



# MS Access

Class 8

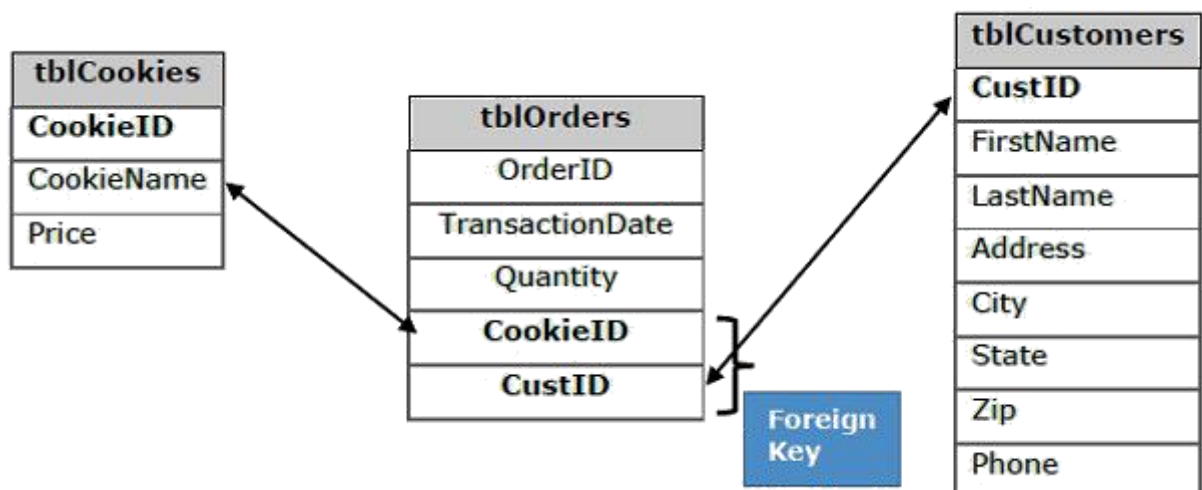
Lab 29

## Lab Objectives:

- Relating Data

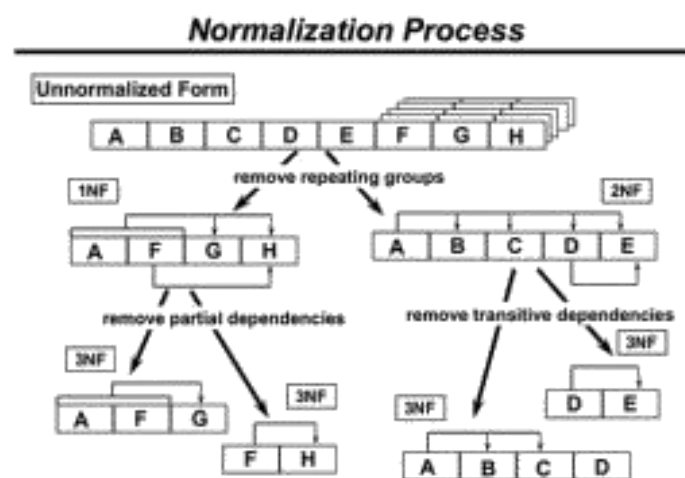
## Relating Data

Before talking about and creating relationships between different data, let us review why we need it. It all goes back to normalization.



## Normalization

Database normalization, or simply normalization, is the process of organizing columns (attributes) and tables (relations) of a relational database to minimize data redundancy. It is the process of splitting data across multiple tables to improve overall performance, integrity and longevity.

