##### [Establishing Connection](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T12)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Hands-on Assignment** | **Topics Covered** | **Status** |
| 1 | Write a java program that establishes a connection to oracle database successfully. If the connection is successful, it should display a message “Connection Established successfully”. In case, it is not able to do so due to any exception, it should display the message “Connection could not be established “. If there is an exception, it should also display the description of the exception. | JDBC, Driver Manager, Connection |  |
| 2 | In the just concluded exercise, where you have established the connection successfully, exclude the registration process(by commenting the line containing the code Class.forName(“..”)). Observe the result. | JDBC, Driver Manager, Connection |  |

##### [Executing Query & Processing Results](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T13)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Hands-on Assignment** | **Topics Covered** | **Status** |
| 1 | Write a java program that connects to oracle database, queries the inbuilt table “emp” and displays the first two columns (empno using column index and ename using column name ) of all the rows. | JDBC, Driver Manager, Connection,Statements |  |
| 2 | Modify the above program to display all the rows where sal is greater than 1000 and less than 2000. Display the columns ename, job, sal and comm. | JDBC, Driver Manager, Connection,Statements |  |

##### [Using PreparedStatement & MetaData objects](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T14)

|  |  |
| --- | --- |
|  | |
| **No.** | | **Hands-on Assignment** | **Topics Covered** | | **Status** | |
| 1 | | Develop a jdbc program containing main method, which should instantiate a class called DAOClass, which should contain methods called insert, delete, modify and display. Description of what each of these methods are expected to do is given below. Necessary details required for executing these methods, are passed from command line argument. For e.g. If the name of the class containing the main method is JDBCCalls, then if you want to insert a record, you will execute this class as java JDBCCalls 1 101 “Ajit” “IV” “20-Nov-2001” 4000  Where 1 is the option for inserting the record and all other details are the values for the columns in each row of the student table. The structure of student table is given below. Similarly, for deleting a record, you have to execute the code as  java JDBCCalls 2 101  where 2 is the option for deleting a record and 101 is the rollno of the student, whose record has to be deleted.  For modifying a record, you will use  java JDBCCalls 3 101 4500, where 3 is the option for modifying a record and the 4500 is the new fee which needs to replace the old fee value.  For Displaying records, if the main class is executed as follows  java JDBCCalls 4 101  it should display only one record, that of the student with roll no. 101. 4 option is for displaying the record.  If the main class is executed as  java JDBCCalls 4 (without specifying the rollno.), it means that details of all the students should be displayed. | PreparedStatement | |  | |
| 2 | | Inserting a record  ABC International School wants to computerize students details. The school maintains a database of students in Oracle. The student table contains information related to students and is shown in the following student table structure.  Column Name Type Constraint  Rollno Number (4) Primary Key  StudentName Varchar (20) Not Null  Standard Varchar (2) Not Null  Date\_Of\_Birth Date  Fees Number (9,2)    When a new student joins the school, the student record is inserted in the student table. The valid student details are as follows:  • Rollno: Valid value is a 4-digit number  • StudentName: Valid value can contain maximum 20 letters in uppercase  • Standard : Valid values are Roman Letters representing I to X(I, II, III, IV….IX, X)  Write a Java program to insert some records to the table | PreparedStatement | |  | |
| 3 | | Deleting a Student’s record  When a student leaves the school, the record related to that student needs to be deleted from the Student table. The student’s roll no, whose record has to be deleted, should be passed as a command line argument.  Upon deletion, the Student details must be stored in another table named StudentLog which will maintain the details such as Rollno, StudentName, Standard and Leaving\_date. | PreparedStatement | |  | |
| 4 | | Modification of Student record  When there is a change in the fee to be paid by a student, the respective row should be appropriately updated. Pass the rollno from the command prompt along with the new fee amount and this amount should be reflected in the table for that particular student. | PreparedStatement | |  | |
| 5 | | Display Student details  Write the code to display details of all the students, if no roll no. is passed, while executing the main program.  If while executing the main program, the roll no. is passed, then it should display the record of that particular student. | PreparedStatement | |  | |
|  | |  |  | |  | |
| No. | | DDL   [DML](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T8) Hands-on Assignment | | | Topics Covered | | Status | |
| 1 | | Run the below script  Create table MY\_EMPLOYEE  as  Select employee\_id,first\_name,last\_name,department\_id,salary from EMPLOYEES where 1=2; | | | DML | |  | |
| 2 | | Test the table creation by viewing the structure using describe command  Name Null Type  ------------------------------ -------- ------------------------------  EMPLOYEE\_ID NUMBER(6)  FIRST\_NAME VARCHAR2(20)  LAST\_NAME NOT NULL VARCHAR2(25)  DEPARTMENT\_ID NUMBER(4)  SALARY NUMBER(8,2)  5 rows selected | | | DML | |  | |
| 3 | | Insert one record without listing the column names in the insert statement. Check whether data is inserted  Eg:  employee\_id first\_name last\_name department\_id salary  201 Michael Hartstein 20 13000 | | | DML | |  | |
| 4 | | Insert one record without listing the column names in the insert statement where salary value remain undetermined. Check whether data is inserted  Eg:  employee\_id first\_name last\_name department\_id salary  201 Michael Hartstein 20 13000  202 Pat Fay 20 (null) | | | DML | |  | |
| 5 | | Insert one record with listing the column names avoiding salary column in the insert statement where salary value remain undetermined. Check whether data is inserted  employee\_id first\_name last\_name department\_id salary  201 Michael Hartstein 20 13000  202 Pat Fay 20 (null)  203 Susan Mavris 40 (null) | | | DML | |  | |
| 6 | | Use the above Script to insert the below given records  employee\_id first\_name last\_name department\_id salary  205 Shelley Higgins 110 12000  100 Steven King 90 24000  101 Neena Kochhar 90 17000  102 Lex De Haan 90 17000  111 Ismael Sciarra 100 7700  112 Jose Manuel Urman 100 7800  204 Hermann Baer 70 10000 | | | DML | |  | |
| 7 | | Create a query to update salary by 10% for all employees in dept 90 | | | DML | |  | |
| 8 | | Create a query to update Last\_name of emp 202 to Higgins | | | DML | |  | |
| 9 | | Delete employees whose name either first or last name has char seq of ‘man’ | | | DML | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | [Oracle 11g Introduction](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T2) | **Topics Covered** | **Status** |
| 1 | Using SQL Developer: Create a database connection using the following information:  Connection Name: myconnection  Username: hr  Password: hr  Hostname: localhost  Port: 1521  SID: ORCL  Ensure that you select the Save Password check box.  Testing and Connecting Using the Oracle SQL Developer Database Connection  If the status is Success, connect to the database using this new connection. | Starting up SQL Plus and SQL Developer |  |
| 2 | Expand MyConnection -- > Explore  All the available table  structure of Employee table - its columns  view the data tab of the Employee tables | Starting up SQL Plus and SQL Developer |  |
| 3 | Start SQLPLUS  using UserName : hr  password : hr | Starting up SQL Plus and SQL Developer |  |

