Datatypes :

→ Character Datatype
→ Number Datatype
→ Date / Time Datatype

→ Character Datatype :-

(i) Char (size)
(ii) Varchar (size)
(iii) Varchar2 (size)
(iv) Nchar (size)
(v) Nvarchar2 (size)
(vi) Raw (size)
(vii) Long
(viii) Long Raw

→ Number Datatype :-

(i) Number
(ii) Number (size)
(iii) Number (P,S)
(iv) Integer
(v) Float
(vi) Decimal

→ Date / Time Datatype :-

(i) DATE
(ii) TIMESTAMP

Q. Create a table STUDENT(Rollno, Name, DOB, Course, CGPA)

→ TO CREATE TABLE :-

Syntax: CREATE TABLE TABLENAME
(column1 datatype(size),
column2 datatype(size),
⋮
columnn datatype(size));

e.g. CREATE TABLE STUDENT
(Rollno Number(5),
Name Varchar2(20),
DOB date,
course varchar2(5),
CGPA Number(3,2));

→ TO VIEW THE TABLE DESCRIPTION :-

Syntax: DESC TABLENAME;

e.g. DESC STUDENT;

→ To KNOW THE EXISTING TABLES:

Syntax: SELECT * FROM TAB;

→ INSERTING DATA INTO TABLE:

Syntax: INSERT INTO TABLENAME VALUES (data for column1, data for column2, ------, data for column n);

e.g. INSERT INTO CTUDENT VALUES (101, 'Sani', '31-Oct-2000', 'MCA', 8.57);

Another way

Syntax: INSERT INTO TABLENAME VALUES (&column1, &column2, ------, &column n);

e.g. INSERT INTO STUDENT VALUES (&Rollno, &Name, &DOB, &course, &CGPA); ↵

⇒ RETRIVING DATA / DATA RETRIVAL

(i) Retriving whole data from the table.

Syntax: SELECT * FROM TABLENAME;

e.g. SELECT * FROM STUDENT;

(ii) Retriving selected columns.

Syntax: SELECT column1, column2, ----, column k FROM TABLENAME;

e.g. SELECT Rollno, course, CGPA FROM STUDENT;

(iii) Retriving Selected Rows.

Syntax: SELECT * FROM TABLENAME WHERE CONDITION;

eg.1. Retrive the information of students who are sequuring CGPA 8.5 or above.

→ SELECT * FROM STUDENT
WHERE CGPA >= 8.5;

eg.2. Retrive the information of Students who have getting CGPA 7.5 or less and belongs to MCA.

→ SELECT * FROM STUDENT
WHERE CGPA <= 7.5 AND Course = 'MCA';

(iv) Retrive selected Rows and columns.

Syntax: SELECT Column 1, Column 2, ...,
Column K FROM TABLENAME
WHERE CONDITION;

eg. Retrive RollNo, Name, DOB, Course of the Students who are getting CGPA 7.5 or less and studied in MCA.

→ SELECT RollNo, Name, DOB, Course
FROM STUDENT
WHERE CGPA <= 7.5 AND Course = 'MCA';

Assignment -1

Create the following tables

a) STUDENT (Rollno, Name, dob, Cource, CGPA, Subcode)

b) SUBJECT (subcode, subname, credits, L-H, PH)

→ a) CREATE TABLE STUDENT
(Rollno Number(5),
Name Varchar2(20),
dob date,
cource Varchar2(5),
CGPA Number(4,2),
Subcode Varchar2(10));

b) CREATE TABLE SUBJECT
(subcode varchar2(10),
Subname varchar2(15),
Credit Number(2),
LH Number(2),
PH Number(2));

(7) Insert atleast 10 records in each table.

