS VALUES('SANDIP','SURAT'); INSERT INTO CUSTOMERS VALUES('SHIVANI','BOMBAY'); INSERT INTO CUSTOMERS VALUES('KRANTI','BOMBAY'); INSERT INTO CUSTOMERS VALUES('NAREN','BOMBAY');

SELECT \* FROM CUSTOMERS;

INSERT INTO BORROW VALUES(201,'ANIL','VRCE',1000.00); INSERT INTO BORROW VALUES(206,'MEHUL','AJNI',5000.00);

INSERT INTO BORROW VALUES(311,'SUNIL','DHARAMPETH',3000.00);

INSERT INTO BORROW VALUES(321,'MADHURI','ANDHERI',2000.00); INSERT INTO BORROW VALUES(375,'PRAMOD','VIRAR',8000.00); INSERT INTO BORROW VALUES(481,'KRANTI','NEHRU PLACE',3000.00);

SELECT \* FROM BORROW;

# PART A

1. Retrieve all data from table DEPOSIT. SELECT \* FROM DEPOSIT;
2. Retrieve all data from table BORROW. SELECT \* FROM BORROW;
3. Retrieve all data from table CUSTOMERS. SELECT \* FROM CUSTOMERS;
4. Display Account No, Customer Name & Amount from DEPOSIT. SELECT ACTNO, CNAME, AMOUNT FROM DEPOSIT;
5. Display Loan No, Amount from BORROW SELECT LOANNO ,AMOUNT FROM BORROW;
6. Display loan details of all customers who belongs to ‘ANDHERI’ branch SELECT \* FROM BORROW

WHERE BNAME='ANDHERI';

1. Give account no and amount of depositor, whose account no is equals to 106. SELECT ACTNO,AMOUNT FROM DEPOSIT

WHERE ACTNO=106;

1. Give name of borrowers having amount greater than 5000. SELECT CNAME FROM BORROW

WHERE AMOUNT >5000.00;

1. Give name of customers who opened account after date '1-12-96' SELECT CNAME FROM DEPOSIT

WHERE ADATE > '1-DEC-96';

1. Display name of customers whose account no is less than 105. SELECT CNAME FROM DEPOSIT

WHERE ACTNO < 105;

# PART B

1. Display name of customer who belongs to either ‘NAGPUR’ Or ‘DELHI’. (OR & IN) SELECT CNAME FROM CUSTOMERS

WHERE CITY IN('NAGPUR','DELHI');

SELECT CNAME FROM CUSTOMERS WHERE CITY='NAGPUR'OR CITY='DELHI'

1. Display name of customers with branch whose amount is greater than 4000 and account no is less than 105.

SELECT CNAME,BNAME FROM DEPOSIT

WHERE AMOUNT > 4000.00 AND ACTNO < 105;

1. Find all borrowers whose amount is greater than equals to 3000 & less than equals to 8000.(AND & BETWEEN)

SELECT CNAME FROM BORROW

WHERE AMOUNT >= 3000.00 AND AMOUNT <= 8000.00;

SELECT CNAME FROM BORROW

WHERE AMOUNT BETWEEN 3000.00 AND 8000.00;

1. Find all depositors who do not belongs to ‘ANDHERI’ branch. SELECT CNAME FROM BORROW

WHERE BNAME='ANDHERI';

1. Display Account No, Customer Name & Amount of such customers who belongs to ‘AJNI’, ‘KAROLBAGH’ Or ‘M.G.ROAD’ and Account No is less than 104.

SELECT ACTNO,CNAME,AMOUNT FROM DEPOSIT

WHERE BNAME IN('AJNI','KAROLBAGH','M.G.ROAD') AND ACTNO < 104;

1. Display all the details of ﬁrst ﬁve customers. SELECT TOP 5 \* FROM DEPOSIT

WHERE ACTNO <= 105;

1. Display all the details of ﬁrst three depositors whose amount is greater than 1000.

SELECT TOP 3 \* FROM DEPOSIT WHERE AMOUNT > 1000;

1. Display Loan No, Customer Name of ﬁrst ﬁve borrowers whose branch name does not belongs to ‘ANDHERI’

SELECT TOP 5 LOANNO,CNAME FROM BORROW WHERE BNAME != 'ANDHERI';

1. Retrieve all unique cities using DISTINCT. (Use Customers Table) SELECT DISTINCT CITY FROM CUSTOMERS;
2. Retrieve all unique branches using DISTINCT. (Use Branch Table) SELECT DISTINCT BNAME FROM BRANCH;
3. Retrieve all the records of customer table as per their city name in ascending order.

SELECT \* FROM CUSTOMERS ORDER BY CITY;

1. Retrieve all the records of deposit table as per their amount column in descending order.

SELECT \* FROM DEPOSIT ORDER BY AMOUNT DESC;

PART C

1. Update deposit amount of all customers from 3000 to 5000. UPDATE DEPOSIT

SET AMOUNT = 5000.00

WHERE AMOUNT = 3000.00;

1. Change branch name of ANIL from VRCE to C.G. ROAD. (Use Borrow Table) UPDATE DEPOSIT

SET BNAME = 'C.G.ROAD' WHERE CNAME = 'ANIL';

1. Update Account No of SANDIP to 111 & Amount to 5000. UPDATE DEPOSIT

SET ACTNO = 111, AMOUNT = 5000.00 WHERE CNAME = 'ANIL';

1. Give 10% Increment in Loan Amount. UPDATE BORROW

SET AMOUNT = AMOUNT + (AMOUNT\*10/100);

1. Update deposit amount of all depositors to 5000 whose account no between 103 & 107.

UPDATE DEPOSIT

SET AMOUNT = 5000.00

WHERE ACTNO BETWEEN 103 AND 107;

1. Update amount of loan no 321 to NULL. UPDATE BORROW

SET AMOUNT = NULL WHERE LOANNO = 321;

1. Display the name of borrowers whose amount is NULL SELECT CNAME FROM BORROW

WHERE AMOUNT IS NULL;

# DBMS LAB 4 SOLUTION

CREATE TABLE EMPLOYEE (

EMPNO INT,

EMPNAME VARCHAR(25), JOININGDATE DATETIME, SALARY DECIMAL(8,2), CITY VARCHAR(20)

);

INSERT INTO EMPLOYEE VALUES(101,'KEYUR','5-JAN-2002',12000.00,'RAJKOT');

INSERT INTO EMPLOYEE VALUES(102,'HARDIK','15-FEB-2004',14000.00,'AHMEDABAD'); INSERT INTO EMPLOYEE VALUES(103,'KAJAL','14-MAR-2006',15000.00,'BARODA');

INSERT INTO EMPLOYEE VALUES(104,'BHOOMI','23-JUNE-2005',12500.00,'AHMEDABAD'); INSERT INTO EMPLOYEE VALUES(102,'HARMIT','15-FEB-2004',14000.00,'RAJKOT');

SELECT \* FROM EMPLOYEE;

# PART A

1. Delete all the records of Employee table having salary greater than and equals to 14000.

DELETE FROM EMPLOYEE WHERE SALARY >= 14000.00;

1. Delete all the Employees who belongs to ‘RAJKOT’ city DELETE FROM EMPLOYEE

WHERE CITY = 'RAJKOT';

1. Delete all the Employees who joined after 1-1-2007 DELETE FROM EMPLOYEE

WHERE JOININGDATE > '1-JAN-2007';

1. Delete all the records of Employee table. (Use Truncate) TRUNCATE TABLE EMPLOYEE;
2. Remove Employee table. (Use Drop) DROP TABLE EMPLOYEE;

# PART B

CREATE TABLE ACCOUNT (

ACTNO INT,

CNAME VARCHAR(50), BNAME VARCHAR(50), AMOUNT DECIMAL(8,2), ADATE DATETIME

);

INSERT INTO ACCOUNT VALUES(101,'ANIL','VRCE',1000.00,'1-MAR-1995'); INSERT INTO ACCOUNT VALUES(102,'SUNIL','AJNI',5000.00,'4-JAN-1996');

INSERT INTO ACCOUNT VALUES(103,'MEHUL','KAROLBAGH',3500.00,'17-NOV-1995'); INSERT INTO ACCOUNT VALUES(104,'MADHURI','CHANDI',1200.00,'17-DEC-1995'); INSERT INTO ACCOUNT VALUES(105,'PRAMOD','M.G.ROAD',3000.00,'27-MAR-1996'); INSERT INTO ACCOUNT VALUES(106,'SANDIP','ANDHERI',2000.00,'31-MAR-1996'); INSERT INTO ACCOUNT VALUES(107,'SHIVANI','VIRAR',1000.00,'5-SEP-1995');

INSERT INTO ACCOUNT VALUES(108,'KRANTI','NEHRU PLACE',5000.00,'2-JULY-1995'); INSERT INTO ACCOUNT VALUES(109,'MINU','POWAI',7000.00,'10-AUG-1995');

SELECT \* FROM ACCOUNT;

1. Delete all the records of Account table having amount greater than and equals to 4000.

DELETE FROM ACCOUNT WHERE AMOUNT >= 4000.00;

1. Delete all the accounts Bname is CHANDI. DELETE FROM ACCOUNT

WHERE BNAME = 'CHANDI';

1. Delete all the accounts having adate after 1-10-1995. DELETE FROM ACCOUNT

WHERE ADATE > '1-OCT-1995';

1. Delete all the records of Account table. (Use Truncate) TRUNCATE TABLE ACCOUNT;
2. Remove Account table. (Use Drop) DROP TABLE ACCOUNT;

# PART C

1. Update the amount of Anil to 5000. UPDATE ACCOUNT

SET AMOUNT = 5000.00 WHERE CNAME ='ANIL';

1. Update amount of actno 109 to NULL. UPDATE ACCOUNT

SET AMOUNT = NULL WHERE ACTNO = 109;

1. Retrieve all the records of account table as per their bname in descending order SELECT \* FROM ACCOUNT

ORDER BY BNAME DESC;

1. Retrieve all the records of account table whose amount is between 3000 and 5000.

SELECT \* FROM ACCOUNT

WHERE AMOUNT BETWEEN 3000.00 AND 5000.00;

1. Remove Account table. (Use Drop) DROP TABLE ACCOUNT;

# DBMS LAB 5 SOLUTION

CREATE TABLE STUDENT (

ENROLLMENT\_NO VARCHAR(20), NAME VARCHAR(25),

CPI DECIMAL(5,2), BIRTHDATE DATETIME

);

# PART A

1. Add two more columns City VARCHAR (20) and Backlog INT. ALTER TABLE STUDENT

ADD CITY VARCHAR(20), BACKLOG INT;

1. Change the size of NAME column of student from VARCHAR (25) to VARCHAR (35).

ALTER TABLE STUDENT

ALTER COLUMN NAME VARCHAR(35);

1. Change the data type DECIMAL to INT in CPI Column. ALTER TABLE STUDENT

ALTER COLUMN CPI INT;

1. Rename Column Enrollment No to ENO. SP\_RENAME 'STUDENT.ENROLLMENT\_NO','ENO';
2. Delete Column City from the STUDENT table. ALTER TABLE STUDENT

DROP COLUMN CITY;

1. Change name of table STUDENT to STUDENT\_MASTER. SP\_RENAME 'STUDENT', 'STUDENT\_MASTER';
2. Remove the table STUDENT\_MASTER DROP TABLE STUDENT\_MASTER;

