**UNIVERSITY COLLEGE OF ENGINEERING,**

**BIT CAMPUS**

**TRICHIRAPPALLI**

**MASTER MANUAL**

****

**DEPARTMENT OF CSE/IT**

**CS 6312-DATABASE MANAGEMENT SYSTEM LABORATORY**

**YEAR: II (2014-2018) SEMESTER: III**

**INTRODUCTION TO SQL**

SQL was invented and developed by IBM in early 1970‟s. SQL stands for

Structured Query Language. IBM was able to demonstrate how to control relational database using SQL. The SQL implemented by ORACLE CORPORATION is 100% compliant with the

ANSI/ ISO standard SQL data language. Oracle‟s database language is SQL, which is used for storing and retrieving information in Oracle. A table is a Primary database object of SQL that is used to store data. A table that holds data in the form of rows and columns.

In order to communicate with the database, SQL supports the following categories of commands: -

Data Definition Language - create, alter and drop commands.

Data Manipulation Language - insert, select, delete and update commands.

Transaction Control Language - commit, savepoint and rollback commands.

Data control Language - grant and revoke commands.

**Data Definition Language:**

DDL is used to create an object ( table ), alter the structure of an object and also to drop the object created. A table is a unit of storage that holds data in the form of rows and columns. DDL is used for table definition.

**Data Manipulation Language:**

DML commands are most frequently used SQL commands. They are used

to query and manipulate existing objects like tables.

**Transaction Control Language:**

A transaction is a logical unit of work. All changes made to the database can be referred to as a transaction. Transaction changes can be made permanent to a database only if they are committed. A transaction begins with an executable SQL statement ends explicitly with either rollback or commit commands and implicitly i.e., automatically, when a DDL statement is used.



The following are the benefits of SQL:

* Non-procedural Language, because more than one record can be accessed rather than one record at a time.
* It is the common language for all relational databases. In other words it is portable and it requires very few modifications so that it can work on other databases.
* Very simple commands for Querying, inserting, deleting and modifying data and

objects.

**SQL Vs SQL \*Plus:**

SQL is a standard language common to all relational databases. SQL is a database language used for storing and retrieving data from the database. Most relational Database Management Systems provide extensions to SQL to make it easier for application developers.

SQL \*Plus is an Oracle specific Program which accepts SQL commands and PL/SQL blocks and executes them. SQL \*Plus enables manipulation of SQL commands and PL/SQL blocks. It also performs many additional tasks as well. Through SQL \*Plus we can

* enter, edit, store, retrieve and run SQL commands and PL/SQL blocks.
* format, perform calculations, store and print query results in the form of reports.
* list column definitions for any table.
* Access and copy data between SQL databases.
* Send messages to and accept responses from an end user.



**SQL (Structured Query Language**) is a standard supported by all the popular relational database management systems in the market place. The basis data structure in RDBMS is a table. SQL provides you the features to define tables, define constraints on tables, query for data in table, and change the data in table by adding, modifying, and removing data. SQL also supports grouping of data in multiple rows, combining tables and other features. All these put together, SQL is a high-level query language standard to access and alter data in RDBMS.

Querying for Data:

The typical syntax to query for data is;

SELECT clause

FROM clause

WHERE clause

There are of course a few other additions to these standard clauses.

* GROUP BY
* HAVING

Querying can be used to: To retrieve existing data from database.

* Get all data from the table
* Get selected columns from the table.
* Get selected rows from the table.
* Get selected columns of selected rows from the table.
* Get computed columns using char, number, data functions, general functions, and aggregating functions.
* Get data in multiple rows grouped on an aggregating function applied on one or more columns.
* Select specific aggregating data on multiple rows using having clause.
* Apply set operations like Union and Intersection on data sets of the same cardinality and type.
* Get data from multiple tables using Cartesian product, equality join, un-equal join, and outer join.
* Create views on physical data.

Data Definition Language (DDL): To do with altering the structure of data base.

* Create tables.
* Create Indexes on tables.
* Alter tables
* Add constraints to tables.
* Drop tables.
* Truncate tables.
* Drop indexes.
* Create sequences.

