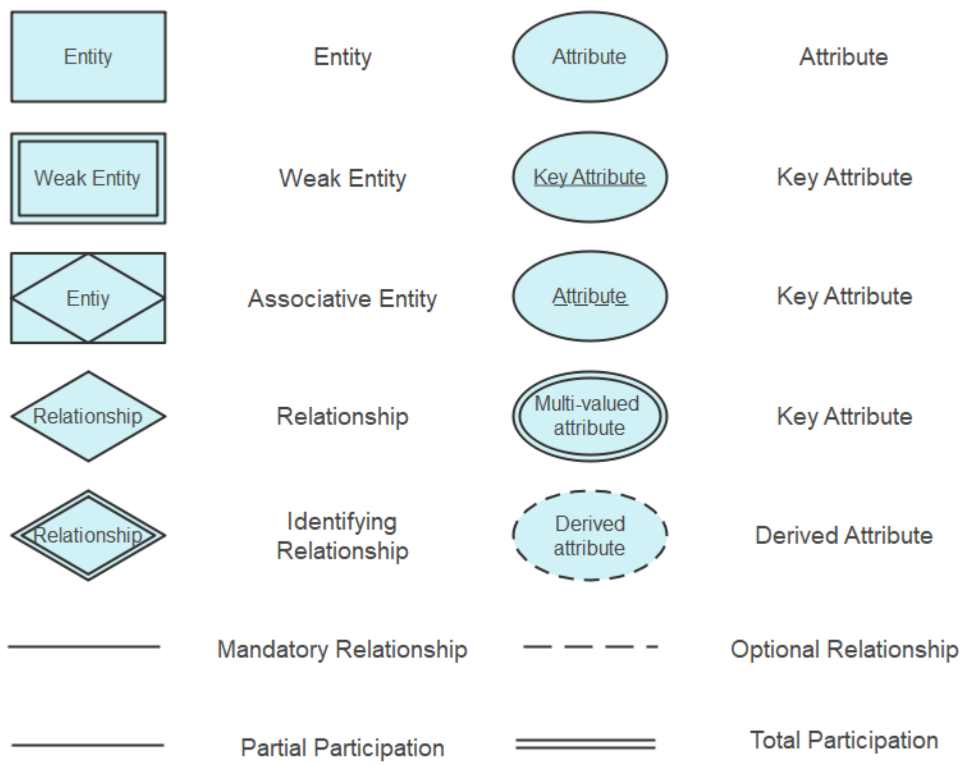
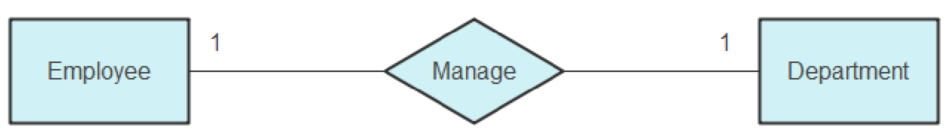
1. Give Symbol used in E-R Diagram and Draw the E-R diagram of Library Management System.

Symbol used in E-R Diagram

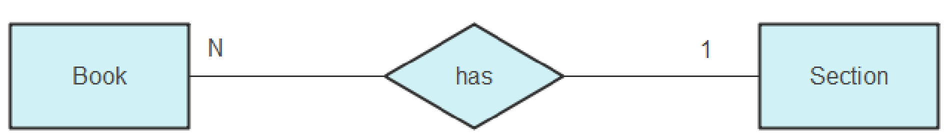


one-to-one (1:1)

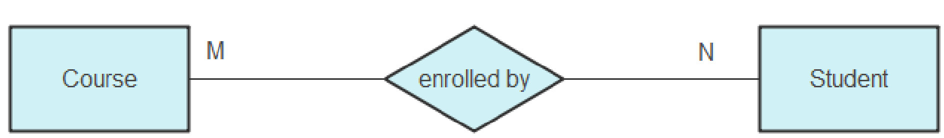


one-to-many (1:N)