# PART B

CREATE TABLE DEPOSIT (

ACTNO INT,

CNAME VARCHAR(50), BNAME VARCHAR(50), AMOUNT DECIMAL(8,2), ADATE DATETIME

);

1. Add two more columns City VARCHAR (20) and Pincode INT. ALTER TABLE DEPOSITE

ADD CITY VARCHAR(20), PINCODE INT;

1. Change the size of CNAME column from VARCHAR (50) to VARCHAR (35). ALTER TABLE DEPOSITE

ALTER COLUMN CNAME VARCHAR(35);

1. Change the data type DECIMAL to INT in amount Column. ALTER TABLE DEPOSITE

ALTER COLUMN AMOUNT INT;

1. Rename Column ActNo to ANO. SP\_RENAME 'DEPOSITE.ACTNO', 'ANO';
2. Delete Column City from the DEPOSIT table. ALTER TABLE DEPOSITE

DROP COLUMN CITY;

# PART C

1. Delete Column ADATE from the DEPOSIT table. ALTER TABLE DEPOSITE

DROP COLUMN ADATE;

1. Rename Column CNAME to CustomerName. SP\_RENAME 'DEPOSITE.CNAME', 'CUSTOMERNAME';
2. Change name of table DEPOSIT to DEPOSIT\_DETAIL. SP\_RENAME 'DEPOSITE', 'DEPOSITE\_DETAIL';
3. Remove the table DEPOSIT\_DETAIL. ALTER TABLE DEPOSITE\_DETAIL DROP COLUMN CITY;

# DBMS LAB 6 SOLUTION

CREATE TABLE STUDENT (

STUID INT,

FIRSTNAME VARCHAR(25), LASTNAME VARCHAR(25), WEBSITE VARCHAR(50), CITY VARCHAR(25)

);

INSERT INTO STUDENT VALUES(1011,'KEYUR','PATEL','ECHONTHENET.COM','RAJKOT'); INSERT INTO STUDENT VALUES(1022,'HARDIK','SHAH','DIGMINECRAFT.COM','AHMEDABAD'); INSERT INTO STUDENT VALUES(1033,'KAJAL','TRIVEDI','BIGACTIVITIES.COM','BARODA'); INSERT INTO STUDENT

VALUES(1044,'BHOOMI','GAJERA','CHECKYOURMATH.COM','AHMEDABAD'); INSERT INTO STUDENT VALUES(1055,'HARMIT','MITEL',NULL,'RAJKOT'); INSERT INTO STUDENT VALUES(1066,'ASHOK','JANI',NULL,'BARODA');

SELECT \* FROM STUDENT;

# PART A

1. Display the name of students whose name starts with ‘k’. SELECT FIRSTNAME FROM STUDENT

WHERE FIRSTNAME LIKE 'K%';

1. Display the name of students whose name consists of ﬁve characters. SELECT FIRSTNAME FROM STUDENT

WHERE FIRSTNAME LIKE ' ';

1. Retrieve the ﬁrst name & last name of students whose city name ends with a & contains six characters.

SELECT FIRSTNAME, LASTNAME FROM STUDENT WHERE CITY LIKE ' \_A';

1. Display all the students whose last name ends with ‘tel’. SELECT \* FROM STUDENT

WHERE LASTNAME LIKE '%TEL';

1. Display all the students whose ﬁrst name starts with ‘ha’ & ends with‘t’. SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE 'HA%T';

1. Display all the students whose ﬁrst name starts with ‘k’ and third character is ‘y SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE 'K\_Y%';

# PART B

1. Display the name of students having no website and name consists of ﬁve characters

SELECT FIRSTNAME, LASTNAME FROM STUDENT WHERE WEBSITE IS NULL AND FIRSTNAME LIKE ' ';

1. Display all the students whose last name consist of ‘jer’. SELECT \* FROM STUDENT

WHERE LASTNAME LIKE '%JER%';

1. Display all the students whose city name starts with either ‘r’ or ‘b’. SELECT \* FROM STUDENT

WHERE CITY LIKE 'R%' OR CITY LIKE 'B%';

SELECT \* FROM STUDENT WHERE CITY LIKE '[R,B]%';

1. Display all the name students having websites SELECT \* FROM STUDENT

WHERE WEBSITE IS NOT NULL;

1. Display all the students whose name starts from alphabet A to H. SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE '[A-H]%';

1. Display all the students whose name’s second character is vowel. SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE '\_[A,E,I,O,U]%';

# PART C

1. Display the name of students having no website and name consists of minimum ﬁve characters.

SELECT FIRSTNAME, LASTNAME FROM STUDENT

WHERE WEBSITE IS NOT NULL AND FIRSTNAME LIKE ' \_%';

1. Display all the students whose last name starts with ‘Pat’. SELECT \* FROM STUDENT

WHERE LASTNAME LIKE 'PAT%';

1. Display all the students whose city name does not start starts with ‘b’. SELECT \* FROM STUDENT

WHERE CITY NOT LIKE 'B%';

1. Display all the students whose name starts from alphabet A or H. SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE 'A%' OR FIRSTNAME LIKE 'H%';

1. Display all the students whose name’s second character is vowel and of and start with H.

SELECT \* FROM STUDENT

WHERE FIRSTNAME LIKE 'H[A,E,I,O,U]%';

# DBMS LAB 7 SOLUTION

PART A

1. Display the result of 5 multiply by 30. SELECT 5\*30
2. Find out the absolute value of -25, 25, -50 and 50. SELECT ABS(-25),ABS(25),ABS(-50),ABS(50)
3. Find smallest integer value that is greater than or equal to 25.2, 25.7 and -25.2. SELECT CEILING(25.2),CEILING(-25.7),CEILING(-25.2)
4. Find largest integer value that is smaller than or equal to 25.2, 25.7 and -25.2. SELECT FLOOR(25.2),FLOOR(25.7),FLOOR(-25.2)
5. Find out remainder of 5 divided 2 and 5 divided by 3. SELECT 5%2 ,5%3
6. Find out value of 3 raised to 2nd power and 4 raised 3rd power. SELECT POWER(3,2),POWER(4,3)
7. Find out the square root of 25, 30 and 50. SELECT SQRT(25),SQRT(30),SQRT(50)
8. Find out the square of 5, 15, and 25. SELECT SQUARE(5),SQUARE(15),SQUARE(25)
9. Find out the value of PI. SELECT PI()
10. Find out round value of 157.732 for 2, 0 and -2 decimal points. SELECT ROUND(157.732,2),ROUND(157.732,0),ROUND(1572.732,-2)
11. Find out exponential value of 2 and 3. SELECT EXP(2),EXP(3)
12. Find out logarithm having base e of 10 and 2. SELECT LOG(10),LOG(2)
13. Find out logarithm having base b having value 10 of 5 and 100. SELECT LOG10(5),LOG10(100)
14. Find sine, cosine and tangent of 3.1415. SELECT SIN(3.1415),COS(3.1415),TAN(3.1415)
15. Find sign of -25, 0 and 25 SELECT SIGN(-25),SIGN(0),SIGN(25)
16. Generate random number using function. SELECT RAND()

# PART B

1. Display the result of Salary plus Commission.

SELECT Salary,Commission,(Salary+Commission) AS Result From Employee

1. Find smallest integer value that is greater than or equal to 55.2, 35.7 and -55.2. SELECT CEILING(55.2),CEILING(35.7),CEILING(-55.2)
2. Find largest integer value that is smaller than or equal to 55.2, 35.7 and -55.2. SELECT FLOOR(55.2),FLOOR(35.7),FLOOR(-55.2)
3. Find out remainder of 55 divided 2 and 55 divided by 3. SELECT 55%2,55%3
4. Find out value of 23 raised to 2nd power and 14 raised 3rd power. SELECT POWER(23,2) AS '23^2',POWER(14,3) AS '14^3'

# PART C

1. Find out the square root of 36, 49 and 81.

SELECT SQRT(36) AS 'Square root of 36',SQRT(49) AS 'Square root of 49',SQRT(81) AS 'Square root of 81'

1. Find out the square of 3, 9, and 12.

SELECT SQUARE(3) AS 'Square of 3',SQUARE(9) AS 'Square of 9',SQUARE(12) AS

'Square of 12'

1. Find out round value of 280.8952 for 2, 0 and -2 decimal points. SELECT ROUND(280.8952,2),ROUND(280.8952,0),ROUND(280.8952,-2)
2. Find sine, cosine and tangent of 4.2014.

SELECT SIN(4.2014) AS 'SINE',COS(4.2014) AS 'COS',TAN(4.2014) AS 'TANGENT'

1. Find sign of -55, 0 and 95.

SELECT SIGN(-55) AS 'Negative',SIGN(0) AS 'Zero',SIGN(95) AS 'Positive'

# DBMS LAB 8 SOLUTION

PART A

1. Find the length of following. (I) NULL (II) ‘ hello ’ (III) Blank SELECT LEN(NULL),LEN('hello'),LEN(' ')
2. Display your name in lower & upper case.

SELECT LOWER('DEEP') as LowerCase,UPPER('deep') as UpperCase

1. Display ﬁrst three characters of your name. SELECT SUBSTRING('DEEP',1,3)
2. Display 3rd to 10th character of your name. SELECT SUBSTRING('DEEP PADALIA',3,8)
3. Write a query to convert ‘abc123efg’ to ‘abcXYZefg’ & ‘abcabcabc’ to ‘ab5ab5ab5’ using REPLACE.

SELECT REPLACE('abc123efg','123','XYZ'),REPLACE('abcabcabc','C','5')

1. Write a query to display ASCII code for ‘a’,’A’,’z’,’Z’, 0, 9. SELECT

ASCII('a') as 'ASCII of a',

ASCII('A') as 'ASCII of A',

ASCII('z') as 'ASCII of z',

ASCII('Z') as 'ASCII of Z', ASCII(0) as 'ASCII of 0', ASCII(9) as 'ASCII of 9'

1. Write a query to display character based on number 97, 65,122,90,48,57. SELECT CHAR(97),CHAR(65),CHAR(122),CHAR(90),CHAR(48),CHAR(57)
2. Write a query to remove spaces from left of a given string ‘ hello world ‘. SELECT LTRIM(' hello world')
3. Write a query to remove spaces from right of a given string ‘ hello world ‘. SELECT RTRIM(' hello world ')
4. Write a query to display ﬁrst 4 & Last 5 characters of ‘SQL Server’. SELECT LEFT('SQL SERVER',4),RIGHT('SQL SERVER',5)
5. Write a query to convert a string ‘1234.56’ to number (Use cast and convert function).

