SQL Group project

Write a text-based, Java program. Everything will be done in Java: creating and dropping tables, inserting, deleting and updating tables. You will be working with your group to come up with your own database that contains 2 tables, a parent table and a child table, such as Party and Candidate (Should be a one to many relationship). Prefix your table names with a funky group name to avoid collisions. DO NOT USE Party and Candidate or Person and Personality as your tables. Come up with your own. At the end of your program, be sure to drop your tables, which means that every time you run your program, you will be creating the tables. Modularize your code. Your methods should not contain more than 20 lines.

Display a menu that allows the user to insert, delete, update or view all the records.

1) Insert
2) Delete
3) Update
4) View
5) Quit

1) Insert--- upon choosing insert, ask the user a series of questions that map the response to the different columns in your tables. Let's take the Candidate and the Party tables as an example. With candidate name, john and political party, republican, you would have to create an entry in the party table if a record does not exist for the republican party and a corresponding record in the candidate table for John. You would have to make sure that the foreign keys and primary keys are assigned properly. You can use sequences to generate your primary keys. Optional: You are more than welcome to do any additional validation although it is not a requirement.

2) Delete – You would ask for some kind of unique information that identifies a record in the child table and then delete the record. If there are no other instances of the foreign key, then also delete the record from the parent table. For example, if we look up john and he is the only republican, then we would delete both john from the candidate table and republican from the party table. Optional: You are more than welcome to do any additional validation although it is not a requirement.

3) Update – You would ask for some kind of unique information that identifies a record in the parent table (must be unique) and then ask what the information should be changed to such as

What party do you want to change? Republican
What do you want to change it to? Conservative

4) View – Display all the records in the child table and their corresponding information from the parent table. Make sure to not include the primary and foreign keys. Here is a sample

John	Republican
Jim	Democrat
Jill	Independent

6) Quit – End the program

What to turn in:

.java file
A single word document that contains the following:

1) An ER diagram of your 2 tables
2) screenshots of a sample run which tests
   - 2 inserts
   - 2 deletes
   - 1 update
   - 1 view

Here is some sample Java code:

```
/*

Download jar file from oracle site.oracle.com/downloads
https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html or use the JDBC file that I
have provided. If you wish to download it directly from the site, you have to make sure that it
matches the version number of Oracle that is being used. To find the version number, use the
following SQL statement.

select * from v$version;

--Once you download the jar file, set the classpath to the jarfile. Include the file name in your
classpath. For example if you are doing this in DOS, you can set the classpath using the following
syntax:
set classpath=


--To set the classpath in JGrasp, goto  settings/path classpath/add the new path. Be sure to include
the name of the file in your classpath
*/


//It can take up to 15 seconds to run the program
import java.sql.*;
class Driver{
   public static void main(String[] args)throws Exception{
      DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
      Connection con=DriverManager.getConnection(
       "jdbc:oracle:thin:@sabzevi2.homeip.net:1521:orcl", "csus", "student");

      Statement st=con.createStatement();
      try {
         st.executeQuery("drop table test");
      } catch (SQLException s){  }

      st.executeQuery("create table test (col1 number, col2 number)");
      st.executeQuery("insert into test values (15,16)");
      ResultSet rs=st.executeQuery("select * from test ");
      while (rs.next())
         System.out.print(rs.getString(1)+" " +rs.getString(2));
   }
}
```