**WEEK 1**

**INSTALLATION OF ORACLE 11G and Practicing DDL commands**

**Objectives:**

Student will able to learn how to install ORACLE 11G and also Create and Manage Tables using DDL Statements.

**Outcomes:**

Student gains the ability to

• Install ORACLE 11G

Categorize the main database objects

• Review the table structure

• List the data types that are available for columns

• Create a simple table

• Describe how constraints are created at the time of table creation

• Describe how schema objects work.

**Installation of ORACLE 11G:**

**Requirements:**

**Description: In this we will install the ORACLE 11G and also practice the “DDL” commands.**

**Installation of ORACLE and Practicing DDL commands:**

**Installation of ORACLE:**

**Why ORACLE:ORACLE is undoubtedly the most popular and widely-used open source database:**

* **It is simple to setup and use.**
* **It is recognised as one of the fastest databse engines.**
* **Most linux( and many Windows-based)web hosts offer ORACLE.**

**Why Install ORACLE Locally?**

**Installing ORACLE on your development PC allows you to safely create and test a web application without affecting the data or systems on your live website.**

**Installing ORACLE on Windows: In this section you will learn how to install SQL on windows system. Windows installer of ORACLE includes auto installer with configuration Wizard that support for easy installation.**

**Click on the following link to install 11G.**

<https://www.filehorse.com/download-oracle-database-express/27798/download/>

**If it asks user name and password, Give System as username and Password as Manager;**

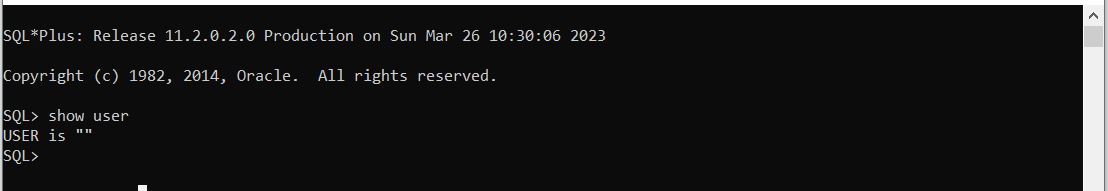
**While installing, read the options carefully and then complete the installation process:**

**Now goto windows button, there you will find the SQL command line, click on it. You will find SQL command prompt like the following:**

****SQL>****

**Before creating a user, we need to know in which user we are. To check that type the following command.**

****SQL>show user;****



To connect to the default user, type the following command.

**SQL > connect system/manager;**

You will see the following message.

Connected.

Type the command as **show user.**

**SQL > show user;**

User is “SYSTEM”

How to create a user:

Type the following commands: In the following command XYZ is user name, where in you can give your own name in place of it, similarly ABC is password, you can give your own password.

**SQL> create user XYZ identified by ABC;**

You will get the message as user created.

**SQL> Grant DBA to XYZ;** ( Granting all the permissions to the created users).

Now we can connect to the created user by using connect username/password.

**SQL>connect XYZ/ABC;**

Now check in which user you are:

**SQL> show user;**

User is XYZ;

To view the list of tables in the user;

**SQL>Select \*from tab;**

No rows selected. ( As there are no tables, we will see this message). Now we will start learning the DDL commands. (Data Definition Language).

**DDL commands:**

1. CREATe
2. ALTER
3. DROP
4. TRUNCATE

**Practicing DDL commands:**

**Creating a table:**

**T**o store data into the database we must create a table with same structure as the data we have to store into the database. To create a table use the create table Command that has the following syntax.

**Syntax:** mysql>Create table <table name> (<colname>< datatype>[constraints],<colname> [constraints]……. colname>< datatype>[constraints],[table level constraints])

**student table:**

Sid int pK

Sname Varchar(30) Notnull

Course Varchar(30) default ‘cpp’

**Query for creating Student Table:**

Mysql>Create table student (Sid int constraint sid\_PK Primary Key, Sname varchar(30) not null. Course varchar(30) constraint default-course default’CPP’)

**Altering Table structure:**

After creating table we may want to add or remove columns and constraints or change the data type of a column to perform all these operations, we have to use alter table that has the following syntax,

**Alter column:** this option is used to change the data type of a column or add or remove not null constraint.

**Syntax:** Alter Table<tablename>modify<colname><datatype>Null/NotNull

**Q**: Sp\_help’student’ //Execute

**EX:** The following example changes the data types of sname column of student table to char with maximum length 40 removes the not Null Column constraint available on it.

**Q**: Alter Table Student modify Sname Char (40) Null

**Q:** sp\_help’student //Execute

**Drop Column**: this option is used to delete columns from the table

Syntax: Alter Table<tableName>Drop Column<Column List>

Ex: The following example deletes the columns Fname and address from the table student.

