

## **Experiment No: 1**

**Environment:** Microsoft Windows

**Tools/ Language:** Oracle/SQL

**Objective:** Write the SQL queries for data definition and data manipulation language.

### **Theory & Concepts:**

#### **Introduction about SQL-**

SQL (Structured Query Language) is a nonprocedural language, you specify what you want, not how to get it. A block structured format of English key words is used in this Query language.

It has the following components.

- DDL (Data Definition Language)
- DML (DATA Manipulation Language)
- View definition
- Transaction Control
- Embedded SQL and Dynamic SQL
- Integrity
- Authorization

#### **Data Definition Language-**

The SQL DDL allows specification of not only a set of relations but also information about each relation, including-

- Schema for each relation
- The domain of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.
- The security and authorization information for each relation.
- The physical storage structure of each relation on disk.

#### **Domain types in SQL-**

**The SQL standard supports a variety of built in domain types, including-**

- Char (n)- A fixed length character length string with user specified length .
- Varchar (n)- A variable character length string with user specified maximum length n.
- Number (p, d)-A Fixed point number with user defined precision.
- Real, double precision- Floating point and double precision floating point numbers with machine dependent precision.
- Float (n)- A floating point number, with precision of at least n digits.
- Date- A calendar date containing a (four digit) year, month and day of the month.
- Time- The time of day, in hours, minutes and seconds Eg. Time '09:30:00'.

### **DDL statement for creating a table-**

#### **Syntax-**

```
Create table tablename  
(columnnamedatatype(size), columnnamedatatype(size));
```

### **Creating a table from a table-**

#### **Syntax-**

```
CREATE TABLE TABLENAME  
[(columnname, columnname, .....)]  
AS SELECT columnname, columnname.....FROM tablename;
```

Rules:

1. Oracle reserved words cannot be used.
2. Underscore, numerals, letters are allowed but not blank space.
3. Maximum length for the table name is 30 characters.
4. 2 different tables should not have same name.
5. We should specify a unique column name.
6. We should specify proper data type along with width.
7. We can include “not null” condition when needed. By default it is ‘null’.

### **Insertion of data into tables-**

#### **Syntax-**

```
INSERT INTO tablename  
[(columnname, columnname, .....)]  
Values(expression, expression);
```

### **Inserting data into a table from another table:**

#### **Syntax-**

```
INSERT INTO tablename  
SELECT columnname, columnname, .....  
FROM tablename;
```

### **Insertion of selected data into a table from another table:**

#### **Syntax-**

```
INSERT INTO tablename  
SELECT columnname, columnname.....  
FROM tablename  
WHERE columnname= expression;
```

### **Retrieving of data from the tables-**

#### **Syntax-**

```
SELECT * FROM tablename;
```

### **The retrieving of specific columns from a table-**

#### **Syntax-**

```
SELECT columnname, columnname, ...  
FROM tablename;
```

### **Elimination of duplicates from the select statement-**

#### **Syntax-**

```
SELECT DISTINCT columnname, columnname  
FROM tablename;
```

### **Selecting a data set from table data-**

#### **Syntax-**

```
SELECT columnname, columnname  
FROM tablename  
WHERE searchcondition;
```

**The SELECT DISTINCT \*** SQL syntax scans through entire rows, and eliminates rows that have exactly the same contents in each column.

#### **Syntax:**

```
SELECT DISTINCT *  
FROM TableName;
```

### **DML ( Data Manipulation Language)**

Data manipulation is

- The retrieval of information stored in the database.
- The insertion of new information into the database.
- The deletion of information from the database.
- The modification of information stored by the appropriate data model. There are basically two types.
  - (i) **Procedural DML:-** require a user to specify what data are needed and how to get those data.
  - (ii) **Non Procedural DML :** require a user to specify what data are needed without specifying how to get those data.

### **Updating the content of a table:**

In creation situation we may wish to change a value in table without changing all values in the tuple . For this purpose the update statement can be used.

Update table name

Set columnname = expression, columnname =expression.....

Where columnname = expression;

### **Deletion Operation:-**

We can delete whole tuple ( rows) we can delete values on only particulars attributes.