SELECT CAST('123.56' AS FLOAT),CONVERT(FLOAT,'123.56')

1. Write a query to convert a ﬂoat 10.58 to integer (Use cast and convert function).

SELECT CAST(10.58 AS INT),CONVERT(INT,10.58)

1. Put 10 space before your name using function. SELECT SPACE(10) + 'DEEP'
2. Combine two strings using + sign as well as CONCAT (). SELECT CONCAT('DEEP',' PADALIA')

SELECT 'DEEP' + ' PADALIA'

1. Find reverse of “Darshan”. SELECT REVERSE('Darshan')
2. Repeat your name 3 times. SELECT REPLICATE('DEEP ',3)

# PART B

1. Find the length of FirstName and LastName columns. SELECT LEN(FirstName),LEN(LastName) from Student
2. Display FirstName and LastName columns in lower & upper case. SELECT LOWER(FirstName),UPPER(FirstName),LOWER(LastName), UPPER(LastName) from Student
3. Display ﬁrst three characters of FirstName column. SELECT LEFT(FirstName,3)from Student
4. Display 3rd to 10th character of Website column. SELECT SUBSTRING(Website,3,10) from Student
5. Write a query to display ﬁrst 4 & Last 5 characters of Website column. SELECT LEFT(Website,4),RIGHT(Website,5)from Student

# PART C

1. Put 10 space before FirstName using function. SELECT SPACE(10) + FirstName from Student
2. Combine FirstName and LastName columns using + sign as well as CONCAT (). SELECT (FirstName + ' ' + LastName) as 'using +' from Student

SELECT CONCAT(FirstName,' ',LastName) as 'using CONCAT()' from Student

1. Combine all columns using + sign as well as CONCAT ().

SELECT (CONVERT(VARCHAR,StudID) + ' ' + FirstName + ' ' + LastName + ' ' +

Website + ' ' + City) as 'using +' from Student

SELECT CONCAT(StudID, ' ',FirstName,' ',LastName,' ',Website,' ',City) as 'using CONCAT()' from Student

1. Find reverse of FirstName column. SELECT REVERSE(FirstName) from Student
2. Repeat FirstName column 3 times.

SELECT REPLICATE(FirstName,3)from Student

# DBMS LAB 9 SOLUTION

PART A

1. Write a query to display the current date & time. Label the column Today\_Date. SELECT GETDATE() AS TODAY\_Date
2. Write a query to ﬁnd new date after 365 day with reference to today. SELECT GETDATE()+365
3. Display the current date in a format that appears as may 5 1994 12:00AM. SELECT CONVERT(VARCHAR, GETDATE())
4. Display the current date in a format that appears as 03 Jan 1995. SELECT CONVERT(VARCHAR, GETDATE(),6)
5. Display the current date in a format that appears as Jan 04, 96. SELECT CONVERT(VARCHAR, GETDATE(),7)
6. Write a query to ﬁnd out total number of months between 31-Dec-08 and 31- Mar-09.

SELECT DATEDIFF(MONTH,'31/DEC/08','31/MAR/09')

1. Write a query to ﬁnd out total number of years between 25-Jan-12 and 14-Sep- 10.

SELECT DATEDIFF(YEAR,'25/JAN/12','14/SEP/10')

1. Write a query to ﬁnd out total number of hours between 25-Jan-12 7:00 and 26-Jan-12 10:30.

SELECT DATEDIFF(HOUR,'25/JAN/12 7:00','26/JAN/12 10:30')

1. Write a query to extract Day, Month, Year from given date 12-May-16. SELECT DAY('12/May/16') as Cday,MONTH('12/May/16') as Cmonth, YEAR('12/May/16') AS CYEAR
2. Write a query that adds 5 years to current date. SELECT DATEADD(YEAR,5,GETDATE())
3. Write a query to subtract 2 months from current date. SELECT DATEADD(MONTH,-2,GETDATE())
4. Extract month from current date using datename () and datepart () function. SELECT DATENAME(MONTH,GETDATE()),DATEPART(MONTH,GETDATE())
5. Write a query to ﬁnd out last date of current month. SELECT EOMONTH(GETDATE())
6. Calculate your age in years and months.

SELECT DATEDIFF(YEAR,'24/OCT/99',GETDATE()) AS AGE\_IN\_YEAR, DATEDIFF(MONTH,'24/OCT/99',GETDATE())/12 AS AGE\_IN\_MONTH, DATEDIFF(DAY,'24/OCT/99',GETDATE()) AS AGE\_IN\_DAYS

# PART B

SELECT \* FROM EMPLOYEE

1. Write a query to ﬁnd new date after 365 day with reference to JoiningDate. SELECT ([JOINNIGDATE] + 365) FROM EMPLOYEE
2. Display the JoiningDate in a format that appears as may 5 1994 12:00AM. SELECT CONVERT(VARCHAR ,[JOINNIGDATE]) FROM EMPLOYEE
3. Display the JoiningDate in a format that appears as 03 Jan 1995. SELECT CONVERT(VARCHAR ,[JOINNIGDATE],6) FROM EMPLOYEE
4. Display the JoiningDate in a format that appears as Jan 04, 96. SELECT CONVERT(VARCHAR ,[JOINNIGDATE],7) FROM EMPLOYEE
5. Write a query to ﬁnd out total number of months between JoiningDate and 31- Mar-09.

SELECT DATEDIFF(MONTH,[JOINNIGDATE],'31/MAR/09') FROM EMPLOYEE

1. Write a query to ﬁnd out total number of years between JoiningDate and 14- Sep-10.

SELECT DATEDIFF(YEAR,[JOINNIGDATE],'14/SEP/10') FROM EMPLOYEE

# PART C

1. Write a query to extract Day, Month, Year from JoiningDate.

SELECT DAY([JOINNIGDATE]) as Cday,MONTH([JOINNIGDATE]) as Cmonth, YEAR([JOINNIGDATE]) AS CYEAR FROM EMPLOYEE

1. Write a query that adds 5 years to JoiningDate.

SELECT DATEADD(YEAR,5,[JOINNIGDATE]) FROM EMPLOYEE

1. Write a query to subtract 2 months from JoiningDate. SELECT DATEADD(YEAR,5,GETDATE())
2. Extract month from JoiningDate using datename () and datepart () function. SELECT DATENAME(MONTH,[JOINNIGDATE]),DATEPART(MONTH, [JOINNIGDATE]) FROM EMPLOYEE
3. Calculate your age in years and months.

SELECT DATEDIFF(YEAR,'24/OCT/99',GETDATE()) AS AGE\_IN\_YEAR, DATEDIFF(MONTH,'24/OCT/99',GETDATE()) AS AGE\_IN\_MONTH,

DATEDIFF(DAY,'24/OCT/99',GETDATE()) AS AGE\_IN\_DAYS

# DBMS LAB 10 SOLUTION

PART A

CREATE TABLE COMPUTER (

RollNo INT,

Name VARCHAR(100)

);

CREATE TABLE ELECTRICAL (

RollNo INT,

Name VARCHAR(100)

);

INSERT INTO COMPUTER VALUES(101,'Ajay') INSERT INTO COMPUTER VALUES(109,'Haresh') INSERT INTO COMPUTER VALUES(115,'Manish')

INSERT INTO ELECTRICAL VALUES(105,'Ajay') INSERT INTO ELECTRICAL VALUES(107,'Mahesh') INSERT INTO ELECTRICAL VALUES(115,'Manish')

1. Display name of students who is either in COMPUTER or in ELECTRICAL. SELECT NAME FROM COMPUTER

UNION

SELECT NAME FROM ELECTRICAL

1. Display name of students who is either in COMPUTER or in ELECTRICAL including duplicate data.

SELECT NAME FROM COMPUTER UNION ALL

SELECT NAME FROM ELECTRICAL

1. Display name of students who is in both COMPUTER and ELECTRICAL. SELECT NAME FROM COMPUTER

INTERSECT

SELECT NAME FROM ELECTRICAL

1. Display name of students who are in COMPUTER but not in ELECTRICAL. SELECT NAME FROM COMPUTER

EXCEPT

SELECT NAME FROM ELECTRICAL

1. Display name of students who are in ELECTRICAL but not in COMPUTER. SELECT NAME FROM ELECTRICAL

EXCEPT

SELECT NAME FROM COMPUTER

1. Display all the details of students who are either in COMPUTER or in ELECTRICAL.

SELECT \* FROM COMPUTER UNION

SELECT \* FROM ELECTRICAL

1. Display all the details of students who are in both COMPUTER and ELECTRICAL. SELECT \* FROM COMPUTER

INTERSECT

SELECT \* FROM ELECTRICAL

# PART B

CREATE TABLE EMPLOYEE1 (

EID INT,

Name VARCHAR(100)

);

CREATE TABLE CUSTOMER (

CID INT,

Name VARCHAR(100)

);

INSERT INTO EMPLOYEE1 VALUES(1,'Ajay') INSERT INTO EMPLOYEE1 VALUES(9,'Haresh')

INSERT INTO EMPLOYEE1 VALUES(5,'Manish') INSERT INTO CUSTOMER VALUES(5,'Ajay') INSERT INTO CUSTOMER VALUES(7,'Mahesh') INSERT INTO CUSTOMER VALUES(5,'Manish')

1. Display name of persons who is either Employee or CUSTOMER. SELECT NAME from EMPLOYEE1

UNION

SELECT NAME from CUSTOMER

1. Display name of persons who is either Employee or CUSTOMER including duplicate data.

SELECT NAME from EMPLOYEE1 UNION ALL

SELECT NAME from CUSTOMER

1. Display name of persons who is both Employee as well as CUSTOMER. SELECT NAME from EMPLOYEE1

INTERSECT

SELECT NAME from CUSTOMER

1. Display name of persons who are Employee but not CUSTOMER. SELECT NAME from EMPLOYEE1

EXCEPT

SELECT NAME from CUSTOMER

1. Display name of persons who are CUSTOMER but not Employee. SELECT NAME from CUSTOMER

EXCEPT

SELECT NAME from EMPLOYEE1

# DBMS LAB 11 SOLUTION

CREATE TABLE CRICKET (

Name VARCHAR(50), City VARCHAR(50),

Age INT

);

INSERT INTO CRICKET VALUES

('Sachin Tendulkar','Mumbai',30), ('Rahul Dravid','Bombay',35),

('M. S. Dhoni','Jharkhand',31),

('Suresh Raina','Gujarat',30);

# PART- A

1. Create table Worldcup from cricket with all the columns. SELECT \* INTO Worldcup FROM CRICKET;
2. Create table T20 from cricket with ﬁrst two columns with no data. SELECT Name,City INTO T20 FROM CRICKET

WHERE 1=2;

1. Create table IPL From Cricket with No Data SELECT \* INTO IPL FROM CRICKET

WHERE 1=2;

1. Insert the Data into IPL from Cricket Whose Second Character Should Be ‘A’ And String Should Have At least 7 Characters in Cricket Name Field.

INSERT INTO IPL2018 SELECT \* FROM CRICKET WHERE Name LIKE '\_A %';

1. Delete All the Rows From IPL. DELETE FROM IPL;
2. Delete the Detail of Cricketer Whose City is Jharkhand. DELETE FROM CRICKET

WHERE City='Jharkhand';

1. Rename the Table IPL to IPL2018. SP\_RENAME 'IPL','IPL2018';
2. Destroy table T20 with All the Data. DROP TABLE T20;

# PART- B

CREATE TABLE EMPLOYEE (

Name VARCHAR(50), City VARCHAR(50),

Age INT

);

INSERT INTO EMPLOYEE VALUES

('Jay Patel','Rajkot',30),

('Rahul Dave','Baroda',35),

('Jeet Patel','Surat',31),

('Vijay Raval','Rajkot',30);

1. Create table Employee\_detail from Employee with all the columns and data. SELECT \* INTO Employee\_detail FROM EMPLOYEE;
2. Create table Employee\_data from Employee with ﬁrst two columns with no data.