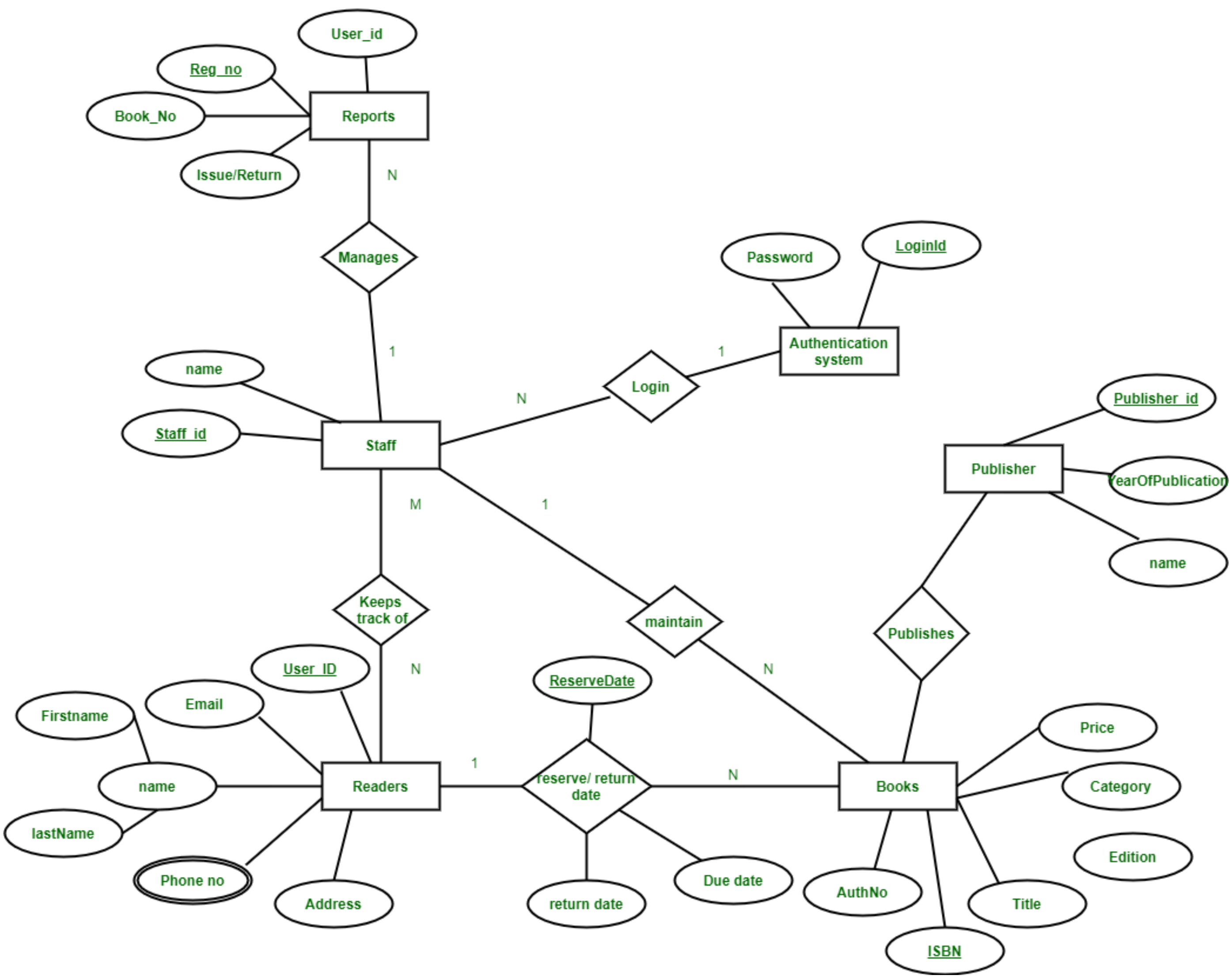
many-to-one (N:1)



