Microsoft Access
Understanding Relationships

Er. Rajan Karmcharya

Lecturer

(St. Xavier's College)

What is a *Relational* Database?

- A relational database is a collection of tables from which data can be accessed in many different ways without having to reorganize the database tables.
  - That is, the tables can "talk" to each other. We can link (*relate*) our tables to find:
    - Which doctors are seeing a patient
    - Which students are in which class
    - Which item is selling the most on Friday's

What is a *Relational* Database?

- A relational database allows data structures, storage and retrieval operations, and integrity constraints.
  - Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency

Review of the Basic Design Rules of Relational Databases

- **Unique Field Names**
  - Keep fields unique across tables, and keep them as clear as possible in each table.

- **No Calculated or Derived Fields**
  - Calculations and derivations can be performed in Queries, Forms and Reports. Doing them in a table only increases the chance of data entry error.

Review of the Basic Design Rules of Relational Databases

- **Data is broken down into Smallest Logical Parts**
  - Smallest "Sortable" parts. Remember it's much easier to pull fields together than it is to pull a field apart.

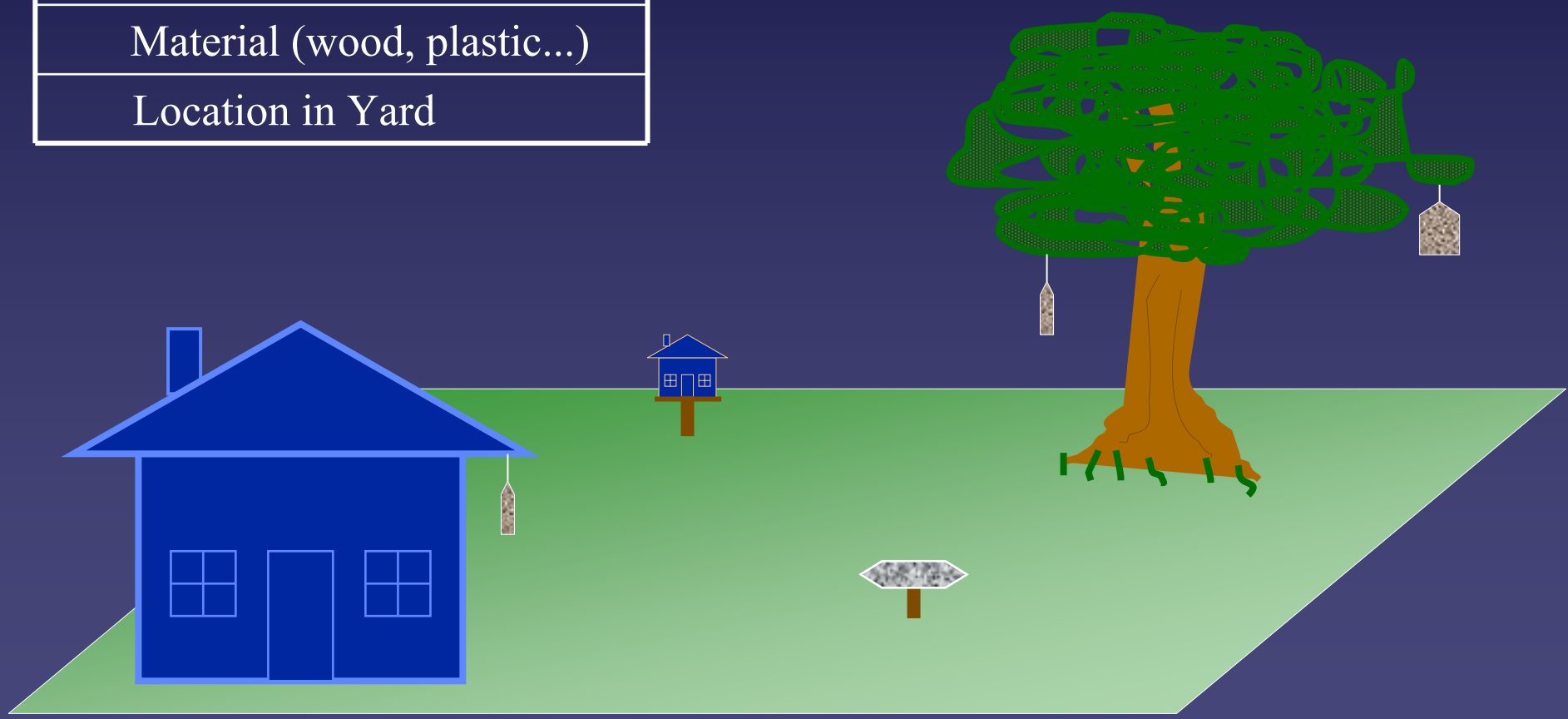
- **Unique Records**
  - Each of your tables should have unique records. We ensure this by setting one field to be a **Primary Key**. This can be a user generated field or an AutoNumber.