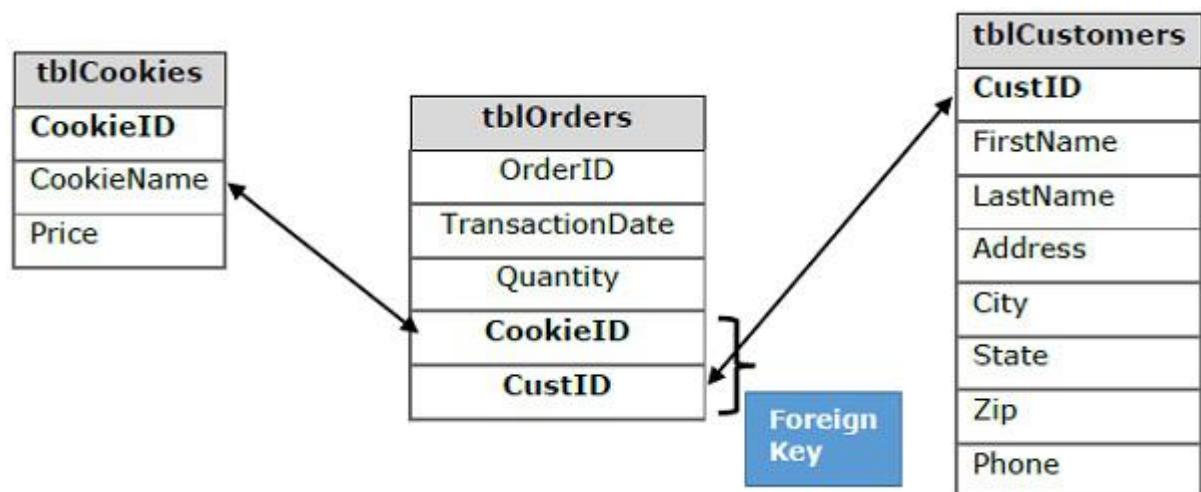


- Normalization is the process of organizing data in a database.
- This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

Let us now look into the following table which contains data, but the problem is that this data is quite redundant which increases the chances of typo and inconsistent phrasing during data entry.

CustID	Name	Address	Cookie	Quantity	Price	Total
1	Ethel Smith	12 Main St, Arlington, VA 22201 S	Chocolate Chip	5	\$2.00	\$10.00
2	Tom Wilber	1234 Oak Dr., Pekin, IL 61555	Choc Chip	3	\$2.00	\$6.00
3	Ethil Smithy	12 Main St., Arlington, VA 22201	Chocolate Chip	5	\$2.00	\$10.00

To solve this problem, we need to restructure our data and break it down into multiple tables to eliminate some of those redundancy as shown in the following three tables.






Here, we have one table for Customers, the 2nd one is for Orders and the 3rd one is for Cookies.

The problem here is that just by splitting the data in multiple tables will not help to tell how data from one table relates to data in another table.

To connect data in multiple tables, we have to add foreign keys to the Orders table.

## Defining Relationships

A relationship works by matching data in key columns usually columns with the same name in both the tables. In most cases, the relationship matches the primary key from one table, which provides a unique identifier for each row, with an entry in the foreign key in the other table. There are three types of relationships between tables. The type of relationship that is created depends on how the related columns are defined.

Customers		
	Field Name	Data Type
	CustomerID	AutoNumber
	Primary Key	Text
	Cut	Text
	Copy	Text

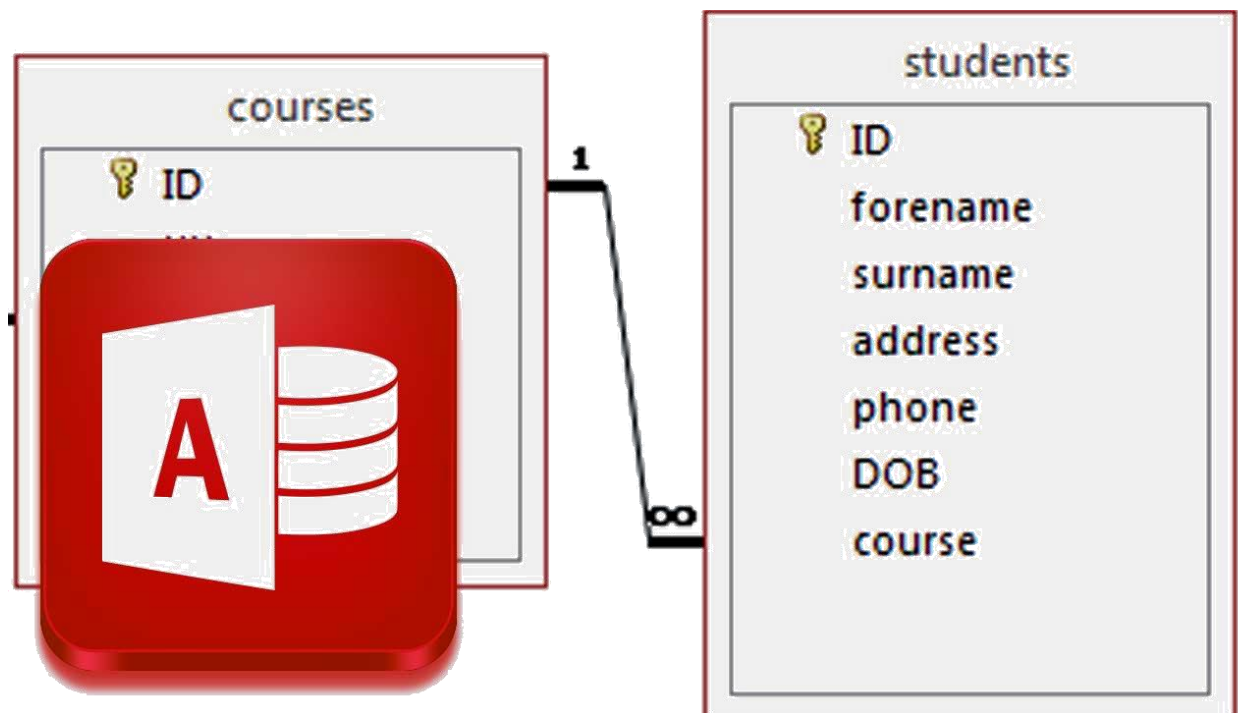
Let us now look into the three types of relationships

