

## Question 7)

```
static Object toEntityClass(ResultSet resultSet, BeanInfo beanInfo, Constructor<?>
constructor) throws SQLException {

    var instance = Utils.newInstance(constructor);
    for(var property : beanInfo.getPropertyDescriptors()) {

        //We call the setter with the correct value from the resultSet
        var setterValue = resultSet.getObject(property.getName());
        var setter = property.getWriteMethod();
        Utils.invokeMethod(instance, setter, setterValue);

    }

    return instance;
}

static List<Object> findAll(Connection connection, String sqlQuery, BeanInfo
beanInfo, Constructor<?> constructor) throws SQLException{

    var list = new ArrayList<>();
    try(var statement = connection.prepareStatement(sqlQuery)) {

        try(var resultSet = statement.executeQuery()) {
            while(resultSet.next()) {
                var instance = toEntityClass(resultSet, beanInfo, constructor);
                list.add(instance);
            }
            return list;
        }
    }
}
```

## Question 8)

```
public static String createSaveQuery(String tableName, BeanInfo beanInfo) {
    var insertInto = "INSERT INTO "+tableName+" ";
    var columns = Arrays.stream(beanInfo.getPropertyDescriptors())
        .filter(property -> !property.getName().equals("class"))
        .map(ORM::findColumnName)
        .collect(Collectors.joining(", ", "(", ")"));

    //we need to add a (?) in the values for each columns
}
```

```
        var values = Arrays.stream(beanInfo.getPropertyDescriptors())
            .filter(property -> !property.getName().equals("class"))
            .map(property -> "?")
            .collect(Collectors.joining(",","(",")"));
        return insertInto + columns + " VALUES " + values;

    }

    public static Object save(Connection connection,String tableName,BeanInfo
    beanInfo,Object bean,PropertyDescriptor idProperty) throws SQLException {

        var query = createSaveQuery(tableName,beanInfo);
        try(var statement = connection.prepareStatement(query)) {

            for(var property : beanInfo.getPropertyDescriptors()) {
                var columnIndex = 1; //we start at 1 for the column index
                if(property.getName().equals("class")) {
                    continue;
                }
                var value = Utils.invokeMethod(bean,getGetterMethod(property));
                statement.setObject(columnIndex++,value);
            }
        }

        return bean;

    }

}
```