Table of Yards

Yards
Yard Number
Owner
Address
Phone Number

Table of Birdfeeders

Bird Feeders
Bird Feeder Number
Material (wood, plastic...)
Location in Yard



Relating Yards and Birdfeeders

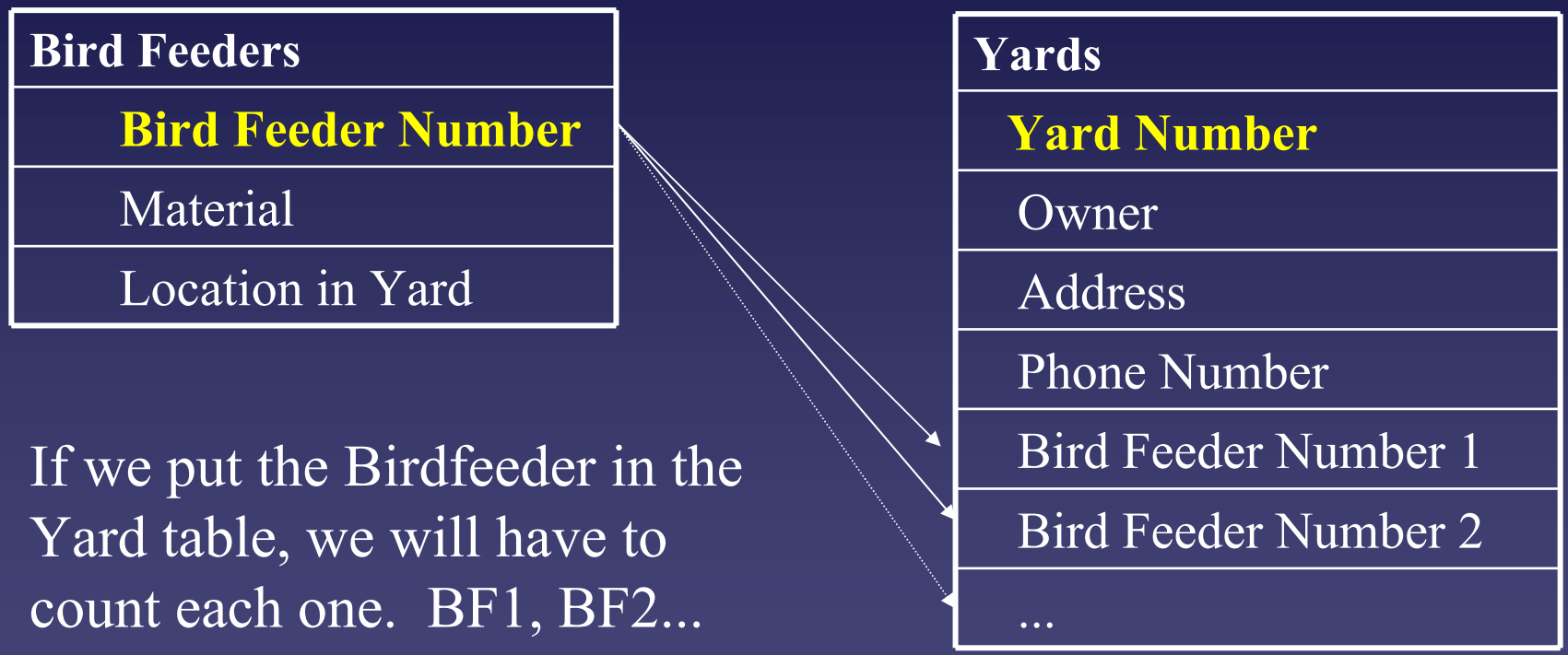
Bird Feeders
Bird Feeder Number
Material
Location in Yard

Yards
Yard Number
Owner
Address
Phone Number

There must be one field in both tables that is the same, so that the database knows how the tables connect.

It's best to use the Primary key as the link.