#### **Deletion of all rows**

##### **Syntax:**

Delete from tablename;

#### **Deletion of specified number of rows**

##### **Syntax:**

Delete from table name  
where search\_condition;

### **Computation in expression lists used to select data:-**

+	Addition	-	Subtraction
*	multiplication	**	exponentiation
/	Division	()	Enclosed operation

Renaming columns used with Expression Lists: - The default output column names can be renamed by the user if required

##### **Syntax:**

```
Select columnname    result_columnname, columnname  
result_columnname  
from tablename;
```

**Logical Operators:**

The logical operators that can be used in SQL sentences are

AND	all of must be included
OR	any of may be included
NOT	none of could be included

**Range Searching:** Between operation is used for range searching.

**Pattern Searching:**

The most commonly used operation on string is pattern matching using the operation 'like'. We describe patterns by using two special characters.

- Percent (%) ; the % character matches any substring we consider the following examples.
  - 'Perry %' matches any string beginning with perry
  - '% idge %' matches any string containing 'idge' as substring.
  - ' - - - ' matches any string exactly three characters.
  - ' - - - %' matches any string of at least three characters.

**Ordering tuples in a particular order:**

- The 'order by' clause is used to sort the table data according to one or more columns of the table.
- The table rows are ordered in ascending order of the column values by default. The keyword used for the same is 'asc'. For sorting the table data according to colname in descending order, keyword 'desc' is used.

Example: select colname1, colname2,... from tablename where search condition order by colname1 asc/desc, colname2 asc/desc,...;

## Practical Assignment - 1

**Department:** Computer Engineering & Applications

**Course:** B.Tech. (CSE)

**Subject:** Database Management System Lab (CSE3083)

**Year:** 2<sup>nd</sup>

**Semester:** 3<sup>rd</sup>



Create these tables which consist of following attributes

### College

Column Name	Data type	Size
cName	varchar2	10
state	varchar2	10
enrollment	int	

### Student

Column Name	Data type	Size
sID	int	
sName	varchar2	10
GPA	number	2,1
sizeHS	int	
DoB	date	

### Apply

Column Name	Data type	Size
sID	int	
cName	varchar2	10
major	varchar2	20
decision	char	1

**About Database:** This database is College Application Database which contain 3 tables.

First one is **Student** that contain information of student such as ID of Student, his name, GPA that he/she scored, size of his High School class i.e. number of students in his/her high school class and student's Date of Birth.

Second table in database is **College** this table contains college information such as name of college, state where it is situated, and its enrollment i.e. number seats in that college.

Third table contains data about applications and this table is named **Apply** each row of this table will contain information about one application. Each application has

- **sID:** This will contain ID (similar to Roll No.) of student who is applying. [Each Application only contain sID of Applicant all other information about student such as his name or GPA can be check from **Student** table]
- **cName:** Name of college where applicant is applying [similar to sID all the other information about college such as its state or enrollment can be retrieve from **College** table]
- **major:** it is the major in which applicant is applying e.g. CS, EE, biology etc.
- **decision:** it is Y or N shows Acceptance or Rejection of Application

Insert the following data to these tables:

**Student**

sID	sName	GPA	sizeHS	DoB
123	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	7-Apr-95
345	Craig	3.5	500	4-Feb-95
456	Doris	3.9	1000	24-Jul-97
567	Edward	2.9	2000	21-Dec-96
678	Fay	3.8	200	27-Aug-96
789	Gary	3.4	800	8-Oct-96
987	Helen	3.7	800	27-Mar-97
876	Irene	3.9	400	7-Mar-96
765	Jay	2.9	1500	8-Aug-98
654	Amy	3.9	1000	26-May-96
543	Craig	3.4	2000	27-Aug-98

**Apply**

sID	cName	major	decision
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

**College**

cName	state	enrollment
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040

**State of the SQL \* PLUS Queries for each of the following:**

1. List the student name, dob from student table
2. List the name of student scoring more than 3.7 in GPA.
3. List the name of student whose High School size is atleast 1000 and born after 1996. [**Hint:** check DoB greater than 31<sup>st</sup> December, 1996]
4. List the name of student who are scoring GPA in between 2.9 and 3.9
5. List all the details of colleges who situated in MA.
6. List the students who are scored more than 2.0 but less than 3.5.
7. List the students who have born after 1<sup>st</sup> Jul 96 in the order of the Date of Birth.
8. List the sID, cName, decision of applications that are accepted.
9. List the sID, cName of applications which are filled at Stanford.
10. List the colleges that that has enrollment greater than 10001.
11. List the colleges not in California.
12. List names of all student who came from high school having size greater than 17000 and scored GPA less than 3.8.
13. Display the description of the Student table.
14. Display the details of all students.

