**BASIC TASKS**

1. **Define the following terms**:
   * **Candidate Key**: A column or combination of columns that can uniquely identify a row in a table.
   * **Composite Key**: A primary key composed of multiple columns used to uniquely identify a row.
   * **Foreign Key**: A column in one table that references the primary key in another table, establishing a relationship between the two tables.
   * **Functional Dependency**: A relationship where the value of one attribute (or a set of attributes) determines the value of another attribute.
2. **Identify the three integrity rules/constraints in the relational model**:
   * **Entity Integrity**: Each table must have a unique, non-null primary key.
   * **Referential Integrity**: Foreign key values must match primary key values in the related table.
   * **Domain Integrity**: Ensures that all columns are of a consistent data type and meet predefined constraints.
3. **Check for violations of relational integrity rules**:
   * Review given relational tables and check for duplicate, null, or invalid references within primary keys and foreign keys.
   * **Table 1**: There’s a foreign key violation with directorNo = 753 not matching any Director table entry.
   * **Table 2**: Possible redundancy in Supplier and PartNo combinations if they’re not explicitly defined as a composite primary key.
   * **Table 3**: No apparent primary key violations if Song and Artist are used as a composite key, but consider additional constraints for clarity.

**MEDIUM TASKS**

1. **Analyze anomalies in the Project-Employee table**:
2.  **Insertion Anomaly** is resolved by separating project and employee information, allowing projects to be added independently of employees.
3.  **Deletion Anomaly** is mitigated by storing employee information separately, so removing a project does not delete associated employee records.
4.  **Update Anomaly** is resolved by storing department information in its own table, so updating a department for an employee only needs to occur in the Employee table.
5. **Normalization**:
   * **1NF**: Ensure each field holds atomic values only.
   * **2NF**: Remove partial dependencies so that all non-key attributes fully depend on the primary key.
   * **3NF**: Eliminate transitive dependencies so that non-key attributes depend only on the primary key.
6. **Normalize the bakery customer information table**:
   * **Anomalies**:
     + **Insertion**: Inserting a new order without customer information might cause issues.
     + **Deletion**: Deleting an order could unintentionally remove customer details.
     + **Update**: Updating order status could lead to inconsistent data.

**Normalization**:

* + **1NF**: Ensure each attribute contains only atomic values.
  + **2NF**: Remove partial dependencies, ensuring non-key attributes fully depend on the primary key.
  + **3NF**: Remove transitive dependencies, so non-key attributes depend only on the primary key.

1. **Generate schema and create relationships and ER Diagram for given tables**:
   * Define primary and foreign keys for each table and use them to establish relationships.
   * Create an ER diagram that shows the connections between these tables based on their relationships.

**ADVANCED TASKS**

1. **Identify functional dependencies and complete 1NF, 2NF, and 3NF for given tables**:
   * **1NF**: Ensure each field contains atomic values.
   * **2NF**: Remove partial dependencies by ensuring all non-key attributes fully depend on the primary key.
   * **3NF**: Eliminate transitive dependencies so that non-key attributes depend only on the primary key.
2. **Normalize the car rental company’s quarterly report table**:
   * **Functional Dependencies**:
     + Identify dependencies between branch details, car information, bills, and supervisor details.
   * **Normalization Steps**:
     + **1NF**: Remove repeated data in each field.
     + **2NF**: Remove partial dependencies.
     + **3NF**: Eliminate transitive dependencies.