1. **(8 points)** The ACID properties are Atomicity, Consistency, Isolation, and Durability. Provide a one sentence description of each of these properties:

   Atomicity – Transactions are "atomic". That is, a transaction executes completely or not at all.

   Consistency – Database integrity is maintained. The database moves from consistent state to consistent state.

   Isolation – Transactions that overlap in time are isolated from each other. Individual transactions execute as if there were no other transactions in the system.

   Durability – All updates from completed transaction are permanent.

2. **(2 points)** The category of data model which reflects the data as seen by the user is referred to as a **either Conceptual or High-Level** data model.

3. **(2 points)** The category of data model which includes all data stored in the database (for example, has all the entities, attributes, and relationships) is referred to as a **Logical** data model.

4. **(2 points)** The category of data model which includes all data as seen by the data administrator (including indices, access paths, etc.) is referred to as a **either Physical or Low-Level** data model.

5. **(6 points)** Fill in the blanks: In the three schema architecture, the three schemas are:

   **External** ________ schemas: Schemas as seen by the users or applications

   **Conceptual** ________ schema: Shows the organization of all the data

   **Internal** ________ schema: The actual database schema

6. **(4 points)** What do the acronyms DDL and DML stand for?

   - DDL – Data Definition Language
   - DML – Data Manipulation Language

7. **(2 points)** In database programming what is a Host Language?

   A Host Language is an ordinary programming language in which you embed calls to the database management system.
8. **(10 points)** In terms of the Relational Data Model define the following:

**Superkey:**

A subset of the fields of the relation that uniquely determines the values of all the other fields of the relation.

**Candidate Key:**

A superkey that has the additional property that no field can be removed from the subset and still be a superkey.

**Primary Key:**

A chosen distinguished Candidate Key. There can be multiple Candidate Keys, there can be only one Primary Key.

**Alternate Key:**

A Candidate Key other than the Primary Key.

**Foreign Key:**

A subset of the fields of one relation that is a Candidate Key (usually the Primary Key) of another relation.

9. **(2 points)** Describe the difference between Relation Schemas and Relation States.

A Relation Schema is the definition of the relation. It is fixed over time.

A Relation State refers to the actual set of tuples in the relation at a particular point in time.

Or,

A Relation Schema is the static definition of a relation. A relation state is the dynamic value of that relation at a point in time.

10. **(4 points)** Briefly explain the relationship between Redundancy in a database and Update Anomalies.

Redundancy in a database means that the same fact is stored in multiple places in the database. When updated, those redundant facts must be correctly updated in all places where they are stored. Update anomalies refers to the need to make multiple updates to correctly reflect the change in a single fact, and the idea that if you fail to make all the required updates, the database will be in an inconsistent state.
Consider the following Entity Relationship diagram:

Are the following True or False. (Write T or F next to each question.)

11. ____ (2 points) A person has exactly one name.
12. ____ (2 points) A name is a simple (or atomic) type.
13. ____ (2 points) A person has exactly one phone number.
14. ____ (2 points) The Project Number uniquely determines the project.
15. ____ (2 points) Hourly Salary is independent of the project a person is working on.
Consider the following relational schema.

**Employee**

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Sex</th>
<th>DeptNum</th>
</tr>
</thead>
</table>

**Department**

<table>
<thead>
<tr>
<th>DeptNum</th>
<th>DeptName</th>
</tr>
</thead>
</table>

**Department_Locations**

<table>
<thead>
<tr>
<th>DeptNum</th>
<th>DeptLoc</th>
</tr>
</thead>
</table>

**Project**

<table>
<thead>
<tr>
<th>ProjName</th>
<th>ProjNum</th>
<th>ProjLoc</th>
<th>DeptNum</th>
</tr>
</thead>
</table>

**Works_On**

<table>
<thead>
<tr>
<th>SSN</th>
<th>ProjNum</th>
<th>Hours</th>
</tr>
</thead>
</table>

Express the following queries using relational algebra. (If you can’t remember the symbols for the relational operators, write out the names.)

16. (8 points) Retrieve the names of all employees in Department 6.

\[
\text{Result} = \pi_{\text{Name}} ( \sigma_{\text{DeptNum} = 6}( \text{Employee} ))
\]

17. (8 points) Retrieve the names of all employees in Department 6 who do not work on Project XYZ.

\[
\begin{align*}
\text{Dept6} \_ \text{SSN} &= \pi_{\text{SSN}} ( \sigma_{\text{DeptNum} = 6}( \text{Employee} )) \\
\text{ProjXYZ} \_ \text{Num} &= \pi_{\text{ProjNum}} ( \sigma_{\text{ProjName} = \text{'XYZ'}}( \text{Project} )) \\
\text{ProjXYZ} \_ \text{SSN} &= \pi_{\text{SSN}} ( \text{Works}_\_ \text{On} \times \pi_{\text{ProjNum}}( \text{ProjXYZ} \_ \text{Num} )) \\
\text{Result} &= \pi_{\text{Name}} ( \text{Employee} \times \pi_{\text{SSN}}( \text{Dept6} \_ \text{SSN} – \text{ProjXYZ} \_ \text{SSN} ))
\end{align*}
\]

18. (8 points) For each project list the project name and the total hours per week (by all employees) spend on the project.

\[
\text{Result} = \text{ProjName} \sum \text{Hours} ( \text{Project} \times \pi_{\text{ProjNum}}( \text{ProjXYZ} \_ \text{SSN} ))
\]

19. (8 points) Find the names of all employees who work on at least one project located in Houston but whose department has no location in Houston.

\[
\begin{align*}
\text{HoustonProjects} &= \pi_{\text{ProjNum}} ( \sigma_{\text{ProjLoc} = \text{'Houston'}}( \text{Project} )) \\
\text{HoustonProjects} \_ \text{SSN} &= \pi_{\text{SSN}} ( \text{HoustonProjects} \times \pi_{\text{ProjNum}}( \text{ProjXYZ} \_ \text{SSN} )) \\
\text{HoustonDepartment} \_ \text{DeptNum} &= \pi_{\text{DeptNum}} ( \sigma_{\text{DeptLoc} = \text{'Houston'}}( \text{Department} \_ \text{Locations} )) \\
\text{All} \_ \text{DeptNum} &= \pi_{\text{DeptNum}}( \text{Department} ) \\
\text{NonHouston} \_ \text{DeptNum} &= \text{All} \_ \text{DeptNum} – \text{HoustonDepartment} \_ \text{DeptNum} \\
\text{NonHouston} \_ \text{SSN} &= \pi_{\text{SSN}} ( \text{Employee} \times \pi_{\text{DeptNum}}( \text{NonHouston} \_ \text{SSN} )) \\
\text{Result} &= \pi_{\text{Name}} ( \text{Employee} \times \pi_{\text{SSN}}( \text{HoustonProjects} \_ \text{SSN} – \text{NonHouston} \_ \text{SSN} ))
\end{align*}
\]
Consider the following relation with the Functional Dependencies as indicated:

<table>
<thead>
<tr>
<th>Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN</td>
</tr>
</tbody>
</table>

{SSN} → Name
{DeptNo} → DName
{ProjNo} → DeptNo, PName, Budget
{DeptNo, PName} → ProjNo, Budget
{SSN, ProjNo} → Hours

20. (16 points) Provide a Third Normal Form (3NF) Schema for the above relation.

Functional Dependencies:

- Projects

People

Departmentss

Projects

Hours

(SSN, ProjNo) is Primary Key