SELECT Name,City INTO Employee\_data FROM EMPLOYEE WHERE 1=2;

1. Create table Employee\_info from Employee with no Data SELECT \* INTO Employee\_info FROM EMPLOYEE

WHERE 1=2;

1. Insert the Data into Employee\_info from Employee Whose Second Character Should Be ‘A’ And String Should Have At least 7 Characters in Name Field. INSERT INTO Employee\_info SELECT \* FROM EMPLOYEE

WHERE Name LIKE '\_A %';

1. Insert the Data into Employee\_info from Employee Whose age is more than 32. INSERT INTO Employee\_info SELECT \* FROM EMPLOYEE

WHERE Age>32;

# PART- C

1. Delete all the Rows from Employee\_info. DELETE FROM Employee\_info;
2. Delete the Detail of employee from Employee Whose City is Rajkot. DELETE FROM EMPLOYEE

WHERE City='Rajkot';

1. Rename the Table Employee to New\_Employee. SP\_RENAME 'EMPLOYEE','NEW\_EMPLOYEE';
2. Delete all the records from New\_Employee table. DELETE FROM NEW\_EMPLOYEE;
3. Destroy table New\_Employee with all the Data. DROP TABLE NEW\_EMPLOYEE;

# DBMS LAB 12 SOLUTION

CREATE TABLE EMPLOYEE (

EID INT,

ENAME VARCHAR(25), DEPT VARCHAR(30), SALARY INT,

JDATE DATETIME, CITY VARCHAR(50)

);

INSERT INTO EMPLOYEE VALUES(101,'RAHUL','ADMIN',56000,'1-JAN-1990','RAKJOT'); INSERT INTO EMPLOYEE VALUES(102,'HARDIK','IT',18000,'25-SEP-1990','AHMEDABAD'); INSERT INTO EMPLOYEE VALUES(103,'BHAVIN','HR',25000,'14-MAY-1991','BARODA'); INSERT INTO EMPLOYEE VALUES(104,'BHOOMI','ADMIN',39000,'8-FEB-1991','RAKJOT'); INSERT INTO EMPLOYEE VALUES(105,'ROHIT','IT',17000,'23-JUL-1990','JAMNAGAR'); INSERT INTO EMPLOYEE VALUES(106,'PRIYA','IT',9000,'18-OCT-1990','AHMEDABAD'); INSERT INTO EMPLOYEE VALUES(107,'NEHA','HR',34000,'25-DEC-1991','RAKJOT');

# PART A

1. Display the Highest, Lowest, Total, and Average salary of all employees. Label the columns Maximum, Minimum, Total\_Sal and Average\_Sal, respectively.

SELECT MAX(SALARY) AS MAXIMUM, MIN(SALARY) AS MINIMUM, SUM(SALARY) AS TOTAL\_SAL, AVG(SALARY) AS AVERAGE\_SAL FROM EMPLOYEE;

1. Find total number of employees of EMPLOYEE table. SELECT COUNT(EID) AS TOTAL\_EMP FROM EMPLOYEE;
2. Give maximum salary from IT department. SELECT MAX(SALARY) FROM EMPLOYEE WHERE DEPT = 'IT';
3. Count total number of cities of employee without duplication. SELECT CITY FROM EMPLOYEE GROUP BY CITY;

SELECT DISTINCT CITY FROM EMPLOYEE;

1. Display city with the total number of employees belonging to each city. SELECT CITY, COUNT(EID) FROM EMPLOYEE

GROUP BY CITY;

1. Display city having more than one employee. SELECT CITY, COUNT(EID) FROM EMPLOYEE GROUP BY CITY

HAVING COUNT(EID) > 1;

1. Give total salary of each department of EMPLOYEE table. SELECT DEPT, SUM(SALARY) FROM EMPLOYEE

GROUP BY DEPT;

1. Give average salary of each department of EMPLOYEE table without displaying the respective department name

SELECT AVG(SALARY) AS AVG\_SAL\_DEPT FROM EMPLOYEE GROUP BY DEPT;

1. Give minimum salary of employee who belongs to Ahmedabad. SELECT MIN(SALARY) FROM EMPLOYEE

WHERE CITY = 'AHMEDABAD';

1. List the departments having total salaries more than 50000 and located in city Rajkot.

SELECT DEPT, SUM(SALARY) FROM EMPLOYEE WHERE CITY = 'RAKJOT'

GROUP BY DEPT

HAVING SUM(SALARY) > 50000;

# PART B

1. Count the number of employees living in Rajkot. SELECT CITY ,COUNT(EID)FROM EMPLOYEE GROUP BY CITY

HAVING CITY = 'RAKJOT';

1. Display the diﬀerence between the highest and lowest salaries. Label the column DIFFERENCE.

SELECT (MAX(SALARY) - MIN(SALARY)) AS DIFFRANCE FROM EMPLOYEE;

1. Display the total number of employees hired before 1st January, 1991. SELECT COUNT(EID),JDATE

FROM EMPLOYEE GROUP BY JDATE

HAVING JDATE < '1-JAN-1991';

1. Display total salary of each department with total salary exceeding 35000 and sort the list by total salary.

SELECT SUM(SALARY) FROM EMPLOYEE GROUP BY DEPT

HAVING SUM(SALARY) > 35000;

1. List out department names in which more than two employees. SELECT DEPT,COUNT(EID) FROM EMPLOYEE

GROUP BY DEPT

HAVING COUNT(EID) > 2;

# PART C

1. Count the number of employees living in Rajkot or Baroda. SELECT CITY ,COUNT(EID)

FROM EMPLOYEE

WHERE CITY IN('RAKJOT','BARODA') GROUP BY CITY

1. Display the diﬀerence between the highest and lowest salaries. Label the column DIFFERENCE.

SELECT (MAX(SALARY) - MIN(SALARY)) AS DIFFRANCE FROM EMPLOYEE;

1. Display the total number of employees hired before 1st January, 1991 in IT department.

SELECT COUNT(EID) FROM EMPLOYEE WHERE JDATE < '1-JAN-1991' AND DEPT = 'IT'

GROUP BY DEPT

1. Display total salary of each department with total salary exceeding 35000 and sort the list by total salary in descending order.

SELECT SUM(SALARY),DEPT FROM EMPLOYEE GROUP BY DEPT

HAVING SUM(SALARY) > 35000 ORDER BY SUM(SALARY) DESC;

# DBMS LAB 13 SOLUTION

CREATE TABLE STUDENT (

RNO INT,

NAME VARCHAR(30), BRANCH VARCHAR(30)

);

CREATE TABLE RESULT (

RNO INT,

SPI DECIMAL(4,2)

);

CREATE TABLE EMPLOYEE (

EMPLOYEENO VARCHAR(30), NAME VARCHAR(30), MANAGERNO VARCHAR(30)

);

INSERT INTO STUDENT VALUES (101,'Raju','CE'),

(102,'Amit','CE'),

(103,'Sanjay','ME'),

(104,'Neha','EC'),

(105,'Meera','EE'),

(106,'Mahesh','ME')

INSERT INTO RESULT VALUES (101,8.8),

(102,9.2),

(103,7.6),

(104,8.2),

(105,7.0),

(107,8.9)

INSERT INTO EMPLOYEE VALUES ('E01','Tarun',NULL),

('E02','Rohan','E02'),

('E03','Priya','E01'),

('E04','Milan','E03'),

('E05','Jay','E01'),

('E06','Anjana','E04')

SELECT \* FROM EMLOYEEE SELECT \* FROM RESULT SELECT \* FROM STUDENT

# PART A

1. Combine information from student and result table using cross join or Cartesian product.

SELECT S.RNO ,NAME,BRANCH,R.SPI FROM STUDENT S CROSS JOIN RESULT R

SELECT \* FROM STUDENT CROSS JOIN RESULT

1. Display Rno, Name, Branch and SPI of all students SELECT S.RNO,NAME,BRANCH,SPI

FROM STUDENT S

LEFT OUTER JOIN RESULT R ON S.RNO=R.RNO

1. Display Rno, Name, Branch and SPI of CE branch’s student only. SELECT S.RNO,NAME,BRANCH,SPI

FROM STUDENT S

LEFT OUTER JOIN RESULT R ON S.RNO=R.RNO

WHERE S.BRANCH='CE'

1. Display Rno, Name, Branch and SPI of other than EC branch’s student only. SELECT S.RNO,NAME,BRANCH,SPI

FROM STUDENT S

LEFT OUTER JOIN RESULT R ON S.RNO=R.RNO

WHERE S.BRANCH !='EC'

1. Display average result of each branch. SELECT BRANCH ,AVG(SPI)

FROM STUDENT S INNER JOIN RESULT R ON S.RNO = R.RNO GROUP BY BRANCH

1. Display average result of each branch and sort them in ascending order by SPI. SELECT BRANCH,AVG(SPI)

FROM STUDENT S INNER JOIN RESULT R ON S.RNO=R.RNO GROUP BY BRANCH ORDER BY AVG(SPI)

1. Display average result of CE and ME branch. SELECT BRANCH,AVG(SPI)

FROM STUDENT S INNER JOIN RESULT R ON S.RNO=R.RNO

WHERE BRANCH IN ('CE','ME') GROUP BY BRANCH

1. Perform the left outer join on Student and Result tables. SELECT S.RNO,BRANCH,NAME

FROM STUDENT S

LEFT OUTER JOIN RESULT R ON S.RNO=R.RNO

1. Perform the right outer join on Student and Result tables. SELECT R.RNO,BRANCH,NAME

FROM STUDENT S

RIGHT OUTER JOIN RESULT R ON S.RNO=R.RNO

1. Perform the full outer join on Student and Result tables. SELECT S.RNO,BRANCH,NAME

FROM STUDENT S

FULL OUTER JOIN RESULT R ON S.RNO=R.RNO

1. Retrieve the names of employee along with their manager name from the Employee table

SELECT EMP.NAME ,MNR.NAME FROM EMLOYEEE EMP, EMLOYEEE MNR

WHERE EMP.MANAGERNO=MNR.EMPLOYEENO

# PART B

CREATE TABLE PERSON (

PERSONID INT, PERSONNAME VARCHAR(50), DEPARTMENTID INT,

SALARY DECIMAL(7,1), JOININGDATE DATETIME, CITY VARCHAR(50)

);

CREATE TABLE DEPARTMENT (

DEPARTMENTID INT , DEPARTMENTNAME VARCHAR(50), DEPARTMENTCODE VARCHAR(50), [LOCATION] VARCHAR(50)

);

INSERT INTO PERSON VALUES

(101,'Rahul Tripathi',2,56000,'01-JAN-2000','Rajkot'), (102,'Hardik Pandya', 3, 18000,'25-SEP-2001','Ahmedabad'), (103,'Bhavin Kanani', 4, 25000, '14-MAY-2000','Baroda'),

(104,'Bhoomi Vaishnav',1,39000,'08-FEB-2005','Rajkot'), (105,'Rohit Topiya',2,17000,'23-JUL-2001','Jamnagar'), (106,'Priya Menpara', NULL,9000,'18-OCT-2000','Ahmedabad'), (107,'Neha Sharma',2,34000,'25-DEC-2002','Rajkot'),

(108,'Nayan Goswami',3,25000,'01-JUL-2001','Rajkot'),

(109,'Mehul Bhundiya',4,13500,'09-JAN-2005','Baroda'), (110,'Mohit Maru',5,14000,'25-MAY-2000','Jamnagar')

INSERT INTO DEPARTMENT VALUES

(1,'Admin','Adm','A-Block'),

(2,'Computer','CE','C-Block'),

(3,'Civil','CI','G-Block'),

(4,'Electrical','EE','E-Block'),

(5,'Mechanical','ME','B-Block')