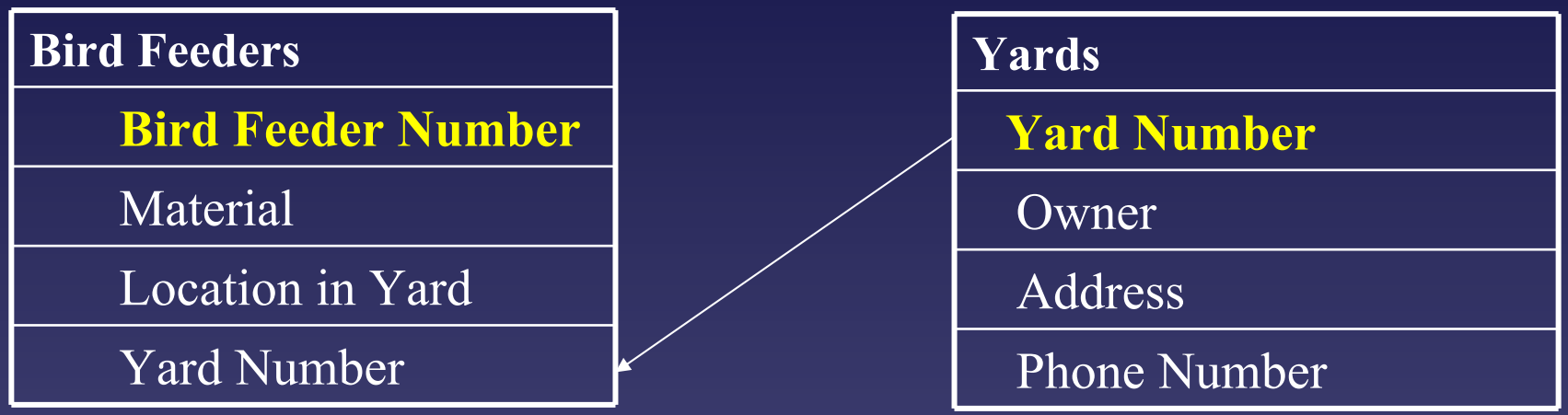
Relating Yards and Birdfeeders



If we put the Birdfeeder in the Yard table, we will have to count each one. BF1, BF2...

Whenever you find yourself numbering the fields in this way, it's a sign you're on the wrong track

Relating Yards and Birdfeeders



But if we reverse the direction, the link (relationship) makes more sense.

Each Birdfeeder can only be in one Yard, but each Yard can have many Birdfeeders. This is called a one to many (1 - ∞) Relationship.

One to Many Relationships

- One to Many relationships are the most common relationships.
  - One Birdfeeder *is visited by* Many Birds
  - One Yard *contains* Many Birdfeeders
  - One Patient *has* Many Prescriptions
  - One Insurance *has* Many Patients
  - One Student *attends* Many Classes
- One to Many includes One to None.
- A record *MUST* be in the One table in order to appear in the Many table.

One to Many Relationships

Primary Key linked to Non Primary Key

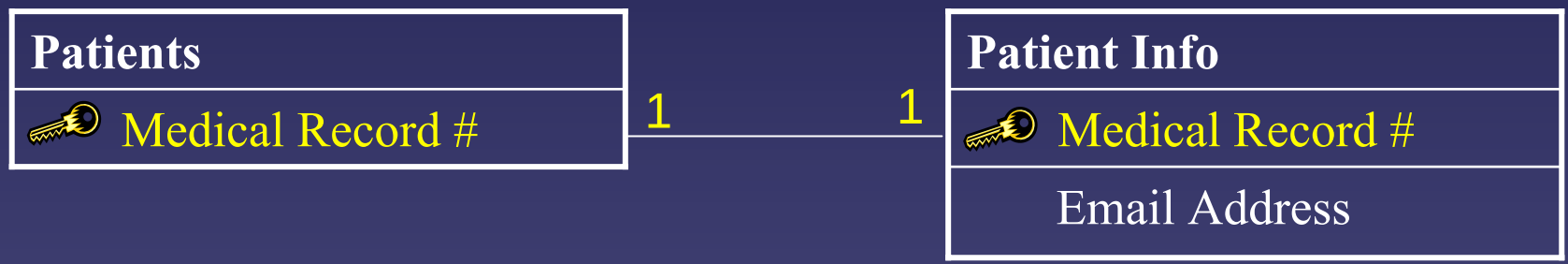
One to One Relationships

One to One relationships can often combine the data into one table.

