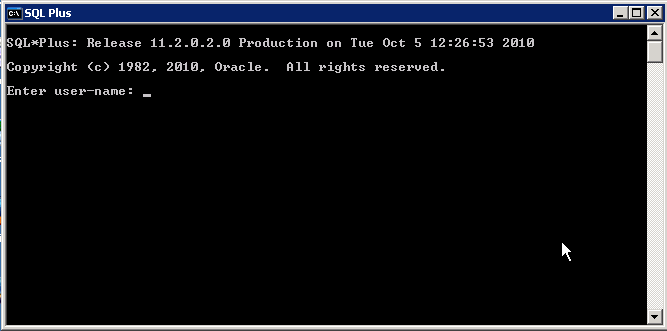
**SQL\*Plus** and **SQL Developer** and other client tools are installed with Oracle Database.

SQL\*Plus is a command line tool for transmitting SQL and PL/SQL to the database. SQL Developer is a graphical development environment for SQL and PL/SQL.

If you have installed Oracle Database locally on your PC, you can follow these steps to start SQL\*Plus or SQL Developer:

1. From the Windows Start menu, select **Oracle – OraDb11g\_home*n***.
2. Choose **Application Development**.
3. Choose the name of the application.
4. When SQL\*Plus starts up it is already connected to the local database. Enter the name of a user ID .You will be prompted to enter the password.
5. 

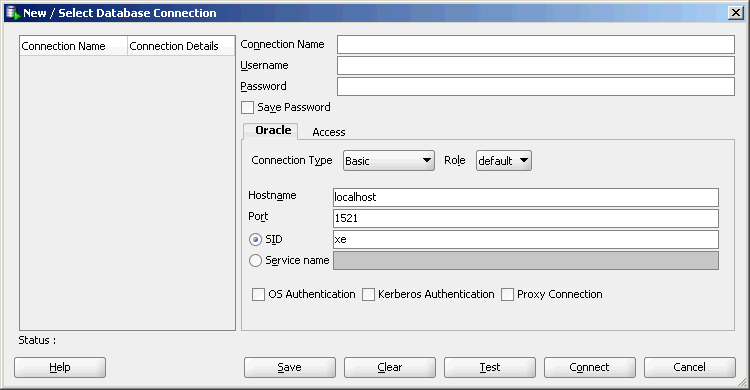
## Connect SQL Developer

The first time you run SQL Developer, you may be prompted to provide the path to the Java executable. If so, specify the full path of jdk\bin\java.exe under the Oracle home directory for the database. For the installation described in [Chapter 2](https://docs.oracle.com/cd/E11882_01/datamine.112/e16807/install_odm.htm#BEHFJDBC), the path would be:

C:\app\myuser\product\11.2.0\dbhome\_1\jdk\bin\java.exe

To define a connection to the local database:

1. Right click **Connections** and choose **New**.
2. The New/Select Database Connection dialog appears.

  
For **Connection Name**, specify a meaningful name that you will remember.

* + For **Username/Password**, provide a user name and password for the connection. The user must have data mining privileges as described in [Chapter 4](https://docs.oracle.com/cd/E11882_01/datamine.112/e16807/security.htm#BGBJIIGB).
  + For **Hostname**, specify localhost to indicate that the database is hosted on the local computer.
  + For **Port**, specify the port where the database will listen to requests from the client. The default port is 1521.
  + For SID (Service Identifier), specify the simple database name.
  + Click Connect to connect to the database using the specified credentials.

Tables used in our lab sessions:

**• EMP table:** which gives details of all the employees.

• **DEPT table:** which gives details of all the departments.

• **SALGRADE table:** which gives details of salaries for various grades.

• **BONUS table:** which gives details of employee’s salaries

**Writing SQL Statements:**

• Within SQL\*Plus, a SQL statement is entered at the SQL prompt, and the subsequent lines are numbered. This is called the SQL buffer. Only one statement can be current at any time within the buffer.

• SQL statements are not case sensitive, unless indicated.

• SQL statements can be entered on one or many lines.

**Displaying Table Structure**:

Use the SQL\*Plus DESCRIBE command to display the structure of a table. Example:

SQL> DESCRIBE dept; or

SQL> DESC dept;

In the result: NOT NULL indicates that a column must contain data

To see all the tables in the scott/tiger workspace, write the following command:

SQL> select \* from cat; or

SQL> select \* from tab;

**Basic Select Commands**

SELECT \* FROM dept;

SELECT deptname, loc FROM dept;

**Use Arithmetic Operators:**

SELECT ename, sal, sal+300 FROM emp;

Defining a Null Value:

SELECT ename, job, sal, comm FROM emp;

Note: some records do not have values in comm field.

**Define a Column Alias:**

SELECT ename AS name, sal AS salary 2 FROM emp;

SELECT ename "Name", sal\*12 "Annual Salary" FROM emp;

**Eliminate Duplicate Rows:**

SELECT DISTINCT deptno FROM emp;

**Limit Rows Selected:**

**Comparison Operators**:

SELECT ename, sal, comm 2 FROM emp 3 WHERE sal<=comm;

Use Comparison Operators:

SELECT ename, sal FROM emp WHERE sal BETWEEN 1000 AND 1500;

SELECT mpno, ename, sal, mgr FROM emp WHERE mgr IN (7902, 7566, 7788);

SELECT ename FROM emp WHERE ename LIKE 'S%';

**Use Logical Operators:**

SELECT empno, ename, job, sal FROM emp WHERE sal>=1100 AND job='CLERK';

SELECT empno, ename, job, sal FROM emp WHERE sal>=1100 OR job='CLERK';

SELECT ename, job FROM emp WHERE job NOT IN ('CLERK','MANAGER','ANALYST');

select ename,job,sal from emp where job not like 'CLERK';

Use ORDER BY Clause:

SELECT ename, job, deptno, hiredate FROM emp ORDER BY hiredate;

SELECT ename, job, deptno, hiredate FROM emp ORDER BY hiredate DESC;

**Write query to:**

1)Show the structure of the EMP table.

1. Create a query to display the name, job, hire date, and employee number for each employee.
2. Save your SQL statement to a file named p1q1.sql.

4) Create a query to display the name and salary of employees earning more than $ 2850

4) Create a query to display unique jobs from the Emp table;

5) display the name and salary for all employees whose salary is not in the range of $1500 and $2850.

7) Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary.

8) Display the names of all employees where the third letter of their name is an A.