In MySQL, **wildcards** are used with the LIKE operator for pattern matching:

1. **%**: Matches zero or more characters.
   * Example: name LIKE 'A%' (names starting with "A").
2. **\_**: Matches exactly one character.
   * Example: name LIKE 'A\_' (two-character names starting with "A").
3. **Escape Wildcards**: Use ESCAPE to match % or \_ literally.
   * Example: name LIKE 'A\%' ESCAPE '\' (matches "A%").

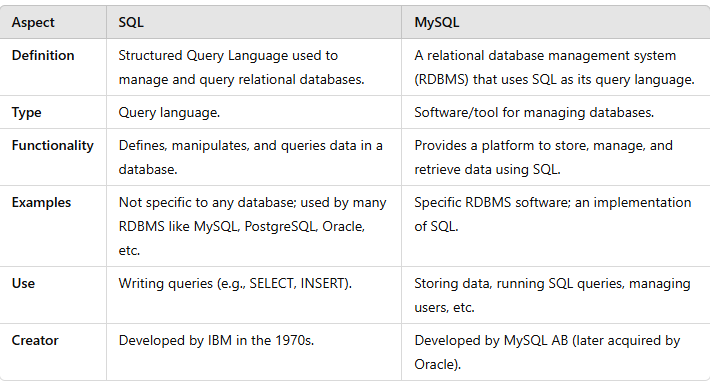
**Example Query**:

SELECT \* FROM users WHERE name LIKE '%John%';

Matches any name containing "John".

 **SQL** is the language.

 **MySQL** is an RDBMS that uses SQL.



A **relationship** in databases connects tables via keys (primary/foreign).

**Types of Relationships:**

1. **One-to-One (1:1)**: Each row in Table A matches one row in Table B.
   * Example: A person ↔ a passport.
2. **One-to-Many (1:N)**: One row in Table A links to many in Table B.
   * Example: A customer ↔ multiple orders.
3. **Many-to-One (N:1)**: Many rows in Table A link to one in Table B.
   * Example: Employees ↔ one department.
4. **Many-to-Many (M:N)**: Many rows in Table A link to many in Table B via a junction table.
   * Example: Students ↔ courses.
5. **Self-Referencing**: A table relates to itself.
   * Example: Employees ↔ managers.

Here’s a concise explanation of **Primary Key**, **Super Key**, and **Candidate Key** in databases:

1. **Primary Key**
   * A unique key that identifies each row in a table.
   * Must be unique and not NULL.
   * **Example**: StudentID in a Students table.
2. **Super Key**
   * Any set of attributes that uniquely identifies a row in a table (can include extra, unnecessary columns).
   * **Example**: {StudentID}, {StudentID, Name}.
3. **Candidate Key**
   * A minimal **super key**—it uniquely identifies a row without any unnecessary attributes.
   * **Example**: {StudentID}, but not {StudentID, Name}.

**Relationship:**

* **Candidate Key** ⊆ **Super Key**
* **Primary Key** is chosen from **Candidate Keys**.