1. For Ms-acess

* Create database
  + - Go to MS-office - Access- select blank database - save to java program location -ok
    - Create table login
    - Add some records to te table

* Create DSN
  + - Go to 🡪 C:\Windows\SysWOW64\ odbcad32.exe then configure
    - Create DSN
    - Browse created database and mapped with DSN
    - Required 32 bit jdk (i.e. jdk 1.7 32 bit) because we r using 32 bit odbc driver
* Create without DSN
  + - String database="student.mdb";//Here database exists in the current directory
    - String url="jdbc:odbc:Driver={Microsoft Access Driver (\*.mdb)};DBQ=" + database + ";DriverID=22;READONLY=true";

1. For mysql

* Requierd mysqlconnnector

C:\Program Files (x86)\Java\jre7\lib\ext\mysql-connector-java-5.1.40-bin.jar

* open mysql prompt
  + - mysql> Give password : root
    - mysql> ;
    - mysql> use student
    - mysql> create table login( id integer(10),name varchar(20));
    - mysql> desc login
    - mysql> insert into login values (1,'reshma');
    - mysql> insert into login values (2,'rama');
    - mysql> select \* from login;

1. For oracle
   * + Copy ojdbc6.jar file to c:\oracle\ folder…\lib\ojdbc6.jar
       - C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar

OR

* + - Open cmd and set class path
      * set classpath=F:\reshma\GPP\Even\_2021\JP-2\chap 4\oracle\ojdbc14.jar;.;
    - open command prompt
    - on cmd write sqlplus
    - SQL>create table login( id integer(10),name varchar(20),age integer(10));
    - SQL>desc login
    - SQL>insert into login (1,'reshma',10);
    - SQL>insert into login (2,'rama',20);
    - SQL>select \* from login;

For callable statements

1. Make sure stored procedure available in the database

create or replace function sum4

(n1 in number,n2 in number)

**return** number

is

temp number(8);

begin

temp :=n1+n2;

**return** temp;

end;

/

1. Create a callable Statements with the procedure call.

* Syntax: CallableStatement   cst = con.prepareCall ("{ call procedurename(?,?...?)}");
* Example:  CallableStatement  cst = con .prepareCall ("{ call sum4 (?,?...?)}");

1. Provide values for every IN Parameter by using corresponding setter method

* cst.setInt (1, 100);
* cst.setInt (2, 200);

↓ ↓

Index value

1. If stored procedure has OUT parameter then to hold that output we should register every OUT parameter by using the following methods.

* Syntax : Public void registerOutParameter (int index, int jdbctype)
* Example : cst. registerOutParameter(3, Types.INTEGER);

1. EXECUTE Procedure call

* cst.execute();

1. Get the result from OUT parameter by using that corresponding getXXX() method.

* Example: int result =cst.getInt(3);

1. Close connection.

* Create procedure
* create table user420(id number(10),name varchar2(20));
* Write Stored procedure

create or replace procedure "INSERTR"

(id IN NUMBER,

name IN VARCHAR2)

is

begin

insert into user420 values(id,name);

end;

/

* select \*from user420;