SELECT \* FROM PERSON SELECT \* FROM DEPARTMENT

1. Find all persons with their department name & code.

SELECT PERSONNAME, DEPARTMENTNAME, D.DEPARTMENTCODE FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID

1. Give department wise maximum & minimum salary with department name. SELECT DEPARTMENTNAME,MAX(SALARY) AS MAXIMUM,MIN(SALARY) AS MINIMUM

FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY DEPARTMENTNAME

1. Find all departments whose total salary is exceeding 100000. SELECT DEPARTMENTNAME,SUM(SALARY) AS TOTAL\_SALARY FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY DEPARTMENTNAME

HAVING SUM(SALARY)>100000

1. Retrieve person name, salary & department name who belongs to Jamnagar city.

SELECT PERSONNAME,SALARY,DEPARTMENTNAME FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID =D.DEPARTMENTID WHERE CITY='JAMNAGAR'

1. Find all persons who does not belongs to any department. SELECT PERSONNAME,SALARY,DEPARTMENTNAME

FROM PERSON P

RIGHT OUTER JOIN DEPARTMENT D

ON P.DEPARTMENTID =D.DEPARTMENTID WHERE DEPARTMENTNAME IS NULL

1. Find department wise person counts.

SELECT DEPARTMENTNAME ,COUNT(PERSONID) FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID =D.DEPARTMENTID GROUP BY DEPARTMENTNAME

1. Find average salary of person who belongs to Ahmedabad city. SELECT AVG(SALARY) AS AVG\_SALARAY

FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID =D.DEPARTMENTID WHERE CITY='AHMEDABAD'

1. Produce Output Like: <PersonName> earns <Salary> from department

<DepartmentName> monthly. (In Single Column)

SELECT CONCAT( PERSONNAME, ' EARNS ', SALARY ,' FROM DEPARTMENT ', DEPARTMENTNAME, ' MONTHLY ')

FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID

1. List all departments who have no persons. SELECT DEPARTMENTNAME

FROM PERSON P

RIGHT OUTER JOIN DEPARTMENT D

ON P.DEPARTMENTID =D.DEPARTMENTID WHERE PERSONNAME IS NULL

1. Find city & department wise total, average & maximum salaries. SELECT CITY,DEPARTMENTNAME,SUM(SALARY) AS TOTAL\_SALARY,AVG(SALARY)AS AVG\_SLARAY,MAX(SALARY) AS MAXIMUM\_SALARY

FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY CITY,DEPARTMENTNAME

# PART C

1. Display Unique city names.

SELECT DISTINCT CITY FROM PERSON

1. List out department names in which more than two persons. SELECT DEPARTMENTNAME ,COUNT(PERSONNAME)

FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY DEPARTMENTNAME

HAVING COUNT(PERSONNAME)>2

1. Combine person name’s ﬁrst three characters with city name’s last three characters in single column.

SELECT CONCAT(LEFT(PERSONNAME,3),RIGHT(CITY,3)) FROM PERSON

1. Give 10% increment in Computer department employee’s salary. UPDATE PERSON

SET SALARY=SALARY + 0.1\*SALARY FROM PERSON P INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID WHERE DEPARTMENTNAME='COMPUTER'

1. Display all the person name’s who’s joining dates diﬀerence with current date is more than 365 days

SELECT PERSONNAME FROM PERSON

WHERE DATEDIFF(DAY,JOININGDATE,GETDATE()) >365

# DBMS LAB 14 SOLUTION

CREATE TABLE CITY (

CityID INT PRIMARY KEY, Name VARCHAR(50) UNIQUE,

Pincode INT NOT NULL, Remakrs VARCHAR(50) NULL

);

INSERT INTO CITY VALUES (1,'Rajkot',360005,'Good'),

(2,'Surat',335009,'Very Good'),

(3,'Baroda',390001,'Awesome'),

(4,'Jamnagar',361003,'Smart'),

(5,'Junagadh',362229,'Historic'),

(6,'Morvi',363641,'Ceramic');

CREATE TABLE VILLAGE (

VID INT PRIMARY KEY, Name VARCHAR(50),

CityId INT FOREIGN KEY REFERENCES CITY(CityId)

);

INSERT INTO VILLAGE VALUES (101,'Raiya',1),

(102,'Madhapar',1),

(103,'Dodka',3),

(104,'Falla',4),

(105,'Bhesan',5),

(106,'Dhoraji',5);

# PART A

1. Display all the villages of Rajkot city. SELECT C.Name,V.Name FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId WHERE C.Name='Rajkot'

1. Display city along with their villages & pin code. SELECT C.Name,V.Name,C.Pincode FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId;

1. Display the city having more than one village. SELECT C.Name,COUNT(C.CityID) FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId GROUP BY C.Name

HAVING COUNT(C.CityID)>1;

1. Display the city having no village. SELECT COUNT(C.CityID) FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId GROUP BY C.CityId

HAVING COUNT(C.CityID)=0;

1. Count the total number of villages in each city. SELECT C.NAME,COUNT(V.CITYID) FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId GROUP BY C.Name

1. Count the number of cities having more than one village.

SELECT COUNT(\*) FROM

(SELECT C.NAME,COUNT(V.CITYID) AS VILLAGE FROM CITY C INNER JOIN VILLAGE V

ON C.CityID=V.CityId GROUP BY C.Name) AS T WHERE VILLAGE >1

# PART B

CREATE TABLE STUDENT (

RNo INT IDENTITY(101,1) PRIMARY KEY, NAME VARCHAR(50) NULL,

BRANCH VARCHAR(50) DEFAULT 'GENRAL',

SPI DECIMAL(3,2) CHECK(SPI BETWEEN 0 AND 10), BKLOG INT CHECK(BKLOG>=0)

);

CREATE TABLE DEPT (

DID INT IDENTITY(1,1) PRIMARY KEY, DNAME VARCHAR(50)

);

CREATE TABLE CITY (

CID INT IDENTITY(1,1) PRIMARY KEY, CNAME VARCHAR(50)

);

CREATE TABLE EMP (

EID INT IDENTITY(1,1) PRIMARY KEY, ENAME VARCHAR(50),

DID INT NOT NULL FOREIGN KEY REFERENCES DEPTT1(DID), CID INT NOT NULL FOREIGN KEY REFERENCES CITYY1(CID), SALARY DECIMAL(8,2),

EXPERIENCE INT CHECK(EXPERIENCE>=0) DEFAULT 0

);

# PART C

1.

CREATE TABLE DEPT (

Did INT IDENTITY(1,1) PRIMARY KEY, Dname VARCHAR(50) NULL

);

CREATE TABLE COUNTRY (

Cid INT IDENTITY(1,1) PRIMARY KEY, Cname VARCHAR(50) NOT NULL

);

CREATE TABLE STATE (

Sid INT IDENTITY(1,1) PRIMARY KEY, Sname VARCHAR(50) NULL,

Cid INT NOT NULL FOREIGN KEY REFERENCES COUNTRY(Cid)

);

CREATE TABLE DISTRICT (

Destid INT IDENTITY(1,1) PRIMARY KEY, Destname VARCHAR(50) NULL,

Sid INT NOT NULL FOREIGN KEY REFERENCES STATE(Sid)

);

CREATE TABLE CITY (

Cid INT IDENTITY(1,1) PRIMARY KEY, Cname VARCHAR(50) NULL,

Destid INT NOT NULL FOREIGN KEY REFERENCES DISTRICT(Destid)

);

CREATE TABLE EMP (

Eid INT IDENTITY(1,1) PRIMARY KEY, Ename VARCHAR(50) NULL,

Did INT NULL FOREIGN KEY REFERENCES DEPT(Did), Cid INT NULL FOREIGN KEY REFERENCES CITY(Cid), Salary DECIMAL(8,2) NULL,

Experience INT CHECK(Experience >= 0) DEFAULT 0

);

1. RECORD INSERT

INSERT INTO DEPT VALUES ('IT'),

('FINANCE'),

('FOREST'),

('AGRICULTURE'),

('HEALTH');

INSERT INTO COUNTRY VALUES

('India'), ('United States'), ('Australia'),

('Canada'),

('Brazil');

INSERT INTO STATE VALUES

('Gujarat',1),

('California',2),

('New South Wales',3), ('Ontario',4),

('Alagoas',5);

INSERT INTO DISTRICT VALUES

('Ahmedabad',1),

('Los Angeles',2), ('Northern Tablelands',3),

('Toronto Division',4), ('Agua Rasa',5);

INSERT INTO CITY VALUES

('Ahmedabad',1),

('San Francisco',2),

('Sydney',3),

('Toronto',4),

('Sao Paulo',5);

INSERT INTO EMP VALUES ('AMIT',1,2,50000,2),

('AMAN',2,2,20000,1),

('JAY',3,3,80000,6),

('RUDRA',4,4,50000,4),

('AARAV',5,5,90000,5);

1. Display employeename, departmentname, Salary, Experience, City, District, State and country of all employees.

SELECT E1.Ename,D1.Dname,E1.Salary,E1.Experience,C1.Cname, DI1.Destname,S1.Sname,CO1.Cname FROM EMP E1

INNER JOIN DEPT D1 ON E1.Did=D1.Did INNER JOIN CITY C1

ON C1.Destid=E1.Did INNER JOIN STATE S1 ON S1.Cid=C1.Cid

INNER JOIN DISTRICT DI1 ON DI1.Sid=S1.Sid INNER JOIN COUNTRY CO1 ON CO1.Cid=S1.Cid

DBMS LAB 15 SOLUTION

CREATE TABLE STUDENT (

RNO INT PRIMARY KEY, NAME VARCHAR(50), CITY VARCHAR(50),

DID INT FOREIGN KEY REFERENCES DEPARTMENT(DID)

);

CREATE TABLE ACADEMIC (

RNO INT FOREIGN KEY REFERENCES STUDENT(RNO), SPI DECIMAL(4,2) CHECK(SPI BETWEEN 0 AND 10), BKLOG INT CHECK(BKLOG >=0)

);

CREATE TABLE DEPARTMENT (

DID INT PRIMARY KEY , DNAME VARCHAR(50)

);

INSERT INTO DEPARTMENT VALUES (10,'COMPUTER'),

(20,'ELECTRICAL'),

(30,'MECHANICAL'),

(40,'CIVIL')

INSERT INTO STUDENT VALUES (101,'RAJU','RAJKOT',10),

(102,'AMIT','AHMEDABAD',20),

(103,'SANJAY','BARODA',40),

(104,'NEHA','RAJKOT',20),

(105,'MEERA','AHMEDABAD',30),

(106,'MAHESH','BARODA',10)

INSERT INTO ACADEMIC VALUES (101,8.8,0),

(102,9.2,2),

(103,7.6,1),

(104,8.2,4),

(105,7.0,2),

(106,8.9,3)

SELECT \* FROM STUDENT SELECT \* FROM ACADEMIC SELECT \* FROM DEPARTMENT

# PART A

1. Display details of students who are from computer department. SELECT \* FROM STUDENT

WHERE DID IN (SELECT DID FROM DEPARTMENT WHERE DNAME='COMPUTER')

1. Displays name of students whose SPI is more than 8. SELECT NAME FROM STUDENT

WHERE RNO IN (SELECT RNO FROM ACADEMIC WHERE SPI >8)

