CS 4611 – Fall 2017 – Laboratory 8
Assigned: 10/24/2017
Due: 11/2/2017 at 11:59 PM. Submit your file(s) to canvas.
Maximum Grade: 100 pts.

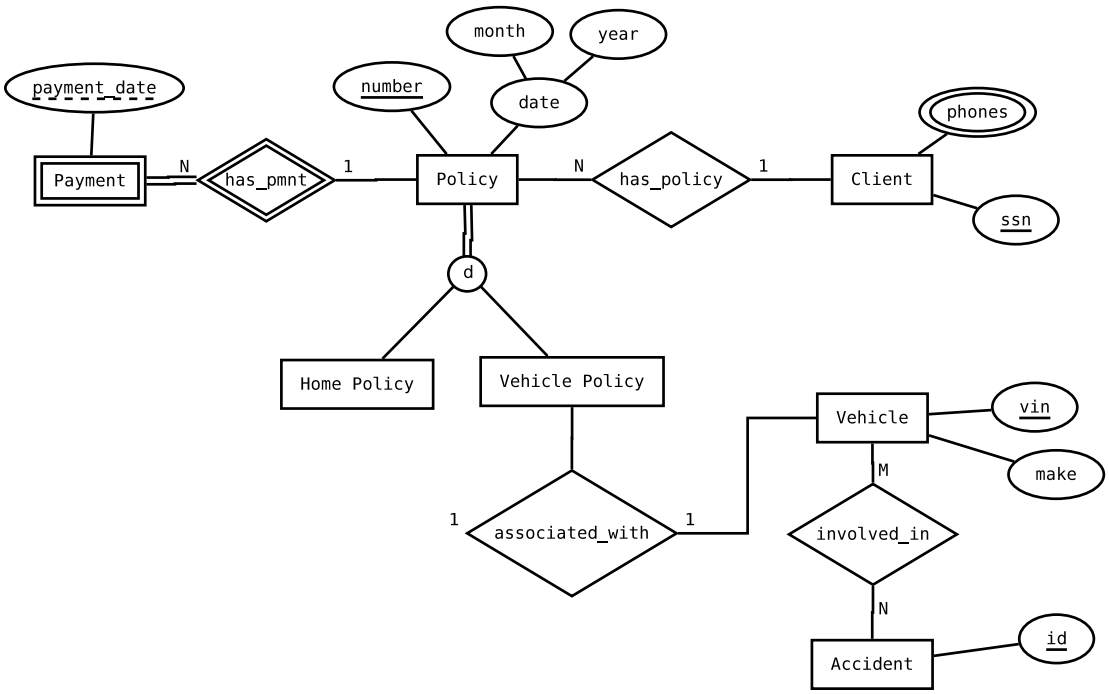
Objectives: The objectives of this lab are the following:

- **Learn how to use Java's JDBC library to connect to Oracle.**

Note:

- **This homework is an assignment to be done individually. No collaboration with anyone else is allowed.**

Consider the following E-R diagram:



with the following constraints:

1. The payment day must follow the date of the policy.
2. The valid makes of cars are 'Toyota', 'Honda', 'Acura', and 'Volkswagen.'
3. Every client has an insurance policy.

Activity 1:

Write a file named `your_last_name_lab8.sql` *containing your name at the top (commented out)*, and also containing the following:

a) An implementation in Oracle SQL of the tables that result from reducing the E-R diagram above. These tables must have all the appropriate primary keys, foreign keys, not null constraints, check constraints, etc.
b) An implementation for the constraints above mentioned.

Activity 2:

Consider the following queries that refer to the E-R diagram in Activity 1:

1. Insert a given vehicle.
2. Associate a given vehicle with an accident.
3. Retrieve the names of the top-K (K is a positive integer parameter) clients that have been involved in the most accidents.
4. Delete all policies with dates prior to a given date.

Write a file named `your_last_name_lab8.java` *containing your name (commented out) at the top.* This file is going to contain a Java method for each query. Such methods must read their inputs from the standard input, and then access Oracle through JDBC. For example, the first query that asks to insert a given vehicle, your Java method must request the appropriate parameters from the standard input ("vin" and "make"), and then perform the insertion into Oracle.

Each method you write in Java must also satisfy the following requirements:

- If the query is an INSERT, then the Java method must display all the rows of the table where you just inserted a new row. These rows must be displayed in a readable form, where the fields of a row are separated by tabs.
  For example, if you insert a row
  ('Jake Maas', 37, 43000)
  into table police_officer, then the result of calling the corresponding method in java must show:
  Insertion of row ('Jake Maas', 37, 43000) successful.
  The rows contained in table 'police officer' are now:

Officer_name	Age	Salary
Diana Liprandi	67	56000
Jake Maas	37	43000

- If the query is an SELECT, then the Java method must display all the rows of the query result set. These rows must be displayed in a readable form, where the fields of a row are separated by tabs.
  For example, if you select all officers whose age is less than 25:

Officer_name	Age	Salary
Victoria Carlucci	24	26000
Dustin Alvarez	22	23000

- If the query is an DELETE, then the Java method must display the deleted row, and all the rows remaining in the table where you just deleted row. These rows must be displayed in a readable form, where the fields of a row are separated by tabs.
  For example, if you insert a row
  ('Jake Maas', 37, 43000)

into table police_officer, then the result of calling the corresponding method in java must show:

Deletion of row ('Jake Maas', 37, 43000) successful.
The rows contained in table 'police officer' are now:

Officer_name	Age	Salary
Diana Liprandi	67	56000

Additional Notes:

- You must use Javadoc to properly document your Java methods and your Java class. Explain what each argument to your method is supposed to do.
- Your file must run in Java 1.8 using JDBC to connect to Oracle 11g.