Data Manipulation Language (DML): To do with adding, updating, and deleting data from tables.

* Insert Data
* Update Data
* Delete Data
* Merge Tables.

Data Control Language (DCL): To do with controlling access to data in tables.

* Grant Permissions
* Revoke Permissions
* Create and update roles.
* Add users to roles.

**Ex.No.1. CREATION OF A DATABASE AND WRITING SQL QUERIES TO** **RETRIEVE INFORMATION FROM THE DATABASE**.

**Data Definition Language:**

DDL (Data Definition Language) statements are used to create, change the objects of a database. Typically a database administrator is responsible for using DDL statements or production databases in a large database system.

The commands used are:

* Create - It is used to create a table.
* Alter - This command is used to add a new column, modify the existing column definition and to include or drop integrity constraint.
* Drop - It will delete the table structure provided the table should be empty.
* Truncate - If there is no further use of records stored in a table and the structure has to be retained, and then the records alone can be deleted.
* Desc - This is used to view the structure of the table.

**SYNTAX:**

**CREATE TABLE:**

* create table <table name> (fieldname-1 data type constraints if any,fieldname-2 data type constraints if any,……. fieldname-n data type constraints if any);
* create table <table name> as (select(attribute-list) from <existing table name>);

**ALTER TABLE:**

* alter table <table name> add/modify (fieldname-1 datatype,fieldname-2 data type,….. fieldname-n data type );
* alter table drop column column name;

Table altered.

**DESCRIBING TABLE:**

* desc <table name>;

**CHANGING NAME OF AN OBJECT:**

To change the name of a table, view, sequence, or synonym, execute the rename statement.

**Syntax:** rename old name to new name;

**TRUNCATE:**

The truncate table statement

● removes all rows from a table

● Release the storage space used by that table

**Syntax:** truncate table <table name>;

**DROP TABLE:**

1. All data and structure in the table is deleted
2. Any pending transactions are committed.
3. All indexes are dropped.

**Syntax:** drop table <table name>;

Table dropped.

**SAMPLE OUTPUT:**

SQL> create table tbl03 (sno number (2), regno number (12), name varchar2 (10), age number (2), marks number (2));

|  |  |  |
| --- | --- | --- |
| Table created. |  |  |
| SQL> desc tbl03; |  |  |
| Name | Null? | Type |
| -------------------- | ---------------------- -------------------------- | |
| SNO |  | NUMBER (2) |
| REGNO |  | NUMBER (12) |
| NAME |  | VARCHAR2 (10) |
| AGE |  | NUMBER (2) |
| MARKS |  | NUMBER (2) |
| SQL> alter table tbl03 add (total number (3)); | | |
| Table altered. |  |  |
| SQL> desc tbl03; |  |  |
| Name | Null? | Type |
| -------------------- | --------------------- | -------------------- |
| SNO |  | NUMBER (2) |
| REGNO |  | NUMBER (12) |
| NAME |  | VARCHAR2 (10) |
| AGE |  | NUMBER (2) |
| MARKS |  | NUMBER (2) |
| TOTAL |  | NUMBER (3) |

SQL> insert into tbl03 values (&sno, &regno,'&name', &age, &marks, &total); Enter value for sno: 02



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enter value for regno: 003 | | | |  |  |  |
| Enter value for name: Abishek | | | |  |  |  |
| Enter value for age: 18 | | |  |  |  |  |
| Enter value for marks: 99 | | |  |  |  |  |
| Enter value for total: 599 | | |  |  |  |  |
| old | 1: insert into tbl03 values (&sno, &regno,'&name', &age, &marks, &total) | | | | | |
| new | 1: insert into tbl03 values (02, 003,'Abishek', 18, 99,599) | | | | |  |
| 1 row created. | | |  |  |  |  |
| SQL> insert into tbl03 values (22, 023,'Isai', 18, 98,598); | | | | | |  |
| 1 row created. | | |  |  |  |  |
| SQL> select \* from tbl03; | | | |  |  |  |
|  | SNO | REGNO | NAME | AGE | MARKS | TOTAL |
| ---------- | | ------------- | -------------------- ---------- | | ---------------- -------------- | |
|  | 2 | 003 | Abishek | 18 | 99 | 599 |
|  | 22 | 023 | Iasi | 18 | 98 | 598 |
| SQL> select name from tbl03; | | | |  |  |  |
| NAME | |  |  |  |  |  |
| ---------- | |  |  |  |  |  |
| Abishek | |  |  |  |  |  |
| Isai |  |  |  |  |  |  |
| SQL> select \* from tbl03 where total=599; | | | | |  |  |
|  | SNO | REGNO | NAME | AGE | MARKS | TOTAL |
| ---------- ------------- | | | ----------------- ---------- ---------------- -------------- | | | |
|  | 2 | 003 | Abishek | 18 | 99 | 599 |





