Power Learn Project

RELATIONSHIP TYPES

week 3

Power Learn Project

Database Design

You are live with Gerald Macherechedze

Learning Objectives

1. **Understand the concept of relationships in a relational database.**

2. **Learn about one-to-one relationships between tables.**

3. **Explore one-to-many relationships and their implications.**

4. **Comprehend many-to-many relationships and their representation.**

5. **Understand the use of primary keys to establish relationships.**

6. **Learn about foreign keys as references to primary keys in related tables.**

7. **Explore cascading actions (CASCADE, SET NULL, SET DEFAULT) in relationships.**

8. **Understand the concept of referential integrity in database design.**

9. **Learn to identify and implement relationships in a database schema.**

INTRODUCTION TO Relationship Types

- Relationships help you see how different parts of a system affect each other.
- For example, the entities STUDENT and COURSE are related to each other.
- To accurately model the business, the relationships between entities are as important as the entities themselves.

Understanding Relationships:

- Represent something of significance or importance to the business
- Show how entities are related to each other
- Exist only between entities (or one entity and itself)
- Are bi-directional
- Are named at both ends
- Have optionality
- Have cardinality

User Relationship Types Summary:

One-to-One Relationship:

- **Notes:** In a one-to-one relationship, each record in Table1 is associated with exactly one record in Table2, and vice versa. This is achieved by having a unique foreign key in one of the tables.

Syntax:

```
CREATE TABLE Table1 (
 column1 INT PRIMARY KEY,
 column2 VARCHAR(255),
 -- Other columns
);
CREATE TABLE Table2 (
 column3 INT PRIMARY KEY,
 column4 VARCHAR(255),
 foreign_key_column INT UNIQUE,
 FOREIGN KEY (foreign_key_column) REFERENCES Table1(column1));
```

Example: Users and their corresponding User Profiles.

One-to-Many Relationship:

Notes: In a one-to-many relationship, each record in the parent table can be associated with multiple records in the child table, but each record in the child table is associated with only one record in the parent table. This is established by having a foreign key in the child table.

Syntax

```
CREATE TABLE ParentTable (
 parent_id INT PRIMARY KEY,
 parent_name VARCHAR(255),
 -- Other columns
);
CREATE TABLE ChildTable (
 child_id INT PRIMARY KEY,
 child_name VARCHAR(255),
 parent_id INT,
  FOREIGN KEY (parent_id) REFERENCES ParentTable(parent_id)
);
```

Example: One User having multiple Orders.

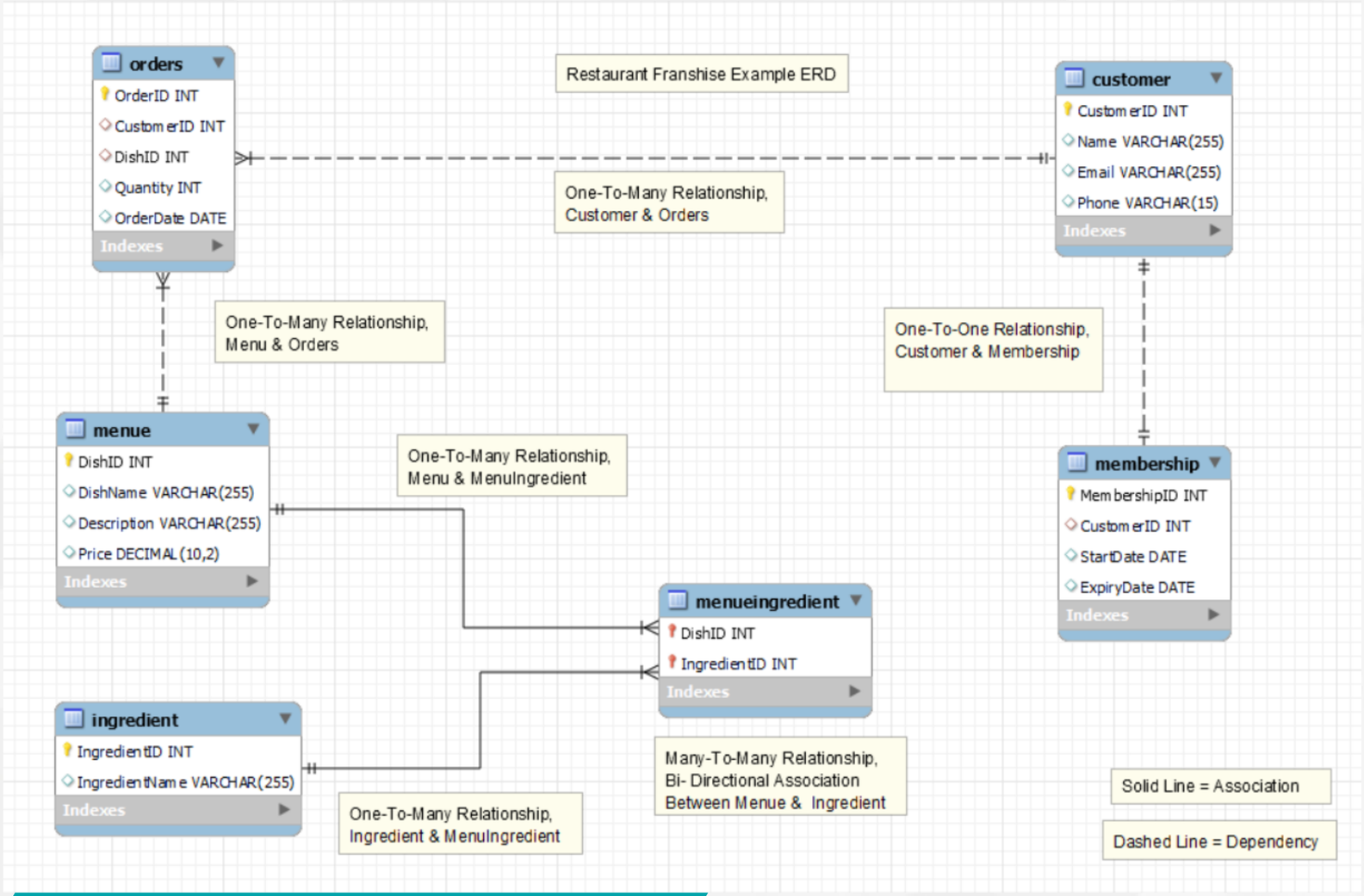
Many-to-Many Relationship:

- Notes: In a many-to-many relationship, records in Table1 can be associated with multiple records in Table2, and vice versa. This is implemented using a junction table, which contains foreign keys referencing both tables.

Syntax

```
CREATE TABLE Table1 (
 table1_id INT PRIMARY KEY,
 column1 VARCHAR(255),
 -- Other columns
);
CREATE TABLE Table2 (
 table2_id INT PRIMARY KEY,
 column2 VARCHAR(255),
 -- Other columns
);
CREATE TABLE JunctionTable (
 table1_id INT,
 table2_id INT,
 PRIMARY KEY (table1_id, table2_id),
 FOREIGN KEY (table1_id) REFERENCES Table1(table1_id),
FOREIGN KEY (table2_id) REFERENCES Table2(table2_id) );
```

PRACTICALS



Power Learn Project

Power Learn Project

Database Design

You are live with Gerald Macherechedze

SQL Queries Click Here

V. Conclusion

- These relationship types are fundamental in designing a normalized and efficient relational database schema.
- Adapt the provided syntax and examples based on the specific database system and requirements.