(7) a) INSERT INTO STUDENT VALUES(101, 'Soumya', '31-OCT-2000', 'MCA', 8.57, '10A23');
INSERT INTO STUDENT VALUES(102, 'Dharmanjay', '24-NOV-2000', 'MCA', 8.88, '10A25');

INSERT INTO STUDENT VALUES (103, 'Boomba', '5-JAN-2000', 'MBA', 8.13, '10B22');
INSERT INTO STUDENT VALUES (104, 'SK', '11-OCT-1998', 'MCA', 8.90, '10A30');
INSERT INTO STUDENT VALUES(105, 'Deepak', '11-DEC-2001', 'MSc', 9.98, '10M01');
INSERT INTO STUDENT VALUES(106, 'Prakash', '8-JUN-2000', 'MBA', 8.92, '10B20');
INSERT INTO STUDENT VALUES(107, 'Anup', '15-FEB-2001', 'MCA', 8.99, '10A11');

INSERT INTO STUDENT VALUES(108, 'Rana', '30-MAY-2000', 'MCA', 7.23, '10A18');
INSERT INTO STUDENT VALUES(109, 'Subrat', '16-FEB-1999', 'MSc', 9.58, '10M22');
INSERT INTO STUDENT VALUES(110, 'Raja', '19-OCT-1954', 'MBA', 6.65, '10B66');

b) INSERT INTO SUBJECT VALUES ('10A23', 'DBMS', 4, 4, 3);
INSERT INTO SUBJECT VALUES ('10A25', 'DS', 4, 3, 4);
INSERT INTO SUBJECT VALUES ('10B22', 'Marketing', 3, 3, 2);
INSERT INTO SUBJECT VALUES ('10A30', 'CSA', 2, 3, 0);
INSERT INTO SUBJECT VALUES ('10M01', 'Topology', 3, 3, 2);

INSERT INTO SUBJECT VALUES ('10B20', 'Finance', 2, 3, 1);
INSERT INTO SUBJECT VALUES ('10A11', 'OS', 4, 3, 2);
INSERT INTO SUBJECT VALUES ('10A18', 'DM', 4, 4, 1);
INSERT INTO SUBJECT VALUES ('10M22', 'Prog', 4, 4, 3);
INSERT INTO SUBJECT VALUES ('10B66', 'Business', 2, 4, 1);

(ii) Retrive the information of all students.

SQL> SELECT * FROM STUDENT; ↵

OUTPUT:

ROLLNO	NAME	DOB	COURCE	CGPA	SUBCODE
101	Soumya	31-OCT-00	MCA	8.57	10A23
102	Dharmanjay	24-NOV-00	MCA	8.88	10A25
103	Bramha	05-JAN-00	MBA	8.13	10B22
104	SK	11-OCT-1988	MCA	8.9	10A30
105	Deepak	11-DEC-01	MSc	9.98	10M01
106	Pritesh	06-JUN-00	MBA	8.92	10B20
107	Arup	18-FEB-01	MCA	8.99	10A11
108	Rima	30-MAY-00	MCA	7.23	10A18
109	Subrat	16-FEB-99	MSc	9.58	10M22
110	Raja	19-OCT-84	MBA	6.65	10B66

(iii) Retrive information about subjects.

SQL> SELECT * FROM SUBJECT;

OUTPUT:

SUBCODE	SUBNAME	CREDIT	LH	PH
10A23	DBMS	4	4	3
10A25	DS	4	3	4
10B22	Marketing	3	3	2
10A30	CSA	2	3	0
10M01	Topology	3	3	2
10B20	Finance	2	3	1
10A11	OS	4	3	2
10A18	DM	4	4	1
10M22	C	4	4	3
10B66	Business	2	4	1

(iii) Retrive the information of students who are studing MCA

SQL) SELECT * FROM STUDENT WHERE Cource = 'MCA';

(iv) Retrive the information of students who are born not earlier than 01-01-1985.

SQL) SELECT * FROM STUDENT
WHERE DOB >= '1-JAN-1985';

(v) Retrive Rollno, name, CGPA of all students.

