1. Explain the following terms. [15pts]
   a) database schema and instance
      schema: the **logical structure** (overall design) of the database
      instance: the actual **content** (information) of the database at a particular point in time
   b) DDL, DML, and query
      DDL: Language for defining the database schema
      DML: Language for accessing and manipulating the data
      Query: part of DML that requests data retrieval
   c) foreign key
      A value in one relation must appear in another relation.

2. Prove or disprove $\Pi_A (r \cap s) = \Pi_A (r) \cap \Pi_A (s)$. [10pts]
   False. A counterexample: $R, S = (A, B), r = \{(a, b)\}, s = \{(a, c)\}$
   $\Pi_A (r \cap s) = \emptyset, \Pi_A (r) \cap \Pi_A (s) = \{a\}$

3. The **anti-join**, written as $r \bowtie s$ where $r$ and $s$ are relations, is similar to the natural join, but its result is only those tuples in $r$ for which there is no tuple in $s$ that is equal on their common attribute names. For example, the result of the anti-join course $\bowtie$prereq is as follows:

<table>
<thead>
<tr>
<th>course_id</th>
<th>title</th>
<th>dept_name</th>
<th>credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIO-301</td>
<td>Genetics</td>
<td>Biology</td>
<td>4</td>
</tr>
<tr>
<td>CS-190</td>
<td>Game Design</td>
<td>Comp. Sci.</td>
<td>4</td>
</tr>
<tr>
<td>CS-315</td>
<td>Robotics</td>
<td>Comp. Sci.</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>course_id</th>
<th>prereq</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIO-301</td>
<td>BIO-101</td>
</tr>
<tr>
<td>CS-190</td>
<td>CS-101</td>
</tr>
<tr>
<td>CS-315</td>
<td>CS-101</td>
</tr>
</tbody>
</table>

   a) Define anti-join operation, $r \bowtie s$, in terms of the basic operations and natural join operation. [10pts]
   $r \bowtie = \Pi_{r.a_1, \ldots, r.a_n} (r \setminus r \cap s)$
   b) Find the IDs and titles of courses in the Comp. Sci. department that do not have any prerequisites, using NOT EXISTS clause. [5pts]
      SELECT c.course_id, c.title
      FROM course c
      WHERE NOT EXISTS (SELECT p.course_id
                          FROM prereq p
                          WHERE c.course_id = p.course_id)
      AND c.dept_name = 'Comp. Sci.';

4. Write a SQL query for the following relational algebra expression: [10pts]
   $\Pi_{\text{dept name}} (\sigma_{\text{salary} > 80000} (\text{instructor}))$
   SELECT DISTINCT dept_name
   FROM instructor
   WHERE salary > 80000;

5. a) What is a **null** value? [2pts]
    An unknown value or that a value does not exist
   b) What is the result for each of the following aggregate functions? [8pts]
      | department | budget |
      |------------|--------|
      | Biology    | 80000  |
      | Comp. Sci. | 100000 |
      | Elec. Eng. | 70000  |
      | History    | 70000  |
      | Music      | Packard | null |
Consider the following database for problem 6 and 7.

\[
\text{movie}(\text{title}, \text{director\_name}, \text{running\_time}) \\
\text{actor}(\text{title}, \text{actor\_name}, \text{role}) \\
\text{theater}(\text{theater\_name}, \text{address}, \text{phone}) \\
\text{schedule}(\text{theater\_name}, \text{title}, \text{showtime})
\]

6. Give an expression in the relational algebra for each of the following queries: [20pts]
   a) Find the names of actors who appeared in a movie titled “Godfather”.
      \[\pi_{\text{actor\_name}}(\sigma_{\text{title} = \text{"Godfather"}}(\text{actor}))\]
   b) Find the names of directors who appeared in their own movie.
      \[\pi_{\text{director\_name}}(\sigma_{\text{director\_name} = \text{actor\_name}}(\text{movie} \bowtie \text{actor}))\]
   c) Find the names of theaters showing a movie which was directed by “Tim Burton” or in which “Johnny Depp” appeared.
      \[\pi_{\text{theater\_name}}(\sigma_{\text{director\_name} = \text{"Tim Burton"}}(\text{movie} \bowtie \text{schedule})) \cup \pi_{\text{theater\_name}}(\sigma_{\text{actor\_name} = \text{"Johnny Depp"}}(\text{actor} \bowtie \text{schedule}))\]
   d) Find the names of theaters showing a movie which is not showing in any other theaters, with the titles of the movies.
      \[\pi_{\text{theater\_name}, \text{title}}(\text{schedule}) - \pi_{\text{theater\_name}, \text{title}}(\sigma_{\text{schedule.theater\_name} = \text{theater\_name} \land \text{schedule.title} = \text{title}}(\text{schedule} \bowtie s(\text{schedule})))\]

7. Write the following queries in SQL: [20pts]
   a) Find the address and phone number of a theater named ‘Cinecube’.
      \[
      \text{SELECT address, phone} \\
      \text{FROM theater} \\
      \text{WHERE theater\_name = \text{"Cinecube"};}
      \]
   b) Find the names of theaters and showtimes for a movie titled ‘Les Miserables’ in ascending order of showtime.
      \[
      \text{SELECT theater\_name, showtime} \\
      \text{FROM schedule} \\
      \text{WHERE title = \text{"Les Miserables"} \land showtime\_order = \text{ASC};}
      \]
   c) Display the schedules(all attributes of schedule) of all movies in the database in which ‘Brad Pitt’ and ‘Angelina Jolie’ do not appear.
      \[
      \text{SELECT theater\_name, title, showtime} \\
      \text{FROM schedule NATURAL JOIN actor} \\
      \text{WHERE actor\_name NOT IN (\text{"Brad Pitt"}, \text{"Angelina Jolie"});}
      \]
   d) Display the list of all movie titles, with the total number of showings of each movie. Make sure to correctly handle movies with no showings.
      \[
      \text{SELECT title, count(showtime)} \\
      \text{FROM movie NATURAL LEFT OUTER JOIN schedule} \\
      \text{GROUP BY title}
      \]