- One Birdfeeder *is located in* One place in the Yard
- One Yard *has* One Address
- One Patient *has* One Home Phone Number
- One Insurance *has* One Contact Person
- One Student *has* One Gatorlink ID

One to One Relationships

Primary Key linked to Primary Key



One to One Relationships

Reasons you may use a One to One…

- You have more than 255 fields
  - the maximum number of columns (fields)
- You have a large set of related data that doesn't need to be accessed every time you look up that item
  - medical history, map of the location, transcript from previous school

Many to Many Relationships

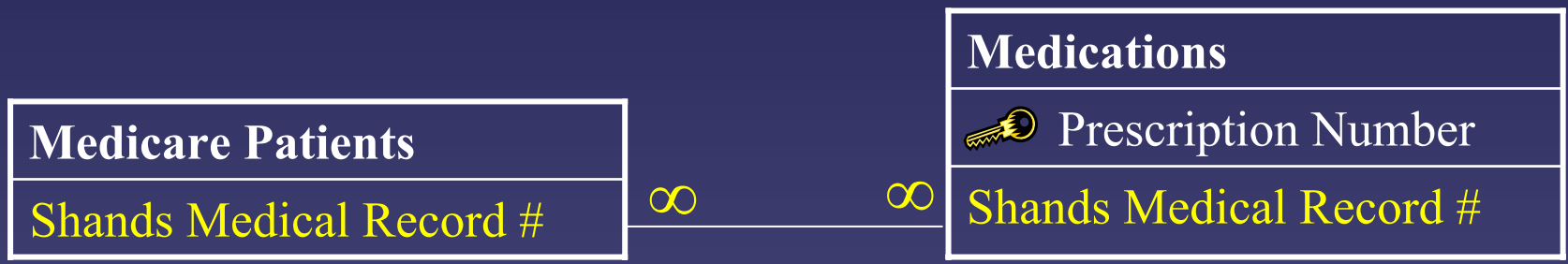
Many to Many relationships are also very common.

- Many Students *are taught by* Many Teachers
- Many Patients *see* Many Doctors
- Many Medications *are taken by* Many Patients
- Many Customers *buy* Many Products

You cannot create a "true" relationship between these tables.

Many to Many Relationships

Non Primary Key linked to Non Primary Key



Access sees this as an Indeterminate relationship

You cannot Enforce Referential Integrity

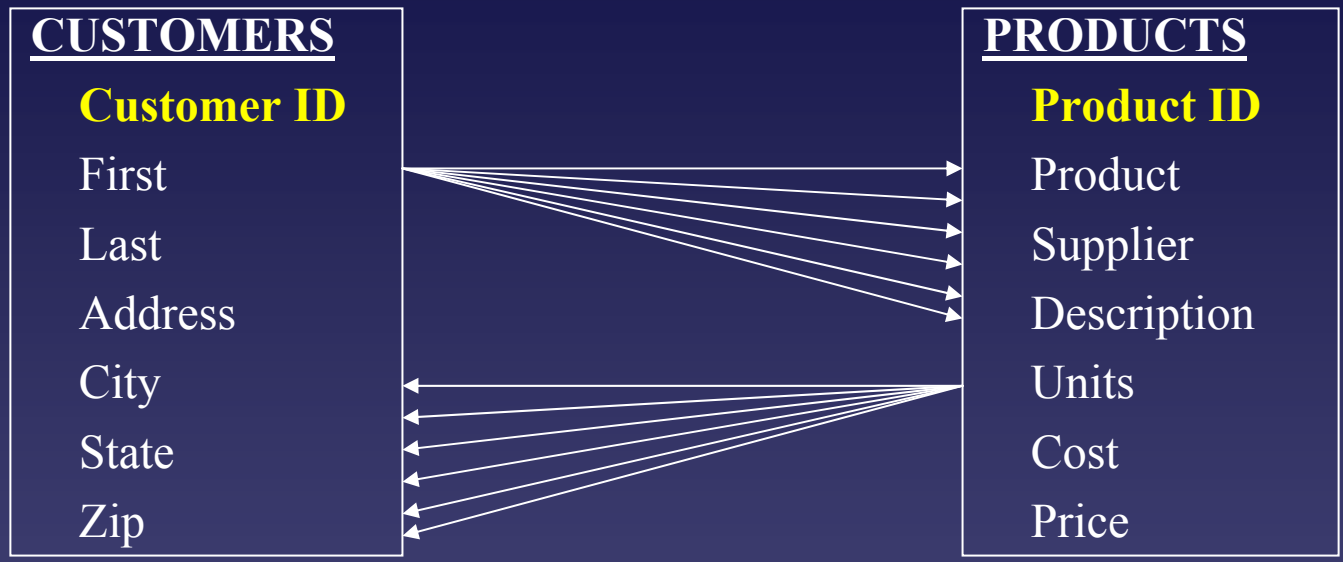
Indeterminate Relationships

Indeterminate relationships are often found when we are linking tables, because Linked Tables cannot have a primary key.