SQL) SELECT Rollno, Name, CGPA FROM STUDENT;

(vi) Retrive Rollno, name, CGPA of all students who secure 8.5 or above CGPA.

SQL) SELECT Rollno, Name, CGPA FROM STUDENT
WHERE CGPA >= 8.5;

(vii) Retrive the subject code and Subjectname along with credit for the subjects having credit <= 3.

SQL) SELECT subcode, subname, credit FROM SUBJECT
WHERE credit <= 3;

(viii) Retrive the information of subjects for which lecture hour is 4 per week and lab hour is 3 hr per week.

SQL) SELECT * FROM SUBJECT
WHERE LH = 4 AND PH = 3;

(x) Retrive the information of students who belongs to MCA dept and secureing CGPA 8.5 or above.

SQL) SELECT * FROM STUDENT
WHERE Cource = 'MCA' AND CGPA >= 8.5;

(x) Retrive the information of students who are not belonging to MBA dept.

SQL) SELECT * FROM STUDENT
WHERE Cource != 'MBA';

(xi) Retrive the information of subjects where lecture hour is equal to practical hours.

SQL) SELECT * FROM ~~STUDENT~~ SUBJECT
WHERE LH = PH;

(xii) Retrive Rollno, name, dob of students who are not enrolled for MCA and secured CGPA 8.0 or above.

SQL) SELECT Rollno, Name, dob FROM STUDENT
WHERE Cource != 'MCA' AND CGPA >= 8.0;

⇒ BETWEEN AND operator :-

In between a range of values are selected where upper and lower limit is cheaked.

eg. Find the employee who are getting salary in the range 10000 to 30000.

SQL) SELECT * FROM EMP
WHERE Sal BETWEEN 10000 AND 30000;

⇒ IN operator :-

→ IN operator is used to match any value from a list in a select statment.
→ This is alternative to multiple OR statment.

eg. Find the employees who are working under dno. 1 or 2 or 3.

multiple OR : SQL) SELECT * FROM EMP
WHERE dno = 1 OR dno = 2 or dno = 3;

IN operator : SQL) SELECT * FROM EMP
WHERE dno IN (1,2,3);

⇒ LIKE operator :-

It is used to determine whether a specific character string matches a specific pattern.
(i) % → it matches zero or more character
(ii) _ (underscore) → it matches any one character.

eg1. Find the employees whose name is starts with letter S.

SQL) SELECT * FROM EMP
WHERE ename LIKE 'S%';

eg2. Find the employes whose second letter of their name starts with letter A.

SQL) SELECT * FROM EMP
WHERE ename LIKE '_a%';

④ IS NULL operator :-

IS NULL operator is used to test for empty values.

eg. SQL) SELECT * FROM STUDENT
WHERE DOB IS NULL;

⑤ DISTINCT keyword :-

The DISTINCT keyword is used to supress the duplicate values in a column.

Syntax: SELECT DISTINCT COLUMN NAME FROM TABLE NAME;

eg. Select different salary from employe table.

SQL) SELECT DISTINCT Sal FROM EMP;

SORTING :

Sorting in a table means specific ordering of rows based on a column / columns.

ASC → For ascending order
DESC → For descending order

Syntax: SELECT COLUMN LIST FROM TABLE NAME
WHERE CONDITION
ORDER BY COLUMN/EXPRESSION [ASC/DECC];

eg1. Sort the salary of employees.

SQL> SELECT SAL FROM EMP
ORDER BY SAL;

eg2. Sort the salary of employees in descending order.

SQL> SELECT SAL FROM EMP
ORDER BY SAL DESC;

eg3. Find the manager with ascending order of their salary.

SQL> SELECT * FROM EMP
WHERE desgn = 'Manager'
ORDER BY SAL;

COLUMN ALIAS :

It is used to give alternative name to a column.

Syntax : SELECT "COLUMN NAME" AS NEWCOLUMNNAME FROM TABLENAME;

eg. Display ename as employee-name.

