Problem 1: Sales and Inventory Management System

Scenario: A company has a sales team with several employees. Each salesperson can manage multiple customers, and each customer can place many orders. Orders can contain several inventory items, and these items are made up of various parts. Employees are responsible for assembling the inventory items from parts. Suppliers provide parts, which may be used in several inventory items.

Requirements:

- **Construct an ER diagram for the sales and inventory management system using Lucidchart.**
- **Export the ER diagram as a PDF and deliver it.**
- **List any assumptions you have made while designing the ER diagram.**
- **Clearly indicate the cardinality mappings (e.g., one-to-many, many-to-many) and any role indicators within the ER diagram.**
- **Create a PostgreSQL database schema that can be used to implement the system, including tables for employees, salespeople, customers, orders, inventory items, parts, and suppliers.**
- **Include a report that contains:**
    - **A chart visualizing the relationships between entities.**
    - **The list of assumptions made.**
    - **A screenshot of the PostgreSQL schema.**

Deliverables:

1. **ER diagram (PDF format)**
2. **List of assumptions made**
3. **Database schema in PostgreSQL (CREATE TABLE statements)**
4. **Report with:**
    - **Chart visualizing entity relationships**
    - **List of assumptions**
    - **Screenshot of PostgreSQL schema**

Problem 2: Relationship Rules for ERD

Scenario: You are given an existing ER diagram for a business process (such as a sales management system). You need to analyze the relationships between entities in the diagram and define the bi-directional rules for each relationship. These rules will define how entities interact with one another based on cardinality and role indicators.

Requirements:

- **Construct an ER diagram based on the provided business requirements for a system, using Lucidchart.**
- **Export the ER diagram as a PDF and deliver it.**
- **List any assumptions you have made while designing the ER diagram.**

- **Clearly indicate the cardinality mappings and role indicators within the ER diagram.**
- **Create a PostgreSQL database schema that can be used to implement the database, detailing the relationships between entities.**
- **Include a report that contains:**
    - **A chart visualizing the relationships between entities.**
    - **The list of assumptions made.**
    - **A screenshot of the PostgreSQL schema.**

Deliverables:

1. **ER diagram (PDF format)**
2. **List of assumptions made**
3. **Database schema in PostgreSQL (CREATE TABLE statements)**
4. **Report with:**
    - **Chart visualizing entity relationships**
    - **List of assumptions**
    - **Screenshot of PostgreSQL schema**

Problem 3: NHL Database Design

Scenario: The National Hockey League (NHL) consists of multiple teams, each with players, coaches, and captains. Each player belongs to only one team, has a name, a position, and a skill level. Some players may have injury records. Games are played between two teams, and each game has a date and score.

Requirements:

- **Construct an ER diagram for the NHL database using Lucidchart.**
- **Export the ER diagram as a PDF and deliver it.**
- **List any assumptions you have made while designing the ER diagram.**
- **Clearly indicate the cardinality mappings (e.g., one-to-many, many-to-many) and any role indicators within the ER diagram.**
- **Create a PostgreSQL database schema that can be used to implement the NHL database, with tables for teams, players, games, and injury records.**
- **Include a report that contains:**
    - **A chart visualizing the relationships between entities.**
    - **The list of assumptions made.**
    - **A screenshot of the PostgreSQL schema.**

Deliverables:

1. **ER diagram (PDF format)**

2. **List of assumptions made**

3. **Database schema in PostgreSQL (CREATE TABLE statements)**

4. **Report with:**

    - **Chart visualizing entity relationships**
    - **List of assumptions**
    - **Screenshot of PostgreSQL schema**