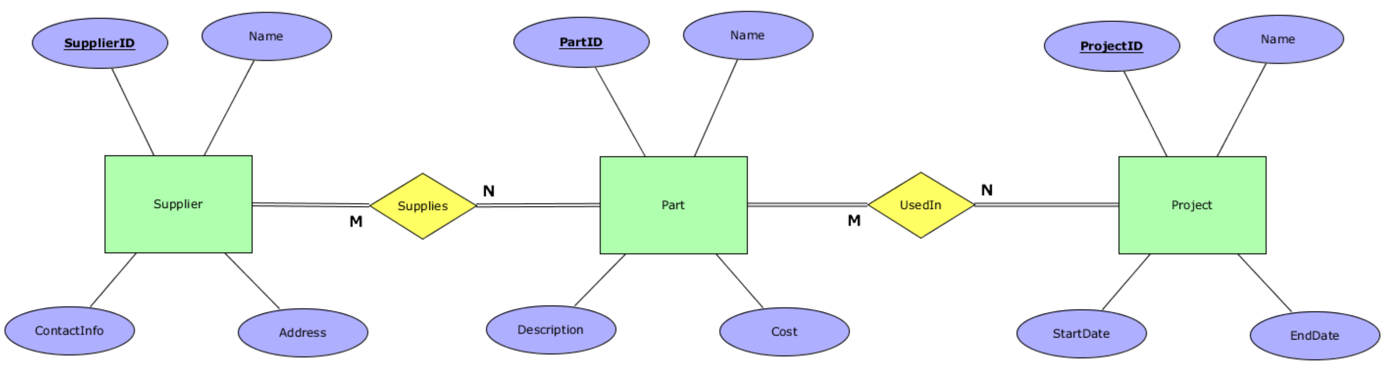
many-to-many (M:N)



E-R diagram of Library Management System



2. Draw E-R diagram for supplier who supplies different parts. The parts are used in different projects. Explain the mapping cardinality used. Assume suitable attributes.



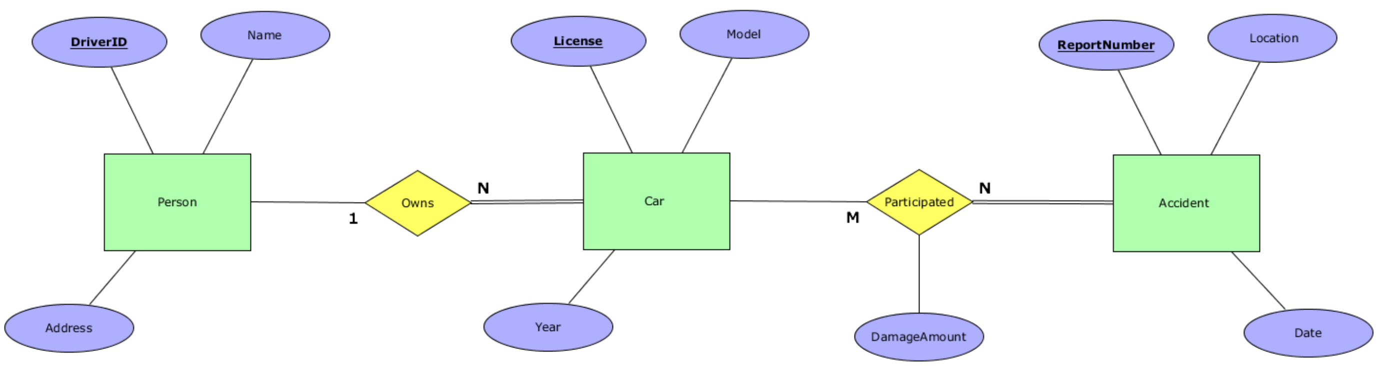
E-R Diagram Explanation:

- **Entities:** Draw rectangles for `Supplier`, `Part`, and `Project`.
- **Relationships:** Draw diamonds for `Supplies` and `UsedIn`.
- **Attributes:** Connect ovals to their respective entities and relationships.
- **Primary Keys:** Underline `SupplierID`, `PartID`, and `ProjectID`.

Mapping Cardinality:

1. **Supplier to Part (M:N):**
   - A supplier can supply multiple parts.
   - A part can be supplied by multiple suppliers.
   - Example: Supplier S1 can supply Parts P1, P2; Part P1 can be supplied by Suppliers S1 and S2.
2. **Part to Project (M:N):**
   - A part can be used in multiple projects.
   - A project can involve multiple parts.
   - Example: Part P1 can be used in Projects X and Y; Project X can use Parts P1, P2, and P3.