Page 12

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SQL> update tbl03 set age=20 where age=18; | | | |  |  |
| 2 rows updated. | |  |  |  |  |
| SQL> select \* from tbl03; | |  |  |  |  |
| SNO | REGNO | NAME | AGE | MARKS | TOTAL |
| ---------- ------------- -------------------- ---------- ---------------- -------------- | | | | | |
| 2 | 003 | Abishek | 18 | 99 | 599 |
| 22 | 023 | Isai | 20 | 98 | 598 |
| SQL> update tbl03 set total=95 where name='Isai'; | | | |  |  |
| 1 row updated. | |  |  |  |  |
| SQL> delete from tbl03 where sno=22; | | |  |  |  |
| 1 row deleted. | |  |  |  |  |
| SQL> select \* from tbl03; | |  |  |  |  |
| SNO | REGNO | NAME | AGE | MARKS | TOTAL |
| ---------- ------------- -------------------- ---------- ---------------- -------------- | | | | | |
| 2 003 | Abishek | 18 | 99 | 599 |  |
| SQL> truncate table tbl03; | |  |  |  |  |
| Table truncated. | |  |  |  |  |
| SQL>rename tbl03 to tbl03z; | |  |  |  |  |
| Table renamed. | |  |  |  |  |
| SQL> drop table tbl03z; | |  |  |  |  |
| Table dropped. | |  |  |  |  |
| SQL> select \* from tbl03z; | |  |  |  |  |
| select \* from tbl03z | |  |  |  |  |
| \* |  |  |  |  |  |
| ERROR at line 1: | |  |  |  |  |
|  | | |  |  |  |
|  | | |  |  |  |
|  | | |  |  | Page 13 |

ORA-00942: table or view does not exist

SQL> desc tbl03z;

\*

ERROR:

ORA-04043: object tbl03z does not exist

**RESULT :**

Thus creation of a database and writing SQL queries to retrieve information from the database was created and the output was verified.

**Ex.No.1b. PERFORMING INSERTION, DELETION, MODIFYING, ALTERING, UPDATING AND VIEWING RECORDS BASED ON CONDITIONS**

**Data Manipulation Language**

DML commands are the most frequently used SQL commands and is used to query and manipulate the existing database objects. Some of the commands are

1. Insert
2. Select
3. Update

4. Delete

**SYNTAX :**

**INSERT:** This is used to add one or more rows to a table. The values are separated by commasand the data types char and date are enclosed in apostrophes. The values must br entered in the same order as they are defined.

**Inserting a single row into a table:**

insert into <table name> values(fieldvalue-1,fieldvalue-2,…,fieldvalue-n);

**Inserting more than one record using a single insert command:**

insert into <table name> values(&fieldname-1,&fieldname-2,…&fieldname-n);

**Skipping the fields while inserting:**

insert into <tablename(coln names to which datas to b inserted)> values (list of values); Other way is to give null while passing the values.

insert into <table name>(select(att\_list) from <existing table name>);

**SELECT:** - It is used to retrieve information from the table.it is generally refered to as queryingthe table. We can either display all columns in a table or only specify column from the table.

SELECT(att\_list) FROM <table name> [WHERE <condition/expression>];

**Retrieval of all columns from a table:**

Select \* from tablename; // This query selects all rows from the table.