SQL> SELECT ename AS employee_name FROM EMP;

CONCATENATION :-

concatenation operator is ||.

eg. Display ename, deptno and salary of all employees in the format xxx having department number 1 is getting salary 60000 per month.

SQL> SELECT ename || 'having department number' || deptno || 'is getting salary' || sal || 'per month' FROM EMP;

DML (Data Manupulation Language)

DML commands are

INSERT, UPDATE, DELETE

① UPDATE Statement :

It is used to update the information in a table.

Syntax:

UPDATE TABLE NAME

SET COLUMN1 = new value, COLUMN 2 = New value .... COLUMN N = new value

WHERE CONDITION;

⊖ DELETE STATEMENT :-

It is used to delete data from a table.

→ Delete all records from table

Syntax: DELETE FROM TABLE NAME;

→ Delete a particular record from table.

Syntax: DELETE FROM TABLE NAME
WHERE CONDITION;

⊖ COMMIT :-

After a DML statement is executed, commit is used to save the changes by the DML statement on the database.

Syntax:

SQL> COMMIT;

⊖ ROLLBACK :-

After a DML statement is executed, the rollback undo the change on the database.

Syntax:

SQL> ROLLBACK;

DDL (Data Definition Language)

The DDL statements are

CREATE, ALTER, DROP, TRUNCATE, COMMENT, RENAME

DROP Statment :-

DROP table command is used to delete a table on database permanently along with its structure.

Syntax: DROP TABLE TABLE NAME;

eg. DROP TABLE STUDENT;

TRUNCATE Statment :-

TRUNCATE is used to delete the data from a table, but structure remains present.

Syntax: TRUNCATE TABLE TABLE NAME;

eg. TRUNCATE TABLE STUDENT;

ALTER TABLE Statment :-

It is used to modify structure of a table.

⇒ Adding a new column to an existing table.

Syntax: ALTER TABLE TABLENAME
ADD (new column1 datatype(size),
new column2 datatype(size),
:
new column n datatype(size));

eg. Add a column contact no. in employee table.

SQL> ALTER TABLE EMP
ADD (Contact_no Number(10));

→ DROP a column from a table.

Syntax: ALTER TABLE TABLE NAME
DROP COLUMN COLUMN NAME;

eg. DROP contact_no.

SQL> ALTER TABLE EMP
DROP COLUMN Contact_no;

⇒ RENAMEING a column to a new column.

Syntax: ALTER TABLE TABLE NAME
RENAME COLUMN oldcolumnname TO Newcolumnname;

eg: ALTER TABLE EMP
RENAME COLUMN eid TO emp_id;

⇒ MODIFING datatype of a column.
If data present in a column, it is not advisable to
→ change the datatype
→ decrease the size of datatype.

Syntax: ALTER TABLE TABLE NAME
MODIFY (column Name new datatype(size));

eg: ALTER TABLE EMP
MODIFY (sal NUMBER(5));

⇒ CREATING a table from another table.

(i) Creating a table from an existing table:

Syntax: CREATE TABLE TABLE NAME
AS SELECT * FROM EXISTING TABLE;

e.g: CREATE TABLE STUDENT_1
AS SELECT * FROM STUDENT;

→ Copying only the structure of an existing table.

Syntax: CREATE TABLE TABLE NAME
AS SELECT * FROM ~~EXISTENT NA~~ EXISTING TABLE NAME
WHERE FALSE CONDITION;

eg: CREATE TABLE STUDENT-2
AS SELECT * FROM STUDENT
WHERE 1=2;

(ii) Creating table with Specific column from an existing table.

Syntax: CREATE TABLE TABLENAME (Column1, Column2 ... Columnk)
AS SELECT Column1, Column2, ... Columnk
FROM EXISTING TABLENAME;

eg: CREATE TABLE EMP-1 (eid, ename, sal, dno)
AS SELECT eid, ename, sal, dno FROM EMP;

14/2/22