



## Database Management System

Name of the Faculty: Mr. A.K.Chakravarthy, Assistant Professor, Department of IT

Subject: Database Management System

Year & Semester: II-II

Topic: SQL: Creating Tables with relationship

Conventional Methods: Chalk & Talk

Teaching Methodology: Practical Implementation

In Database Management System, creating tables using SQL is very important topic. If this topic explained only through chalk & talk, students cannot understand practical exposure. So, there is a need to demonstrate creating tables practically.

### **References:**

1. [https://www.w3schools.com/sql/sql\\_create\\_table.asp](https://www.w3schools.com/sql/sql_create_table.asp)
2. <https://www.tutorialspoint.com/sql/sql-create-table.htm>

### **Types of table relationships:**

There are three types of table relationships in Access.

A one-to-many relationship

Let's use an order tracking database that includes a Customers table and an Orders table as an example. A customer can place any number of orders. It follows that for any customer represented in the Customers table, there might be many orders represented in the Orders table. The relationship between the Customers table and the Orders table is a one-to-many relationship.

To represent a one-to-many relationship in your database design, take the primary key on the "one" side of the relationship and add it as an additional field or fields to the table on the "many" side of the relationship. In this case, for example, you add a new field — the ID field from the

Customers table — to the Orders table and name it Customer ID. Access can then use the Customer ID number in the Orders table to locate the correct customer for each order.

### A many-to-many relationship

Now let's look at the relationship between a Products table and an Orders table. A single order can include more than one product. On the other hand, a single product can appear on many orders. Therefore, for each record in the Orders table, there can be many records in the Products table. In addition, for each record in the Products table, there can be many records in the Orders table. This relationship is called a many-to-many relationship. Note that to detect existing many-to-many relationships between your tables, it is important that you consider both sides of the relationship.

To represent a many-to-many relationship, you must create a third table, often called a junction table, that breaks down the many-to-many relationship into two one-to-many relationships. You insert the primary key from each of the two tables into the third table. As a result, the third table records each occurrence, or instance, of the relationship. For example, the Orders table and the Products table have a many-to-many relationship that is defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.

### A one-to-one relationship

In a one-to-one relationship, each record in the first table can have only one matching record in the second table, and each record in the second table can have only one matching record in the first table. This relationship is not common because, most often, the information related in this way is stored in the same table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. When you do identify such a relationship, both tables must share a common field.

### Top of Page

### Why create table relationships?

You can create table relationships explicitly by using the Relationships window, or by dragging a field from the Field List pane. Access uses table relationships to decide how to join tables when you need to use them in a database object. There are several reasons why you should create table relationships before you create other database objects, such as forms, queries and reports.

### Table relationships inform your query designs

To work with records from more than one table, you often must create a query that joins the tables. The query works by matching the values in the primary key field of the first table with a foreign key field in the second table. For example, to return rows that list all of the orders for

each customer, you construct a query that joins the Customers table with the Orders table based on the Customer ID field. In the Relationships window, you can manually specify the fields to join. But, if you already have a relationship defined between the tables, Access supplies the default join, based on the existing table relationship. In addition, if you use one of the query wizards, Access uses the information it gathers from the table relationships you have already defined to present you with informed choices and to prepopulate property settings with appropriate default values.

Table relationships inform your form and report designs

When you design a form or report, Access uses the information it gathers from the table relationships you have already defined to present you with informed choices and to prepopulate property settings with appropriate default values.

Table relationships are the foundation upon which you can enforce referential integrity to help prevent orphan records in your database. An orphan record is a record with a reference to another record that does not exist — for example, an order record that references a customer record that does not exist.

When you design a database, you divide your information into tables, each of which has a primary key. You then add foreign keys to related tables that reference those primary keys. These foreign key-primary key pairings form the basis for table relationships and multi-table queries. It's important that these foreign key-primary key references stay synchronized. Referential integrity, which is dependent on table relationships, helps ensure that references stay synchronized.

### **SQL: Creating Tables with relationship practical demonstration:**

Faculty explained the creation of tables using practical demonstration. In this method, students practically understand how tables are created.