Q: Alter table Drop Column Fname, Address.

**Altering Table structure:**

After creating table we may want to add or remove columns and constraints or change the data type of a column to perform all these operations, we have to use alter table that has the following syntax,

**Alter column:** this option is used to change the datatype of a column or add or remove not null constraint.

**Syntax:** Alter Table<tablename>AlterColumn<colname><datatype>Null/NotNull

**EX:** The following example changes the data types of sname column of student table to char with maximum length 40 removes the not Null Column constraint available on it.

**Drop Column**: this option is used to delete columns from the table

Syntax: Alter Table<tablEName>Drop Column<Column List>

Ex: The following example deletes the columns Fname and address from the table student.

Q: Alter table <tablename> Drop Column Fname, Address.

**Truncate:** This deletes all the records from a table.

Syntax: Truncate table <table name>

Ex: The following example deletes all records from the table student.

Q: Truncate table student

**VIVA QUESTIONS**

1. Write the syntax for all DDL commands?
2. What is the difference between drop and truncate command?
3. What is DDL interpreter?
4. What is the command to delete columns from table?

**WEEK 2**

**Practicing DML commands**

**SELECT INSERT, UPDATE, DELETE.**

**Data Manipulation Language commands**

1**) Inserting rows into the table:** To insert rows into the table we have to use **insert Command** that has the following syntax

***Syntax:*** insert [into] <tablename>values ([values>)

🡪While using the syntax of insert command we must provide value for every column. Whenever we want to insert null into a column then use the keyword **null** and to insert default value for a column use the keyword **default.**

**Ex:**The following examples insert rows into the table **student**

SQL>Insert into student Values (1001,’A’,’Dotnet’)

SQL>Insert into student Values (1002,’B’,’Defaultt)

SQL>Insert into student Values (1003,’C’,’SQL’)

SQL>Select \* from student //Execute this line

**Ex:**The following examples insert rows into the table **marks**

SQL>Insert into Marks Values (1001, 77, 54, 67, null, null, null)

SQL>Insert into Marks Values (1002, 70, 45, 56, null, null, null)

SQL>Insert into Marks Values (1005, 78, 65, 89, null, null, null)

SQL>Select \* from marks //select this line and execute

**2) Creating a table from another table:** When we have to create a new table from existing table then use the following Syntax of Select statement.

**Syntax:** Select\*|<column list>into<new table name>from <old table name>[where<condition>]

**Ex:** the following example creates a table with name **student3** from the table student.

**Q:** Select \* into student3 from student //Execute

**3) Updating Rows in the table:** When we have to modify the existing data in the table then we have to use **update Command** that has the following syntax

***Syntax: Update*** <tablename> Set<column>=<value>,<colname>=<value>…[Where<condition>]

**Ex:**The following examples updates name and course of the student with id 1004 to **S** and **DOTNET**

**Q**: Update student set name-‘S’, course=”DOTNET” where sid=1004;

Q: Select \* from student;

**EX:** The following example updates marks table by calculating total and average for all rows.

**Q**: update marks set total=C+CPP+SQL,aveg=(C+CPP+SQL)/3.0

**Q:** Select \* from marks

**Ex: T**the following example updates marks table by calculating the grade on average marks.

Q: Update marks set grade=’C’

When Aveg>=70 then ‘distinction’

When **Aveg**>=60 and Aveg<70 then ‘first class’

When Aveg>=50 and Aveg<60 then’Second class’

When Aveg>=35 And Aveg<50 then’third class’

Else’Fail’

End

**4) Deleting Rows from the table:**To delete rows from table use the delete Command that has the following syntax:

Syntax: Delete Rows from Table

>delete[from]<tablename>[where<condition>];

**Ex:** The following example deletes the student record with si4-1004 from marks table

**Q:** Delete from marks where sid=1004;

**Q:** Select \* from marks;

**EX:** The following example deletes all rows from the marks.

**Q:** delete from marks;

**Output**

|  |  |
| --- | --- |
| **RATING** | **MINAGE** |
| 8 | 25.5 |
| 7 | 35 |
| 3 | 25.5 |

4. Create the table using following attributes **BUS** **(**BUSNO: VARCHAR2 (10): **PK**, SOURCE: VARCHAR2 (50), DESTINATION: VARCHAR2 (50)**)**

**Creating table**

**Syntax**

SQL>Create table <table name> (col1 datatype,col2 datatype,col3 datatype);

**Query**

SQL>Create table BUS (BUSNO VARCHAR2(10) Primary Key, SOURCE VARCHAR2 (50), DESTINATION VARCHAR2 (50));

**Describing table**

**Query**

Mysql>desc bus;

