NAME: ______________________________

Instructions
1. Do all of the problems
2. You have 50 minutes for the exam
3. Show all your work
4. Do not separate exam papers

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Easy</th>
<th>Difficulty Level</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **E-R Diagram:** (25 pts) Draw the E-R diagram for a real-estate firm that lists property for sale. The following describes this organization.

   - The firm has a number of sales offices in several states. Attributes of sales office include Office_Number (identifier) and location.
   - Each sales office is assigned one or more employees. Attributes of employee include Employee_ID (identifier) and Name. An employee must be assigned to only one sales office.
   - For each sales office, there is always one employee assigned to manage that office. An employee may manage only the sales office to which he or she is assigned.
   - The firm lists property for sale. Each property has a Property_ID (identifier) and location. Components of location include address, city state and zipcode.
   - Each unit of property must be listed with one (and only one) of the sales offices. A sales office may have any number of properties listed, or may have no property listed.
   - Each unit of property has one or more owners. Each owner has an owner_id (identifier) and name. An owner may own one or more units of property. If there are multiple owners, each owner owns a percentage of the property.
Figure 1: ER Diagram
2. **Relational Algebra:** (25 pts) Consider the following database schema that keeps track of student enrollment in courses and the books adopted for each course.

\[
\begin{align*}
\text{Student} & \left(*ssn, \text{name}, \text{major}, \text{bdate}\right) \\
\text{Course} & \left(*\text{course}\#, \text{cname}, \text{dept}\right) \\
\text{Enroll} & \left(\text{ssn}, *\text{course}\#, *\text{semester}, \text{grade}\right) \\
\text{BookAdoption} & \left(*\text{course}\#, *\text{semester}, \text{isbn}\right) \\
\text{Textbook} & \left(*\text{testisbn}, \text{title}, \text{publisher}, \text{author}\right)
\end{align*}
\]

Sample tuple for each relation is given below

\[
\begin{align*}
\text{Student}(123456, \text{John Smith, Computer Science, 11/15/2000}) \\
\text{Course}(3743, \text{Database, Computer Science}) \\
\text{Enroll}(123456, 3743, \text{Spring 2018, A}) \\
\text{BookAdoption}(3743, \text{Spring 2018, 11234595}) \\
\text{Textbook}(11234595, \text{Introduction to Databases, AWL Publishing, Steve Knight})
\end{align*}
\]

Keys are denoted by *.

Specify the following queries in Relational Algebra using above database schema.

(a) List the textbooks published by 'AWL Publishing'.
\[
\Pi_{\text{title}} \left(\sigma_{\text{publisher}='AWLPublishing'} \text{Textbook} \right)
\]

(b) List the names of the courses taken by 'Susan Smith' in fall 2005.
\[
\begin{align*}
S_{SMITH} & \leftarrow \sigma_{\text{name}='Susan Smith' \ \text{AND \ Semester=FALL 2005}} \left(\text{Student} \times \text{Enroll}\right) \\
SUSAN\_COURSE\_NAMES & \leftarrow \pi_{\text{cname}}(S_{SMITH} \times \text{Course})
\end{align*}
\]

(c) For each course, list the number of books used in that course.
\[
\begin{align*}
\text{COURSE\_BOOK} & \leftarrow \text{course}\# \ \exists_{\text{COUNT}} \text{BookAdoption}
\end{align*}
\]

(d) Find the departments whose offered courses have total enrollment more than 1000 in spring 2005.
\[
\begin{align*}
SP05 & \leftarrow \sigma_{\text{Semester='Spring2005'}} \text{Enroll} \\
\text{DEPT\_ENROLL}(\text{DEP, DCOUNT}) & \leftarrow \text{dept} \ \exists_{\text{Count}} (\text{Course} \times SP05) \\
\text{ENROLL\_1000} & \leftarrow \Pi_{\text{DEP}} \sigma_{\text{DCOUNT}>1000} \text{DEPT\_ENROLL}
\end{align*}
\]
3. **SQL Queries:** (25 pts) Consider the following relations for keeping track of students and the jobs they are seeking

\[
\text{Student(*sno, sname, major, GPA)} \\
\text{Appliedfor(*asno, *ajno)} \\
\text{Job(*jno, jname, company, salary)}
\]

Sample tuple for each relation is given below

\[
\text{Student(3014, John Smith, Computer Science, 3.2)} \\
\text{Appliedfor(3014, 24)} \\
\text{Job(24, Programmer, IBM, 60000)}
\]

Keys are denoted by ‘*’. Write the following queries in SQL.

(a) Find the number of jobs 'John Smith' applied for.

\[
\text{select count(*)} \\
\text{from Student, Appliedfor} \\
\text{where sno=asno and sname = 'John Smith'}
\]

(b) Print the names of students who hasn’t applied for a job.

\[
\text{select sname} \\
\text{from Student} \\
\text{where not exists (select *} \\
\text{from Appliedfor} \\
\text{where sno=asno)}
\]

(c) For each student, print the name of the student and the average salary of the jobs he/she has applied for.

\[
\text{select sname, avg(salary)} \\
\text{from Student, Appliedfor, Job} \\
\text{where sno=asno and jno=ajno} \\
\text{group by sname}
\]

(d) For each major with more than ten students, print the major name and the maximum and minimum GPA for students in that major.

\[
\text{select major, max(gpa), min(gpa)} \\
\text{from Student} \\
\text{group by major} \\
\text{having count(*)>10}
\]
4. **SQL Queries:** (25 pts) Consider the following database scheme that keeps track of World War II capital ships. It involves the following relations

- Classes(*cclass, type, country, numguns, bore, displacement)
- Ships(*sname, sclass, launched)
- Battles(*sname, date)
- Outcomes(*ship, battle, result)

Sample tuple for each relation is given below

- Classes(Bismark, bb, Germany, 8, 15, 42000)
- Ships(Royal Oak, Revenge, 1916)
- Battles(Guadalcanal, 11/15/42)
- Outcomes(Bismark, Danmark Strait, Sunk)

Ships are built in *classes* from the same design and the class is usually named for the first ship of that class. The relation *classes* records the name of the class, the type (bb for battleship or bc for battlecruiser), the country that built the ship, the number of main guns, the bore (diameter of the gun barrel) of the main guns, and the displacement (weight, in tons). Relation *ships* records the name of the ship, the name of its class and the year in which the ship was launched. Relation *battles* gives the name and date of battles involving the ships. The relation *Outcomes* gives the result (sunk, damaged or ok) for each ship in the battle.

Keys are denoted by *. Specify the following queries in **SQL**

(a) Find the average number of guns of battleships.

```sql
select avg(numguns)
from Classes
where type = 'bb'
```

(b) Find the battles in which ships of the *Revenge* class participated.

```sql
select battle
from Outcomes
where ship in (select sname
                from ships
                where sclass = Revenge)
```

(c) Find for each class the year in which the first ship of that class was launched.

```sql
select cclass, min(launched)
from Classes, Ships
where cclass = sclass
group by cclass
```
(d) Find the ships sunk in the battle of *Denmark Strait*.

```sql
select ship
from Outcomes
where battle = 'Denmark Strait' and result = 'Sunk'
```