3. Construct an E-R Diagram for an insurance company with a set of customers, each of whom owns number of cars, also each can have number of recorded accident associated with it.



4. Given relation R with attributes A, B, C, D, E, F and set of FDs as A-> BC, E-> CF, B ->E and CD->EF. Find out closure {A, B}+ of the set of attributes.

Functional Dependencies:

1. $A \to BC$
2. $E \to CF$
3. $B \to E$
4. $CD \to EF$

Step 1: Start with the initial set $\{A, B\}$.

Initially, $\{A, B\}^+ = \{A, B\}$.

Step 2: Apply the functional dependencies iteratively.

Apply $A \to BC$:

Since A is in $\{A, B\}$, add B and C (from $A \to BC$) to the closure:

$$\{A, B\}^+ = \{A, B, C\}.$$

Apply $B \to E$:

Since B is in the closure, add E (from $B \to E$) to the closure:

$$\{A, B\}^+ = \{A, B, C, E\}.$$

Apply $E \to CF$:

Since E is in the closure, add C (already present) and F (from $E \to CF$) to the closure:

$$\{A, B\}^+ = \{A, B, C, E, F\}.$$

Apply $CD \to EF$:

This dependency requires both C and D in the closure. Since D is not yet in the closure, this dependency does not apply now.

Step 3: Final Closure

The closure of $\{A, B\}$ is:

$$\{A, B, C, E, F\}.$$

5. Consider table R(A, B, C, D, E) with FDs as A->B, BC->E and ED->A. The table is in which normal form? Justify your answer.

Step 1: Find the Candidate Keys

- **Candidate keys** are the smallest sets of attributes that can determine all other attributes in the table.
- Based on the functional dependencies (FDs) given:
  - FDs:
    - $A \to B$
    - $BC \to E$
    - $ED \to A$
- From these, we can conclude that **EDC**, **ACD**, and **BCD** are candidate keys. These sets of attributes can uniquely identify every row in the table.

Step 2: Check for 1NF (First Normal Form)

- **1NF** means that all values in the table must be atomic (indivisible).
- In this case, it's assumed that the table is in 1NF, meaning no multi-valued attributes or nested tables.

Step 3: Check for 2NF (Second Normal Form)

- **2NF** requires the table to be in 1NF and that there are no partial dependencies. A **partial dependency** occurs when a non-prime attribute depends on part of a candidate key.
- Let's analyze the FDs:
  - $A \to B$: A is part of the candidate key **ACD**. But B is a **prime attribute** (it's part of the candidate key), so this is **not** a partial dependency.
  - $BC \to E$: Both B and C are part of a candidate key, so this is **not** a partial dependency.
  - $ED \to A$: E and D are part of a candidate key, so this is **not** a partial dependency.
- **Conclusion**: Since there are no partial dependencies, the table is in **2NF**.

Step 4: Check for 3NF (Third Normal Form)

- **3NF** requires the table to be in 2NF, and for every non-prime attribute to depend only on the candidate key (no transitive dependencies).
- Let's check the FDs:
  - $A \to B$: B is a **prime attribute** (part of a candidate key), so no issue here.
  - $BC \to E$: E is a **prime attribute** (part of a candidate key), so no issue here.
  - $ED \to A$: A is a **prime attribute** (part of a candidate key), so no issue here.
- **Conclusion:** Since all non-prime attributes depend only on candidate keys, the table is in **3NF**.

Step 5: Check for BCNF (Boyce-Codd Normal Form)

- **BCNF** requires that for every functional dependency, the left side (the determinant) must be a **superkey** (i.e., it must be a candidate key or a superset of one).
- Check the FDs:
  - $A \to B$: **A** is **not** a superkey (it doesn't determine all attributes), so this violates BCNF.
  - $BC \to E$: **BC** is a candidate key, but it's **not** a superkey in every case.
  - $ED \to A$: **ED** is **not** a superkey in every case.
- **Conclusion**: Since the left side of all the FDs is not a superkey, the table is **not in BCNF**.

Final Conclusion:

- The table is in **3NF** but **not in BCNF**.