15. Display unique majors.
16. List the student names those are having three characters in their Names.
17. List the student names those are starting with 'H' and with five characters.
18. List the student names those are having third character and fifth char. must be 'e'.
19. List the student names ending with 'y'.
20. List the Students in the order of their GPA.
21. List the details of the students in order of the ascending of GPA and descending of DoB.
22. List the sIDs of student who apply in either 'Stanford', 'Cornell' or 'MIT' college.
23. Delete all applications filled at Stanford (*Choose table wisely*)
24. Delete the college Stanford from college table.
25. Modify the GPA of all students by giving 10% raise in their GPA.
26. Increment the GPA of the students by 1.5 whose GPA is less than 3.5 and belong to High School having size greater than 1500.
27. Delete the students who have scored less than 3.2 GPA.

### Exercise

Determine the appropriate datatype:

#### Dept

deptno	dname	loc
1	ACCOUNTING	ST LOUIS
2	RESEARCH	NEW YORK
3	SALES	ATLANTA
4	OPERATIONS	SEATTLE

Column name	datatype	size
Deptno		
Dname		
loc		

#### Employee

empno	ename	job	mgr	hiredate	sal	comm	dept
1	JOHNSON	ADMIN	6	12/17/1990	18000	(null)	4
2	HARDING	MANAGER	9	2/2/1998	52000	300	3
3	TAFT	SALES I	2	1/2/1996	25000	500	3
4	HOOVER	SALES I	2	4/2/1990	27000	(null)	3
5	LINCOLN	TECH	6	6/23/1994	22500	1400	4
6	GARFIELD	MANAGER	9	5/1/1993	54000	(null)	4
7	POLK	TECH	6	9/22/1997	25000	(null)	4
8	GRANT	ENGINEER	10	3/30/1997	32000	(null)	2
9	JACKSON	CEO	(null)	1/1/1990	75000	(null)	4
10	FILLMORE	MANAGER	9	8/9/1994	56000	(null)	2
11	ADAMS	ENGINEER	10	3/15/1996	34000	(null)	2
12	WASHINGTON	ADMIN	6	4/16/1998	18000	(null)	4

13	MONROE	ENGINEER	10	12/3/2000	30000	(null)	2
14	ROOSEVELT	CPA	9	10/12/1995	35000	(null)	1
15	HANCOCK	SALES I	2	3/2/1990	27500	(null)	3

### Solve the following queries:

- Q1. Employee Name and Hire Date sorted by Hire Date (Recent to Old).
- Q2. Employee Name and Job sorted by Job (Alphabetically)
- Q3. Employee Name and Job for all Engineers, sorted by Employee Name Alphabetically
- Q4. Job, Employee Name, Salary and Commission for employees with salary over 50000 sorted by Salary (Largest to Smallest).
- Q5. Job, Employee Name, Salary and Commission for employees with a Commission sorted by Salary (Largest to Smallest).
- Q6. Job, Employee Name, Salary and Commission for employees whose name starts with the letter H
- Q7. Job, Employee Name, Salary and Commission for employees whose name starts with the letter H and who do not get commission.
- Q8. Job, Employee Name for employees in Dept No. 3.
- Q9. Dept Name and Loc for employees in Dept No. 3.
- Q10. Job, Employee Name, Dept, Salary sorted first by Dept (Smallest to Largest) and then Salary (Largest to Smallest)

Column name	datatype	size
<b>empno</b>		
<b>ename</b>		
<b>job</b>		
<b>mgr</b>		
<b>hiredate</b>		
<b>sal</b>		
<b>comm</b>		
<b>dept</b>		

### Pre Experiment Questions

1. What is Relation?
2. What are attributes?
3. What are different DML commands?
4. What is the need of pattern matching?

### Post Experiment Questions

1. Difference between DDL and DML?
2. How to create a Table?
3. What is the difference between truncate and delete?
4. How we use IN command?