Detailed Comparison: Cardinality vs Constraints in DBMS

Aspect	Cardinality	Constraints
Definition	Describes the numerical relationship (mapping) between entities in an ER model or tables in a database.	Rules applied to table columns or relationships to enforce data integrity, accuracy, and validity.
Purpose	Determines how many instances of one entity can be associated with instances of another entity.	Ensures that only valid data is stored in the database, preventing invalid, duplicate, or inconsistent entries.
Types	- One-to-One (1:1): Each entity in A relates to one in B. - One-to-Many (1:N): One in A relates to many in B. - Many-to-One (N:1): Many in A relate to one in B. - Many-to-Many (M:N): Many in A relate to many in B.	- NOT NULL: Prevents null values. - UNIQUE: Ensures all values are unique. - PRIMARY KEY: Uniquely identifies rows. - FOREIGN KEY: Enforces referential integrity. - CHECK: Restricts values based on a condition. - DEFAULT: Sets a default value.
Enforcement	Implemented during database design, often visualized in ER diagrams, and realized through foreign keys and table structure.	Enforced at the schema level using SQL statements (e.g., `CREATE TABLE`, `ALTER TABLE`).
Scope	Focuses on the structure of relationships between tables/entities (how they are linked and in what quantity).	Applies to columns, rows, or tables, governing data entry, updates, and deletions.
Examples	- A department has many employees (1:N). - A student enrolls in multiple courses, and each course has multiple students (M:N).	- EmployeeID must be unique (UNIQUE). - Salary must be between 20,000 and 100,000 (CHECK). - Every order must have a valid customer ID (FOREIGN KEY).
Visualization	Shown in ER diagrams using lines and symbols (crow's foot notation, etc.).	Not typically visualized in ER diagrams, but defined in DDL and sometimes shown in schema diagrams.

Relation to Data	Can also refer to the uniqueness of values in a column (high or low cardinality).	Directly restricts what data can be stored or how tables relate, but not about the uniqueness or frequency of values themselves.

Additional Details

Cardinality in Practice:

- Cardinality is crucial for understanding and designing the relationships between tables, which affects normalization, query performance, and integrity.
- Examples:
  - One-to-One: Each student has one student ID.
  - One-to-Many: One customer places many orders.
  - Many-to-Many: Students enroll in multiple courses, and each course has multiple students.

Constraints in Practice:

- Constraints are implemented in SQL as part of table definitions to enforce rules and prevent invalid data.
- Examples:
  - NOT NULL: Ensures a column cannot have NULL values.
  - UNIQUE: Ensures all values in a column are distinct.
  - PRIMARY KEY: Uniquely identifies each row in a table.
  - FOREIGN KEY: Ensures a column's values exist in another table, maintaining referential integrity.
  - CHECK: Restricts values based on a condition (e.g., salary between $20,000 and $100,000).