1. Display details of students of computer department who belongs to Rajkot city. SELECT \* FROM STUDENT

WHERE DID IN (SELECT DID FROM DEPARTMENT WHERE DNAME='COMPUTER') AND CITY='RAJKOT'

1. Find total number of students of electrical department. SELECT COUNT(RNO) FROM STUDENT

WHERE DID IN(SELECT DID FROM DEPARTMENT WHERE DNAME ='COMPUTER')

# PART B

1. Display name of student who is having maximum SPI. SELECT NAME FROM STUDENT

WHERE RNO= (SELECT RNO FROM ACADEMIC

WHERE SPI =(SELECT MAX(SPI) FROM ACADEMIC))

1. Display details of students having more than 1 backlog. SELECT \* FROM STUDENT

WHERE RNO IN (SELECT RNO FROM ACADEMIC WHERE BKLOG >1)

1. Display name of student who is having second highest SPI. SELECT NAME FROM STUDENT

WHERE RNO =(SELECT RNO FROM ACADEMIC

WHERE SPI=(SELECT MAX(SPI) FROM ACADEMIC

WHERE SPI<(SELECT MAX(SPI) FROM ACADEMIC)))

# PART C

1. Display name of students who are either from computer department or from mechanical department.

SELECT NAME FROM STUDENT

WHERE DID IN (SELECT DID FROM DEPARTMENT WHERE DNAME IN('COMPUTER','MECHNAICAL'))

1. Display name of students who are in same department as 102 studying in. SELECT NAME FROM STUDENT

WHERE DID IN (SELECT DID FROM STUDENT WHERE RNO=102)

1. Display name of students whose SPI is more than 9 and who is from electrical department.

SELECT NAME FROM STUDENT

WHERE RNO IN (SELECT RNO FROM ACADEMIC WHERE SPI>9) AND DID = ( SELECT DID FROM DEPARTMENT WHERE DNAME='ELECTRICAL')

# DBMS LAB 16 SOLUTION

CREATE TABLE STUDENT (

STUID INT,

NAME VARCHAR (100), ENROLLMENTNO VARCHAR(12), DIVISION VARCHAR(50),

SEM INT,

BIRTHDATE DATETIME, EMAIL VARCHAR(100), CONTACTNO VARCHAR(50)

);

INSERT INTO STUDENT VALUES

(101,'Naimish Patel','090200107051','BCX-3',3,'06-DEC-

1992','naimishp49@gmail.com','8866205253'), (102,'Firoz A. S.','090200107090','BCY-3',3,'03-MAY-

1994','Firoz.me@gmail.com','8885999922'), (103,'Krunal Vyas','090243107101','BCZ-5',5,'01-MAR-

1984','Krunal.vyas@gmail.com','9990888877'), (104,'Vijay Patel','090200107102','BCX-5',5,'15-FEB-

1985','Vijay.patel123@gmail.com','8787878787'), (105,'Vimal Trivedi','090200107103','BCY-3',3,'20-JAN-

1988','Maulik123@gmail.com','8789564512'); SELECT \* FROM STUDENT;

# PART A

1. Display Name of Student who belongs to either Semester 3 or 5. (Use IN & OR) SELECT NAME FROM STUDENT

WHERE SEM=3 OR SEM=5;

SELECT NAME FROM STUDENT WHERE SEM IN(3,5);

1. Find Student Name & Enrollment No in which Student Id between 103 to 105. SELECT NAME,ENROLLMENTNO FROM STUDENT

WHERE STUID BETWEEN 103 AND 105;

1. Find Student Name & Enrollment No with their Email Who belongs to 5th Semester.

SELECT NAME,ENROLLMENTNO,EMAIL FROM STUDENT WHERE SEM=5;

1. Display All the Details of ﬁrst three students. SELECT TOP 3 \* FROM STUDENT;
2. Display Name & Enrollment no of ﬁrst 30% Students whos contact number ends with 7.

SELECT TOP 30 PERCENT NAME,ENROLLMENTNO FROM STUDENT WHERE CONTACTNO LIKE '%7';

1. Display Unique Semesters.

SELECT DISTINCT SEM FROM STUDENT;

1. Retrieve All the Students who have no Enrollment. SELECT \* FROM STUDENT

WHERE ENROLLMENTNO IS NULL;

1. Find All Students whose Name does not start with V . SELECT \* FROM STUDENT

WHERE NAME NOT LIKE 'V%';

1. Find All Students in which Email Contains 3@G & Only Six Characters. SELECT \* FROM STUDENT

WHERE EMAIL LIKE '%3@G%' AND EMAIL LIKE ' ';

1. Find Out All the Students whose First Name Starts with F And third character must be R.

SELECT \* FROM STUDENT WHERE NAME LIKE 'F\_R%';

# PART B

1. Find All the Student Details whose Contact No contains 877. SELECT \* FROM STUDENT

WHERE CONTACTNO LIKE '%877%';

1. Display Student Name in Which Student belongs to Semester 3 & Contact Number Does Not Contains 8 & 9.

SELECT \* FROM STUDENT

WHERE SEM=3 AND CONTACTNO NOT LIKE '%[8,9]%';

1. Find Students who born after date 01-01-1990. SELECT \* FROM STUDENT

WHERE BIRTHDATE>'01-JAN-1990';

1. Update Division to BCX-5 & Semester 5 whose Student Id Is 102. UPDATE STUDENT

SET DIVISION='BCX-5',SEM=5 WHERE STUID=102;

1. Change the Student s Name to Firoz Sherasiya in which Email is [Firoz.Me@Gmail.Com](mailto:Firoz.Me@Gmail.Com) & Contact No is 888

UPDATE STUDENT

SET NAME='Firoz Sherasiya'

WHERE EMAIL='Firoz.Me@Gmail.Com' AND CONTACTNO LIKE '%888%';

# PART C

1. Add one more Column City Varchar (50) in Student Table. ALTER TABLE STUDENT

ADD CITY VARCHAR(50);

1. Remove All BCX-3 Division Students. DELETE FROM STUDENT

WHERE DIVISION='BCX-3';

1. Change Column Name Email to EmailAddress. SP\_RENAME 'STUDENT.EMAIL','EMAILADDRESS';
2. Remove All the Data from Student Table Using Truncate. TRUNCATE TABLE STUDENT;
3. Write an SQL query to clone a new table Student\_New from Student table with all data.

SELECT \* INTO STUDENT\_NEW FROM STUDENT;

# DBMS LAB 17 SOLUTION

CREATE TABLE EMPLOYEE (

EID INT,

ENAME VARCHAR(50), GENDER VARCHAR(10), JOININGDATE DATETIME, SALARY DECIMAL(8,2), CITY VARCHAR(100)

);

INSERT INTO EMPLOYEE VALUES (1,'Nick','Male','01-JAN-13',4000,'London'),

(2,'Julian','Female','01-OCT-14',3000,'New York'),

(3,'Roy','Male','01-JUN-16',3500,'London'),

(4,'Tom','Male',NULL,4500,'London'),

(5,'Jerry','Male','01-FEB-13',2800,'Sydney'),

(6,'Philip','Male','01-JAN-15',7000,'New York'),

(7,'Sara','Female','01-AUG-17',4800,'Sydney'),

(8,'Emily','Female','01-JAN-15',5500,'New York'),

(9,'Michael','Male',NULL,6500,'London'),

(10,'John','Male','01-JAN-15',8800,'London');

# PART A

1. Display all the employees whose name starts with “m” and 4th character is “h”. SELECT \* FROM EMPLOYEE

WHERE ENAME LIKE 'M H%';

1. Find the value of 3 raised to 5. Label the column as output. SELECT POWER(3,5) AS 'OUTPUT';
2. Write a query to subtract 20 days from the current date. SELECT DATEADD(day,-20,GETDATE())
3. Write a query to display name of employees whose name starts with “j” and contains “n” in their name.

SELECT \* FROM EMPLOYEE WHERE ENAME LIKE 'J%N%';

1. Display 2nd to 9th character of the given string “SQL Programming”. SELECT SUBSTRING('SQL Programming',2,9) AS 'STRING';

# PART B

1. Display name of the employees whose city name ends with “ney” & contains six characters.

SELECT \* FROM EMPLOYEE WHERE CITY LIKE ' NEY';

1. Write a query to convert value 15 to string. SELECT CONVERT(VARCHAR(10),15);

SELECT CAST(15 AS VARCHAR(10))

1. Add department column with varchar (20) to Employee table. ALTER TABLE EMPLOYEE

ADD DEPARTMENT VARCHAR(20);

1. Set the value of department to Marketing who belongs to London city. UPDATE EMPLOYEE

SET DEPARTMENT='MARKETING' WHERE CITY='LONDON'

1. Display all the employees whose name ends with either “n” or “y”. SELECT \* FROM EMPLOYEE

WHERE ENAME LIKE '%N' OR ENAME LIKE '%Y';

SELECT \* FROM EMPLOYEE WHERE ENAME LIKE '%[N,Y]';

# PART C

1. Find smallest integer value that is greater than or equal to 63.1, 63.8 and

-63.2.

SELECT CEILING(63.1) AS 'V1',CEILING(63.8) AS 'V2',CEILING(-63.2) AS 'V3';

1. Display all employees whose joining date is not available. SELECT \* FROM EMPLOYEE

WHERE JOININGDATE IS NULL;

1. Display name of the employees in capital letters and city in small letters. SELECT UPPER(ENAME) AS 'ENAME', LOWER(CITY) AS 'CITY' FROM EMPLOYEE;
2. Change the data type of Ename from varchar (30) to char (30). ALTER TABLE EMPLOYEE

ALTER COLUMN ENAME CHAR(30);

1. Display city wise maximum salary. SELECT CITY,MAX(SALARY) AS SALARY FROM EMPLOYEE

GROUP BY CITY;

# DBMS LAB 18 SOLUTION

CREATE TABLE EMPLOYEE (

EID INT,

ENAME VARCHAR(50), GENDER VARCHAR(10), JOININGDATE DATETIME, SALARY DECIMAL(8,2), CITY VARCHAR(100)

);

INSERT INTO EMPLOYEE VALUES (1,'Nick','Male','01-JAN-13',4000,'London'),

(2,'Julian','Female','01-OCT-14',3000,'New York'),

(3,'Roy','Male','01-JUN-16',3500,'London'),

(4,'Tom','Male',NULL,4500,'London'),

(5,'Jerry','Male','01-FEB-13',2800,'Sydney'),

(6,'Philip','Male','01-JAN-15',7000,'New York'),

(7,'Sara','Female','01-AUG-17',4800,'Sydney'),

(8,'Emily','Female','01-JAN-15',5500,'New York'),

(9,'Michael','Male',NULL,6500,'London'),

(10,'John','Male','01-JAN-15',8800,'London');

# PART A

1. Produce output like <Ename> works at <city> and earns <salary>.

SELECT CONCAT(ENAME,' WORKS AT ',CITY,' AND EARNS ',SALARY) AS 'EMPLOYEE DETAILS' FROM EMPLOYEE;

1. Find total number of employees whose salary is more than 5000. SELECT COUNT(EID) FROM EMPLOYEE

WHERE SALARY>5000;

1. Write a query to display ﬁrst 4 & last 3 characters of all the employees. SELECT LEFT(ENAME,4) , RIGHT(ENAME,3) FROM EMPLOYEE;
2. List the city having total salaries more than 15000 and employees joined after 1st January, 2014.