**Retrieval of specific columns from a table:**It retrieves the specified columns from the table.

Select column\_name1, …..,column\_namen from table name;

**Elimination of duplicates from the select clause:** It prevents retriving the duplicated values

.Distinct keyword is to be used.

Select DISTINCT col1, col2 from table name;

**Select command with where clause:** To select specific rows from a table we include ’where’ clause in the select command. It can appear only after the’from’ clause.

**Syntax:** Select column\_name1, …..,column\_namen from table name where condition;

**Select command with order by clause:**

**Syntax:** Select column\_name1, …..,column\_namen from table name where condition orderby colmnname;

**Select command to create a table:**

**Syntax:** create table tablename as select \* from existing\_tablename;

**Select command to insert records:**

**Syntax:** insert into tablename ( select columns from existing\_tablename);

**UPDATE** - It is used to alter the column values in a table. A single column may be updated ormore than one column could be updated.

update <table name> set (fieldname-1 = value, fieldname-2 = value,…,fieldname-n = value) [WHERE <condition/expression>];

**DELETE** - After inserting row in a table we can also delete them if required. The deletecommand consists of a from clause followed by an optional where clause.

delete from <table name> [where <condition/expression>];

**ALTER TABLE**

* alter table table\_name add column\_name datatype
* alter table <table name> add (fieldname-1 datatype,fieldname-2 datatype,….. fieldname-n datatype );
* alter table <table name> modify (fieldname-1 data type (new size));
* alter table drop column column name;

Table altered.

**DATA CONTROL LANGUAGE (DCL) COMMANDS**

|  |  |  |
| --- | --- | --- |
| SQL> select \* from person; | | 3 Raj Anita Chennai 27  4 Kumar Ashok Coimbatore 30  5 Hinn Benny Britain 55  6 Prakash Bhaskar Assam 40  7 Kumar Chander Coimbatore 45  7 rows selected.  **SQL> savepoint s1;**  **Savepoint created.**  SQL> delete from person;  7 rows deleted.  SQL> select \* from person; no rows selected  **SQL> rollback to savepoint s1;**  **Rollback complete.**  SQL> select \* from person;  PID LASTNAME FIRSTNAME ADDRESS AGE  ------ ---------- ---------- ------------- -------  1 Prettina Anne BAngalore 14  2 Benitto Anish Trichy 24  3 Raj Anita Chennai 27  4 Kumar Ashok Coimbatore 30  5 Hinn Benny Britain 55  6 Prakash Bhaskar Assam 40  7 Kumar Chander Coimbatore 45  7 rows selected. |
| PID LASTNAME FIRSTNAME ADDRESS AGE | |
| ------- ---------- ---------- ------------ -------- | |
| 1 | Prettina Anne BAngalore 14 |
| 2 | Benitto Anish Trichy 24 |
| 3 | Raj Anita Chennai 27 |
| 4 | Kumar Ashok Coimbatore 30 |
| 5 | Hinn Benny Britain 55 |
| 6 | Prakash Bhaskar Assam 40 |
| 7 | Kumar Chander Coimbatore 45 |
| 7 rows selected. | |
| **SQL> commit;** | |
| **Commit complete.** | |
| DELETE COMMAND | |
| SQL> delete from person where lastname='Kumar'; | |
| 2 rows deleted. | |
| SQL> select \* from person; | |
| PID LASTNAME FIRSTNAME ADDRESS AGE | |
| ------ ---------- ---------- ------------ -------- | |
| 1 | Prettina Anne BAngalore 14 |
| 2 | Benitto Anish Trichy 24 |
| 3 | Raj Anita Chennai 27 |
| 5 | Hinn Benny Britain 55 |
| 6 | Prakash Bhaskar Assam 40 |
| **SQL> rollback;** | |
| **Rollback complete.** | |
| SQL> select \* from person; | |
| PID LASTNAME FIRSTNAME ADDRESS AGE | |
| ------- ---------- ---------- ------------ ------- | |
| 1 | Prettina Anne BAngalore 14 |
| 2 | Benitto Anish Trichy 24 |

**RESULT :**

Thus performing insertion, deletion, modifying, altering, updating and viewing records based on conditions was created and the output was verified.