—

## One-to-Many Relationships

A one-to-many relationship is the most common type of relationship. In this type of relationship, a row in table A can have many matching rows in table B, but a row in table B can have only one matching row in table A.

For example, the Customers and Orders tables have a one-to-many relationship: each customer can place many orders, but each order comes from only one customer.

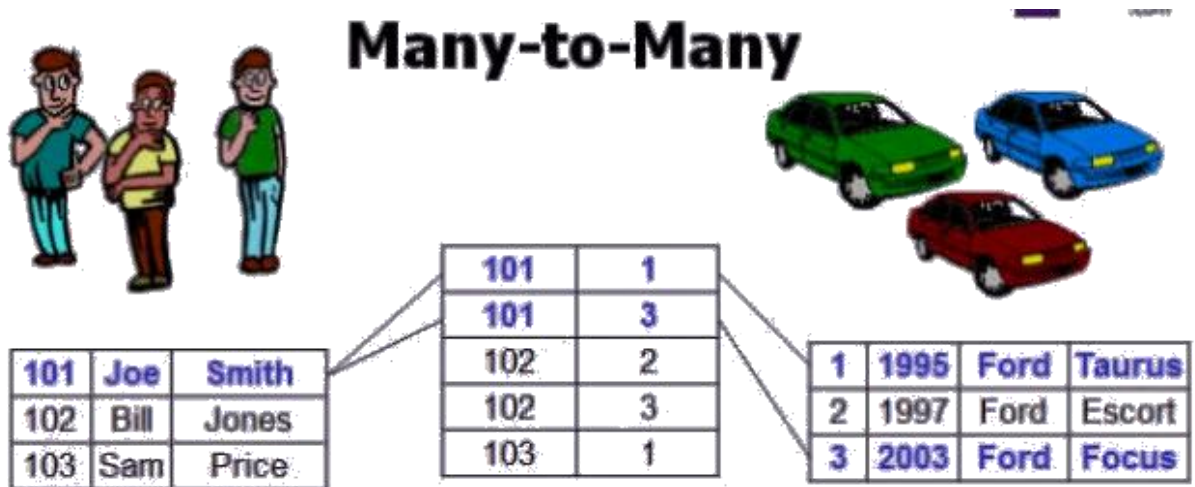


## Many-to-Many Relationships

In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa.

You create such a relationship by defining a third table, called a junction table, whose primary key consists of the foreign keys from both table A and table B.

For example, the Customers table and the Cookies table have a many-to-many relationship that is defined by a one-to-many relationship from each of these tables to the Orders table.





## ***One-to-One Relationships***

In a one-to-one relationship, a row in table A can have no more than one matching row in table B, and vice versa. A one-to-one relationship is created if both the related columns are primary keys or have unique constraints.

This type of relationship is not common because most information related in this way would be all in one table. You might use a one-to-one relationship to –

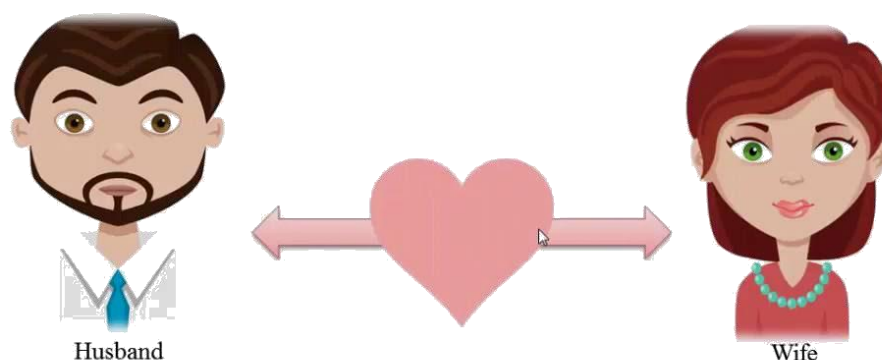
Divide a table into many columns.

Isolate part of a table for security reasons.

Store data that is short-lived and could be easily deleted by simply deleting the table.

Store information that applies only to a subset of the main table.

## **One to One**



[www.geekyshows.com](http://www.geekyshows.com)