SELECT CITY FROM EMPLOYEE WHERE JOININGDATE>'01-JAN-2014' GROUP BY CITY

HAVING SUM(SALARY)>15000;

1. Write a query to replace u with oo in Ename. SELECT REPLACE(ENAME,'u','oo') FROM EMPLOYEE;

# PART B

1. Display city with average salaries and total number of employees belongs to each city.

SELECT CITY ,COUNT(EID),AVG(SALARY) AS AVGSALARY FROM EMPLOYEE GROUP BY CITY;

1. Display total salaries paid to female employees. SELECT SUM(SALARY) FROM EMPLOYEE

WHERE GENDER='FEMALE';

1. Display name of the employees and their experience in years.

SELECT ENAME,DATEDIFF(YEAR,JOININGDATE,GETDATE()) FROM EMPLOYEE

1. Remove column department from employee table. ALTER TABLE EMPLOYEE

DROP COLUMN DEPARTMENT

1. Update the value of city from syndey to null. UPDATE EMPLOYEE

SET CITY=NULL WHERE CITY='Sydney';

# PART C

1. Retrieve all Employee name, Salary & Joining date. SELECT ENAME,SALARY,JOININGDATE FROM EMPLOYEE;
2. Find out combine length of Ename & Gender. SELECT LEN(ENAME) + LEN(GENDER) FROM EMPLOYEE;
3. Find the diﬀerence between highest & lowest salary.

SELECT MAX(SALARY) - MIN(SALARY) AS 'DIFFERENCE' FROM EMPLOYEE;

1. Rename a column from Ename to FirstName. SP\_RENAME 'EMPLOYEE.ENAME','FIRSTNAME';
2. Rename a table from Employee to EmpMaster. SP\_RENAME 'EMPLOYEE','EMPMASTER';

# DBMS LAB 19 SOLUTION

CREATE TABLE DEPARTMENT (

DEPARTMENTID INT PRIMARY KEY IDENTITY(1,1), DEPARTMENTNAME VARCHAR(100) NOT NULL UNIQUE, DEPARTMENTCODE VARCHAR(50) NOT NULL UNIQUE, LOCATION VARCHAR(50) NOT NULL

);

INSERT INTO DEPARTMENT VALUES

('Admin','Adm','A-Block'),

('Computer','CE','C-Block'),

('Civil','CI','G-Block'),

('Electrical','EE','E-Block'),

('Mechanical','ME','B-Block');

CREATE TABLE PERSON (

PERSONID INT PRIMARY KEY IDENTITY(101,1), PERSONNAME VARCHAR(100) NOT NULL,

DEPARTMENTID INT FOREIGN KEY REFERENCES DEPARTMENT(DEPARTMENTID), SALARY DECIMAL(8,2) NOT NULL,

JOININGDATE DATETIME NOT NULL, CITY VARCHAR(100) NOT NULL

);

INSERT INTO PERSON VALUES

('Rahul Tripathi',2,56000,'01-JAN-2000','Rajkot'), ('Hardik Pandya',3,18000,'25-SEP-2001','Ahmedabad'), ('Bhavin Kanani',4,25000,'14-MAY-2000','Baroda'), ('Bhoomi Vaishnav',1,39000,'08-FEB-2005','Rajkot'), ('Rohit Topiya',2,17000,'23-JUL-2001','Jamnagar'),

('Priya Menpara',NULL,9000,'18-OCT-2000','Ahmedabad'), ('Neha Sharma',2,34000,'25-DEC-2002','Rajkot'),

('Nayan Goswami',3,25000,'01-JUL-2001','Rajkot'),

('Mehul Bhundiya',4,13500,'09-JAN-2005','Baroda'),

('Mohit Maru',5,14000,'25-MAY-2000','Jamnagar');

# PART A

1. Find all persons with their department name & code.

SELECT PERSONNAME,DEPARTMENTNAME,DEPARTMENTCODE FROM PERSON P INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID;

1. Give department wise maximum & minimum salary with department name. SELECT DEPARTMENTNAME,MIN(SALARY) AS 'MIN SALARY',MAX(SALARY) AS 'MAX SALARY' FROM PERSON P

INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID GROUP BY DEPARTMENTNAME;

1. Find all departments whose total salary is exceeding 100000. SELECT DEPARTMENTNAME FROM DEPARTMENT D

INNER JOIN PERSON P

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY DEPARTMENTNAME

HAVING SUM(SALARY)>100000;

1. Retrieve person name, salary & department name who belongs to Jamnagar city.

SELECT PERSONNAME,SALARY,DEPARTMENTNAME FROM PERSON P INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID WHERE CITY='JAMNAGAR';

1. Find all persons who does not belongs to any department. SELECT PERSONNAME FROM PERSON P

INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID WHERE DEPARTMENTNAME IS NULL;

# PART B

1. Find department wise person counts.

SELECT DEPARTMENTNAME,COUNT(PERSONID) FROM DEPARTMENT D INNER JOIN PERSON P

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY D.DEPARTMENTNAME;

1. Find average salary of person who belongs to Ahmedabad city. SELECT AVG(SALARY) FROM PERSON

WHERE CITY='AHMEDABAD'

1. Produce Output Like: <PersonName> earns <Salary> from department

<DepartmentName> monthly. (In Single Column)

SELECT CONCAT(PERSONNAME,' EARNS ',SALARY,' FROM DEPARTMENT ',DEPARTMENTNAME,' MONTHLY. ') AS 'PERSON DETAILS' FROM PERSON P INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID;

1. List all departments who have no persons. SELECT DEPARTMENTNAME FROM DEPARTMENT D INNER JOIN PERSON P

ON P.DEPARTMENTID=D.DEPARTMENTID WHERE PERSONNAME IS NULL

GROUP BY D.DEPARTMENTNAME;

1. Find city & department wise total, average & maximum salaries.

SELECT CITY,DEPARTMENTNAME,AVG(SALARY) AS 'AVG SALARY',MAX(SALARY) AS 'MAX SALARY' FROM PERSON P

INNER JOIN DEPARTMENT D

ON D.DEPARTMENTID=P.DEPARTMENTID GROUP BY CITY,DEPARTMENTNAME;

# PART C

1. Display Unique city names.

SELECT DISTINCT(CITY) FROM PERSON;

1. List out department names in which more than two persons. SELECT DEPARTMENTNAME FROM DEPARTMENT D

INNER JOIN PERSON P

ON P.DEPARTMENTID=D.DEPARTMENTID GROUP BY DEPARTMENTNAME

HAVING COUNT(PERSONID)>2;

1. Combine person name's ﬁrst three characters with city name s last three characters in single column.

SELECT CONCAT(LEFT(PERSONNAME,3),RIGHT(CITY,3)) FROM PERSON

1. Give 10% increment in Computer department employee s salary. UPDATE PERSON

SET SALARY=SALARY+SALARY\*0.10 FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DEPARTMENTID=D.DEPARTMENTID WHERE DEPARTMENTNAME='COMPUTER'

1. Display all the person name s who s joining dates diﬀerence with current date is more than 365 days.

SELECT PERSONNAME FROM PERSON

WHERE DATEDIFF(DAY,JOININGDATE,GETDATE())>365

# DBMS LAB 20 SOLUTION

CREATE TABLE STUDENT (

Rno Int Primary Key, Name Varchar (50),

Branch Varchar (50),

SPI Decimal (4,2), Bklog Int

);

INSERT INTO STUDENT VALUES (101 ,'Raju','CE',8.80, 0),

(102,'Amit','CE',2.20,3),

(103 ,'Sanjay','ME', 1.50 ,6),

(104,'Neha','EC',7.65,1),

(105,'Meera','EE',5.52,2),

(106,'Mahesh','EC',4.50,3)

# PART A

1. Create a view Personal with all columns. CREATE VIEW PERSONAL AS

SELECT \* FROM STUDENT

1. Create a view Student\_Details having columns Name, Branch & SPI. CREATE VIEW STUDENT\_DETAILS AS

SELECT NAME,BRANCH,SPI FROM STUDENT

1. Create a view Academic having columns RNo, Name, Branch. CREATE VIEW Academic AS

SELECT RNO,NAME,BRANCH FROM STUDENT

1. Create a view Student\_Data having all columns but students whose bklog more than 2.

CREATE VIEW STUDENT\_DATA AS SELECT \* FROM STUDENT WHERE BKLOG>2

1. Create a view Student\_Pattern having RNo, Name & Branch columns in which Name consists of four letters.

CREATE VIEW STUDENT\_PATTERN AS

SELECT RNO,NAME,BRANCH FROM STUDENT WHERE NAME LIKE ' '

1. Insert a new record to Academic view. (107, Meet, ME) INSERT INTO Academic VALUES(107,'MEET','ME')
2. Update the branch of Amit from CE to ME in Student\_Details view. UPDATE STUDENT\_DETAILS

SET BRANCH='ME' WHERE NAME='AMIT'

1. Delete a student whose roll number is 104 from Academic view. DELETE FROM Academic

WHERE RNO=104

# PART B

CREATE TABLE STUDENT (

Rno Int Primary Key,

Name Varchar (50) Not Null, Branch Varchar (50) Not Null, SPI Decimal (4,2) Not Null, Bklog Int Not Null

)

INSERT INTO STUDENT VALUES (101 ,'Raju','CE',8.80, 0),

(102,'Amit','CE',2.20,3),

(103 ,'Sanjay','ME', 1.50 ,6),

(104,'Neha','EC',7.65,1),

(105,'Meera','EE',5.52,2),

(106,'Mahesh','EC',4.50,3)

1. Create a view Personal with all columns. CREATE VIEW PERSONAL AS

SELECT \* FROM STUDENT

1. Create a view Student\_Details having columns Name, Branch & SPI. CREATE VIEW STUDENT\_DETAILS AS

SELECT NAME,BRANCH,SPI FROM STUDENT

1. Create a view Academic having columns RNo, Name, Branch. CREATE VIEW Academic AS

SELECT RNO,NAME,BRANCH FROM STUDENT

1. Create a view Student\_Data having all columns but students whose bklog more than 2.

CREATE VIEW STUDENT\_DATA AS SELECT \* FROM STUDENT WHERE BKLOG>2

1. Create a view Student\_Pattern having RNo, Name & Branch columns in which Name consists of four letters.

CREATE VIEW STUDENT\_PATTERN AS

SELECT RNO,NAME,BRANCH FROM STUDENT WHERE NAME LIKE ' '

1. Insert a new record to Academic view. (107, Meet, ME) INSERT INTO Academic VALUES(107,'MEET','ME')
2. Update the branch of Amit from CE to ME in Student\_Details view. UPDATE STUDENT\_DETAILS

SET BRANCH='ME' WHERE NAME='AMIT'

# PART C

1. Delete a student whose roll number is 104 from Academic view. DELETE FROM Academic

WHERE RNO=104

1. Create a view that displays information of all students whose spi is above 8.5 CREATE VIEW INFO AS

SELECT \* FROM STUDENT WHERE SPI>8.5

1. Create a view that displays 0 backlog students. CREATE VIEW BKLOG AS

SELECT \* FROM STUDENT WHERE BKLOG=0

1. Create a view Computer that displays CE branch data only. CREATE VIEW COMPUTER AS

SELECT \* FROM STUDENT WHERE BRANCH='CE'