##### [Select Statement](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T3)

Determine the structure of the DEPARTMENTS table and its contents.

Select Statement2

Create a query to display the last name, job ID, hire date, and employee ID for each employee, with the employee ID appearing first. Provide an alias STARTDATE for the HIRE\_DATE column.

Select Statement3

Create a query to display all unique job IDs from the EMPLOYEES table.

Select Statement4

Create a query to display employee id, last name, job id and hiredate from employee table with more describing column names. Name the column headings

Emp # , Employee , Job , and Hire Date , respectively.

Select Statement5

Create a report of all employees and their job IDs. Display the last name concatenated with the job ID (separated by a comma and space) and name the column as "Employee and Title"

Select Statement

##### [Restricting and Sorting Data](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T4)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Hands-on Assignment** | | **Topics Covered** | | | **Status** | |
| 1 | Create a report that displays the last name and salary of employees who earn more than $12,000. | | Restricting and Sorting Data | | |  | |
| 2 | Create a report that displays the last name and department number for employee number 176. | | Restricting and Sorting Data | | |  | |
| 3 | To find high-salary and low-salary employees. Create a query to display the last name and salary for any employee whose salary is not in the range of $5,000 to $12,000 | | Restricting and Sorting Data | | |  | |
| 4 | Create a report to display the last name, job ID, and hire date for employees with the last names of Matos and Taylor. Order the query in ascending order by the hire date. | | Restricting and Sorting Data | | |  | |
| 5 | Display the last name and department ID of all employees in departments 20 or 50 in ascending alphabetical order by name. | | Restricting and Sorting Data | | |  | |
| 6 | List employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns as  Employee and Monthly Salary, respectively. | | Restricting and Sorting Data | | |  | |
| 7 | Create a report that displays the last name and hire date for all  employees who were hired in 1994 | | Restricting and Sorting Data | | |  | |
| 8 | Create a report to display the last name and job title of all employees who do not have a manager | | Restricting and Sorting Data | | |  | |
| 9 | Create a report to display the last name, salary, and commission of all employees who earn commissions. Sort data in descending order of salary and commissions. Use the column’s numeric position in the ORDER BY clause. | | Restricting and Sorting Data | | |  | |
| 10 | Create a report that displays the last name and salary of employees who earn more than an amount that the user specifies after a prompt. If you enter 12000, it should display all employees earning more than 12000.  Eg:  Salary\_value: 12000 | | Restricting and Sorting Data | | |  | |
| 11 | Create a query that prompts the user for a manager ID and generates the employee ID, last name, salary and department for that manager’s employees and prompts a column name by which result should be sorted.  Eg:  manager\_id :103  sorted\_by : last\_name | | Restricting and Sorting Data | | |  | |
| 12 | Display all employee last names in which the third letter of the name is “a.” | | Restricting and Sorting Data | | |  | |
| 13 | Display the last names of all employees who have both an “a” and an “e” in their last name. | | Restricting and Sorting Data | | |  | |
| 14 | Display the last name, job, and salary for all employees whose jobs are either those of a sales representative or of a stock clerk, and whose salaries are not equal to $2,500, $3,500, or $7,000. | | Restricting and Sorting Data | | |  | |
| No. | | [Group Functions](http://59.162.104.101:25000/PBLApp/tmodule.action?module=RDBMS%20/%20SQL%20/%20JDBC&stream=Java&tm=TM8" \l "T6) Hands-on Assignment | | Topics Covered | Status | |
| 1 | | Find the highest, lowest, sum, and average salary of all employees. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Run the query. | | Group Functions |  | |
| 2 | | Modify the above query to display the minimum, maximum, sum, and average salary for each job type. | | Group Functions |  | |
| 3 | | Determine the number of managers without listing them. Label the column Number of Managers | | Group Functions |  | |
| 4 | | Find the difference between the highest and lowest salaries. Label the column DIFFERENCE | | Group Functions |  | |
| 5 | | Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary. | | Group Functions |  | |