**Output**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Column** | **Data Type** | **Length** | **Precision** | **Scale** | **Primary Key** | **Nullable** | **Default** | **Comment** |
| [BUS](javascript:ret_Column('TIRU.BUS');) | [BUSNO](javascript:ret_Column('BUSNO');) | Varchar2 | 10 | - | - | 1 | - | - | - |
|  | [SOURCE](javascript:ret_Column('SOURCE');) | Varchar2 | 50 | - | - | - | nullable | - | - |
|  | [DESTINATION](javascript:ret_Column('DESTINATION');) | Varchar2 | 50 | - | - | - | nullable | - | - |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | | | | | | | | | | |

**Inserting records into “Bus” table**

**Syntax**

SQL>Insert into <table name> values (val 1,val2,val3)

**Query**

SQL>insert into bus values(1234,'hyderabad',’tirupathi’);

SQL>insert into bus values(2345,'hyderabad',’tirupathi’);

SQL>insert into bus values(23,'hyderabad',’kolkata’);

SQL>insert into bus values(45,' tirupathi ',’banglore’);

SQL>insert into bus values(34,'hyderabad',’chennai’);

**Display table**

**Syntax**

SQL>Select <select list> from <table name>

**Query**

SQL>Select \* from bus;

**Output**

|  |  |  |
| --- | --- | --- |
| **BUSNO** | **SOURCE** | **DESTINATION** |
| 1234 | hyderabad | tirupathi |
| 2345 | hyderabad | tirupathi |
| 23 | hyderabad | kolkata |
| 45 | tirupathi | banglore |
| 34 | hyderabad | chennai |
|  |  |  |

5. Create the table using following attributesPASSENGER (PPNO: VARCHAR2 (15): PK, NAME: VARCHAR2 (15), AGE: INT (4), SEX: CHAR (10): MALE/FEMALE, ADDRESS: VARCHAR2 (50))

**Creating table**

**Syntax**

SQL>Create table <table name> (col1 datatype,col2 datatype,col3 datatype);

**Query**

SQL>CREATE TABLE PASSENGER (PPNO VARCHAR2 (15) PRIMARY KEY, NAME VARCHAR2 (15), AGE NUMBER (4), SEX CHAR (10), ADDRESS VARCHAR2 (50));

**Describing table**

**Query**

SQL>Desc PASSENGER;

**Output**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Column** | **Data Type** | **Length** | **Precision** | **Scale** | **Primary Key** | **Nullable** | **Default** | | **Comment** |
| [PASSENGER](javascript:ret_Column('TIRU.PASSENGER');) | [PPNO](javascript:ret_Column('PPNO');) | Varchar2 | 15 | - | - | 1 | - | - | | - |
|  | [NAME](javascript:ret_Column('NAME');) | Varchar2 | 15 | - | - | - | nullable | - | | - |
|  | [AGE](javascript:ret_Column('AGE');) | Number | - | 4 | 0 | - | nullable | - | | - |
|  | [SEX](javascript:ret_Column('SEX');) | Char | 10 | - | - | - | nullable | - | | - |
|  | [ADDRESS](javascript:ret_Column('ADDRESS');) | Varchar2 | 50 | - | - | - | nullable | - | | - |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | 1 - 5 |  |  | | | | | | | | | |

**Inserting records into “PASSENGER” table**

**Syntax**

SQL>Insert into <table name> values (val 1,val2,val3);

**Query**

SQL>insert into PASSENGER values(1,'TIRUMALAY',19,'MALE','AMBERPET');

SQL>insert into PASSENGER values(2,'SUPRIYA',20,'FEMALE','B.B NAGAR');

SQL>insert into PASSENGER values(3,'AMULYA',20,'FEMALE','ECIL');

SQL>insert into PASSENGER values(4,'NAGARAJU',20,'MALE','NAGARAM');

SQL>insert into PASSENGER values(5,'AVS.RAVI',20,'MALE','B.B NAGAR');

**Display table**

**Syntax**

SQL>Select <select list> from <table name>;

**Query**

SQL>Select \* from PASSENGER;

**Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PPNO** | **NAME** | **AGE** | **SEX** | **ADDRESS** |
| 1 | TIRUMALAY | 19 | MALE | AMBERPET |
| 2 | SUPRIYA | 20 | FEMALE | B.B NAGAR |
| 3 | AMULYA | 20 | FEMALE | ECIL |
| 4 | NAGARAJU | 20 | MALE | NAGARAM |
| 5 | AVS.RAVI | 20 | MALE | B.B NAGAR |

**VIVA QUESTIONS**

1. What is the syntax for insert command?
2. Define Key constraint?
3. What is the difference between NULL Values and NOT NULL Values?
4. What is the command to display table?