1. Create a view Result\_EC that displays the name and SPI of students with SPI less than 5 of branch EC.

CREATE VIEW RESULT\_EC AS SELECT NAME,SPI FROM STUDENT WHERE SPI<5 AND BRANCH='EC'

1. Update the result of student Sanjay to 4.90 in Result\_EC view. UPDATE RESULT\_EC

SET SPI=4.90

WHERE NAME='SANJAY'

1. Create a view Stu\_Bklog with RNo, Name and Bklog columns in which name starts with M and having bklogs more than 5.

CREATE VIEW Stu\_Bklog AS

SELECT RNO,NAME,BKLOG FROM STUDENT WHERE NAME LIKE'M%' AND BKLOG>5

1. Drop Computer view form the database. DROP VIEW COMPUTER

# DBMS LAB 21 SOLUTION

CREATE TABLE DEPARTMENT (

DepartmentID Int Primary Key, DepartmentName Varchar(100) Not Null Unique,

);

CREATE TABLE DESIGNATION (

DesignationID Int Primary Key, DesignationName Varchar(100) Not Null Unique,

);

CREATE TABLE PERSON (

WorkerID Int Primary Key IDENTITY(101,1), FirstName Varchar (100) Not Null, LastName Varchar (100) Not Null,

Salary Decimal (8,2) Not Null, JoiningDate Datetime Not Null,

DepartmentID Int FOREIGN KEY REFERENCES DEPARTMENT(DepartmentID) Null, DesignationID Int FOREIGN KEY REFERENCES DESIGNATION(DESIGNATIONID) Null

);

# PART A

1. Department, Designation & Person Table s INSERT, UPDATE & DELETE Procedures.

INSERT

CREATE PROCEDURE INSERT\_DEPARTMENT @DEPARTMENTID INT,

@DEPNAME VARCHAR(100) AS

BEGIN

INSERT INTO DEPARTMENT VALUES(

END

@DEPARTMENTID, @DEPNAME)

EXEC INSERT\_DEPARTMENT 1,'Admin' EXEC INSERT\_DEPARTMENT 2,'IT' EXEC INSERT\_DEPARTMENT 3,'HR'

EXEC INSERT\_DEPARTMENT 4,'Account'

CREATE PROCEDURE INSERT\_DESIGNATION @DESIGID INT,

@DSIGNAME VARCHAR(100) AS

BEGIN

INSERT INTO DESIGNATION VALUES(

END

@DESIGID, @DSIGNAME)

EXEC INSERT\_DESIGNATION 11,'Jobber' EXEC INSERT\_DESIGNATION 12,'Welder' EXEC INSERT\_DESIGNATION 13,'Clerk' EXEC INSERT\_DESIGNATION 14,'Manager' EXEC INSERT\_DESIGNATION 15,'CEO'

CREATE PROCEDURE INSERT\_PERSON @FN VARCHAR(100),

@LN VARCHAR(100), @SAL DECIMAL(8,2), @JD DATETIME, @DID INT,

@DESID INT AS

BEGIN

INSERT INTO PERSON VALUES(

@FN, @LN, @SAL,

END

@JD, @DID, @DESID)

EXEC INSERT\_PERSON 'Rahul','Anshu',56000,'1990-01-01',1,12 EXEC INSERT\_PERSON 'Hardik','Hinsu',18000,'1990-09-25',2,11

EXEC INSERT\_PERSON 'Bhavin','Kamani',25000,'1991-05-14',NULL,11 EXEC INSERT\_PERSON 'Bhoomi','Patel',39000,'2014-02-20',1,13 EXEC INSERT\_PERSON 'Rohit','Rajgor',17000,'1990-07-23',2,15

EXEC INSERT\_PERSON 'Priya','Mehta',25000,'1990-10-18',2,NULL EXEC INSERT\_PERSON 'Neha','Trivedi',18000,'2014-02-20',3,15

DELETE

CREATE PROCEDURE DELETE\_DEPARTMENT @DEPARTMENTID INT

AS BEGIN

DELETE FROM DEPARTMENT

WHERE DepartmentID = @DEPARTMENTID END

CREATE PROCEDURE DELETE\_DESIGNATION @DESIGID INT

AS BEGIN

DELETE FROM DESIGNATION

WHERE DesignationID = @DESIGID END

CREATE PROCEDURE DELETE\_PERSON @WID INT

AS BEGIN

DELETE FROM PERSON

WHERE WorkerID = @WID END

UPDATE

CREATE PROCEDURE UPDATE\_DEPARTMENT @DEPARTMENTID INT,

@DEPNAME VARCHAR(100) AS

BEGIN

UPDATE DEPARTMENT

SET DepartmentName=@DEPNAME WHERE DepartmentID=@DEPARTMENTID END

CREATE PROCEDURE UPDATE\_DESIGNATION @DESIGID INT,

@DSIGNAME VARCHAR(100) AS

BEGIN

UPDATE DESIGNATION

SET DesignationName=@DSIGNAME WHERE DesignationID=@DESIGID END

CREATE PROCEDURE UPDATE\_PERSON @WID INT,

@FN VARCHAR(100), @LN VARCHAR(100), @SAL DECIMAL(8,2), @JD DATETIME, @DID INT,

@DESID INT AS

BEGIN

UPDATE PERSON

SET FirstName=@FN, LastName=@LN, Salary=@SAL, JoiningDate=@JD,

DepartmentID=@DID, DesignationID=@DESID

WHERE WorkerID = @WID END

1. Department, Designation & Person Table s SELECTBYPRIMARYKEY CREATE PROCEDURE SELECTBYPRIMARYKEY\_DEPARTMENT @DEPARTMENTID INT

AS BEGIN

SELECT \* FROM DEPARTMENT

WHERE DepartmentID=@DEPARTMENTID END

CREATE PROCEDURE SELECTBYPRIMARYKEY\_DESIGNATION @DESIGID INT

AS BEGIN

SELECT \* FROM DESIGNATION

WHERE DesignationID=@DESIGID END

CREATE PROCEDURE SELECTBYPRIMARYKEY\_PERSON @WID INT

AS BEGIN

SELECT \* FROM PERSON

WHERE WorkerID = @WID END

1. Department, Designation & Person Table s (If foreign key is available then do write join and take columns on select list)

CREATE PROCEDURE JOIN\_DEPARTMENT\_DESIGNATION\_PERSON AS

BEGIN

SELECT

P.WorkerID,P.FirstName,P.LastName,P.JoiningDate,P.Salary,DEP.DepartmentNam e,DESIG.DesignationName FROM PERSON P

LEFT JOIN DEPARTMENT DEP

ON DEP.DepartmentID = P.DepartmentID LEFT JOIN DESIGNATION DESIG

ON DESIG.DesignationID = P.DesignationID END

EXEC JOIN\_DEPARTMENT\_DESIGNATION\_PERSON

1. Create a Procedure that shows details of the ﬁrst 3 persons. CREATE PROCEDURE TOP3\_PERSON

AS BEGIN

SELECT TOP 3 \* FROM PERSON END

# PART B

1. Create a Procedure that takes the department name as input and returns a table with all workers working in that department.

CREATE PROCEDURE DEPARTMENT\_INPUT @DEPNAME VARCHAR(100)

AS BEGIN

SELECT P.FirstName FROM PERSON P INNER JOIN DEPARTMENT D

ON D.DepartmentID = P.DepartmentID WHERE D.DepartmentName = @DEPNAME END

EXEC DEPARTMENT\_INPUT 'IT'

1. Create Procedure that takes department name & designation name as input and returns a table with worker s ﬁrst name, salary,joining date & department name.

CREATE PROCEDURE DEPTDESIG\_INPUT @DEPTNAME VARCHAR(100), @DESIGNAME VARCHAR(50)

AS BEGIN

SELECT P.FirstName,P.Salary,P.JoiningDate,D.DepartmentName FROM PERSON P INNER JOIN DEPARTMENT D

ON D.DepartmentID = P.DepartmentID INNER JOIN DESIGNATION DESIG

ON P.DesignationID = DESIG.DesignationID

WHERE D.DepartmentName = @DEPTNAME AND DESIG.DesignationName = @DESIGNAME

END

EXEC DEPTDESIG\_INPUT 'IT','JOBBER'

1. Create a Procedure that takes the ﬁrst name as an input parameter and display all the details of the worker with their department & designation name.

CREATE PROCEDURE FIRSTNAME\_INPUT @FN VARCHAR(100)

AS BEGIN

SELECT P.WorkerID,

P.FirstName, P.LastName, P.JoiningDate, P.Salary, D.DepartmentName,

DESIG.DesignationName FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DepartmentID=D.DepartmentID INNER JOIN DESIGNATION DESIG

ON P.DesignationID = DESIG.DesignationID WHERE P.FirstName = @FN

END

EXEC FIRSTNAME\_INPUT 'HARDIK'

1. Create Procedure which displays department wise maximum, minimum & total salaries.

CREATE PROCEDURE DEPTWISE\_MAXMINTOTAL AS

BEGIN

SELECT D.DepartmentName,MAX(SALARY),MIN(SALARY),SUM(SALARY) FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DepartmentID = D.DepartmentID GROUP BY DepartmentName

END

EXEC DEPTWISE\_MAXMINTOTAL

1. Create Procedure which displays designation wise average & total salaries. CREATE PROCEDURE DESIGNATION\_AVGTOTALSAL

AS BEGIN

SELECT D.DesignationName,MAX(SALARY),MIN(SALARY),SUM(SALARY) FROM PERSON P

INNER JOIN DESIGNATION D

ON P.DesignationID= D.DesignationID GROUP BY D.DesignationName

END

# PART C

1. Create Procedure that Accepts Department Name and Returns Person Count.

CREATE PROCEDURE DEPARTMENTNAME\_INPUT @DN VARCHAR(100)

AS BEGIN

SELECT COUNT(WorkerID) FROM PERSON P INNER JOIN DEPARTMENT D

ON P.DepartmentID = D.DepartmentID

WHERE D.DepartmentName = @DN END

1. Create Procedure that accepts Department Name & Designation as a parameter with given test cases and returns a table with FirstName, LastName, Salary, JoiningDate, DepartmentName & Designation.

CREATE PROCEDURE DEPT\_DESG\_INPUT @DEPNAME VARCHAR(100), @DESIGNAME VARCHAR(100)

AS BEGIN

SELECT P.FirstName,P.LastName,P.Salary,P.JoiningDate FROM PERSON P INNER JOIN DEPARTMENT D

ON P.DepartmentID = D.DepartmentID INNER JOIN DESIGNATION DESIG

ON P.DesignationID = DESIG.DesignationID

WHERE (D.DepartmentName = @DEPNAME OR @DEPNAME IS NULL)

AND (DESIG.DesignationName = @DESIGNAME OR @DESIGNAME IS

NULL) END

EXEC DEPT\_DESG\_INPUT 'IT',NULL

1. Create Procedure that returns DepartmentID, DepartmentName & Count of all person belongs to that department. i.e. 1 | Admin | 2

CREATE PROCEDURE DEPTWISE\_COUNT AS

BEGIN

SELECT D.DepartmentID,D.DepartmentName,COUNT(P.WorkerID) FROM PERSON P

INNER JOIN DEPARTMENT D

ON P.DepartmentID = D.DepartmentID

GROUP BY D.DepartmentID,D.DepartmentName ORDER BY D.DepartmentID

END

EXEC DEPTWISE\_COUNT