If you have imported or created a table, it's very rare to have a *need* for an indeterminate Relationship.

These relationships show Access that the data saved in the field from the first table is the same kind of data saved in the second table, but there *can be no data integrity rules applied* on indeterminate relationships.

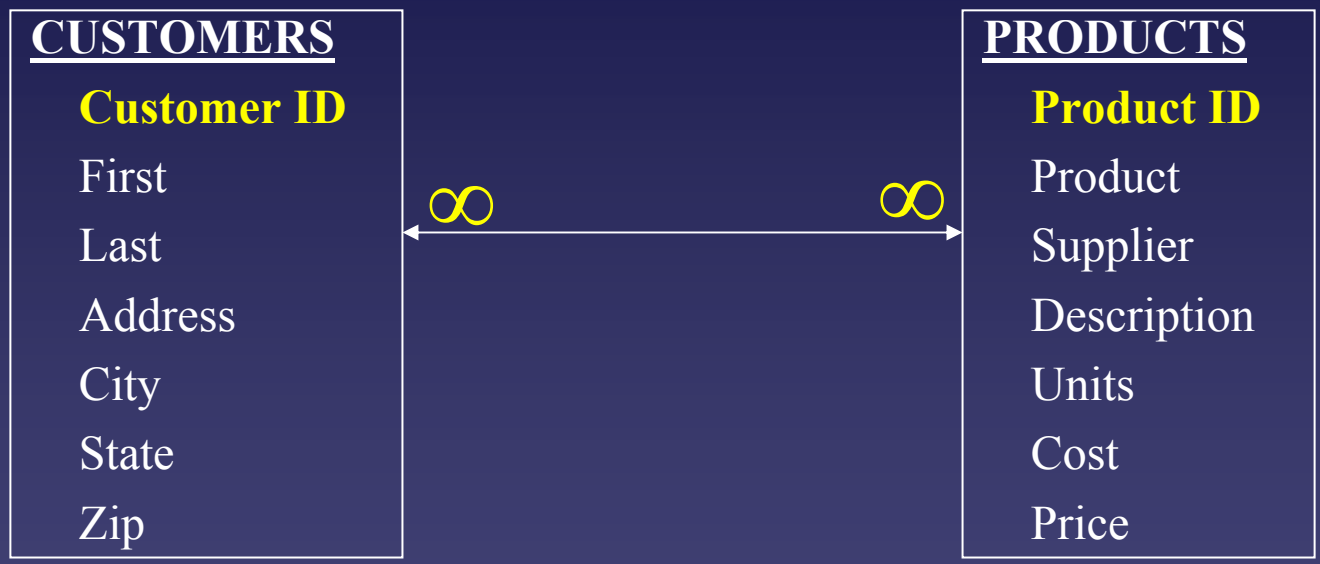
Sales Database



Jack bought:
2 Hats
4 Pairs of Socks
3 Pairs of Shoes

Shoes bought by:
Jack Johnson
Jill Jones
Jerry Jacks

Sales Database



Many to Many Relationship

Sales Database

CUSTOMERS

- **Customer ID**
- First
- Last
- Address
- City
- State
- Zip

PRODUCTS

- **Product ID**
- Product
- Supplier
- Description
- Units
- Cost
- Price
- Customer1
- Customer2
- Customer3
- .....

One Product can be purchased by an unlimited number of Customers.

Sales Database

CUSTOMERS

- **Customer ID**
- First
- Last
- Address
- City
- State
- Zip
- Product1
- Product2
- Product3
- .....

PRODUCTS

- **Product ID**
- Product
- Supplier
- Description
- Units
- Cost
- Price

One Customer can purchase an unlimited number of Products.

Sales Database

Main Table

CUSTOMERS

- **Customer ID**
- First
- Last
- Address
- City
- State
- Zip

Junction Table

SALES

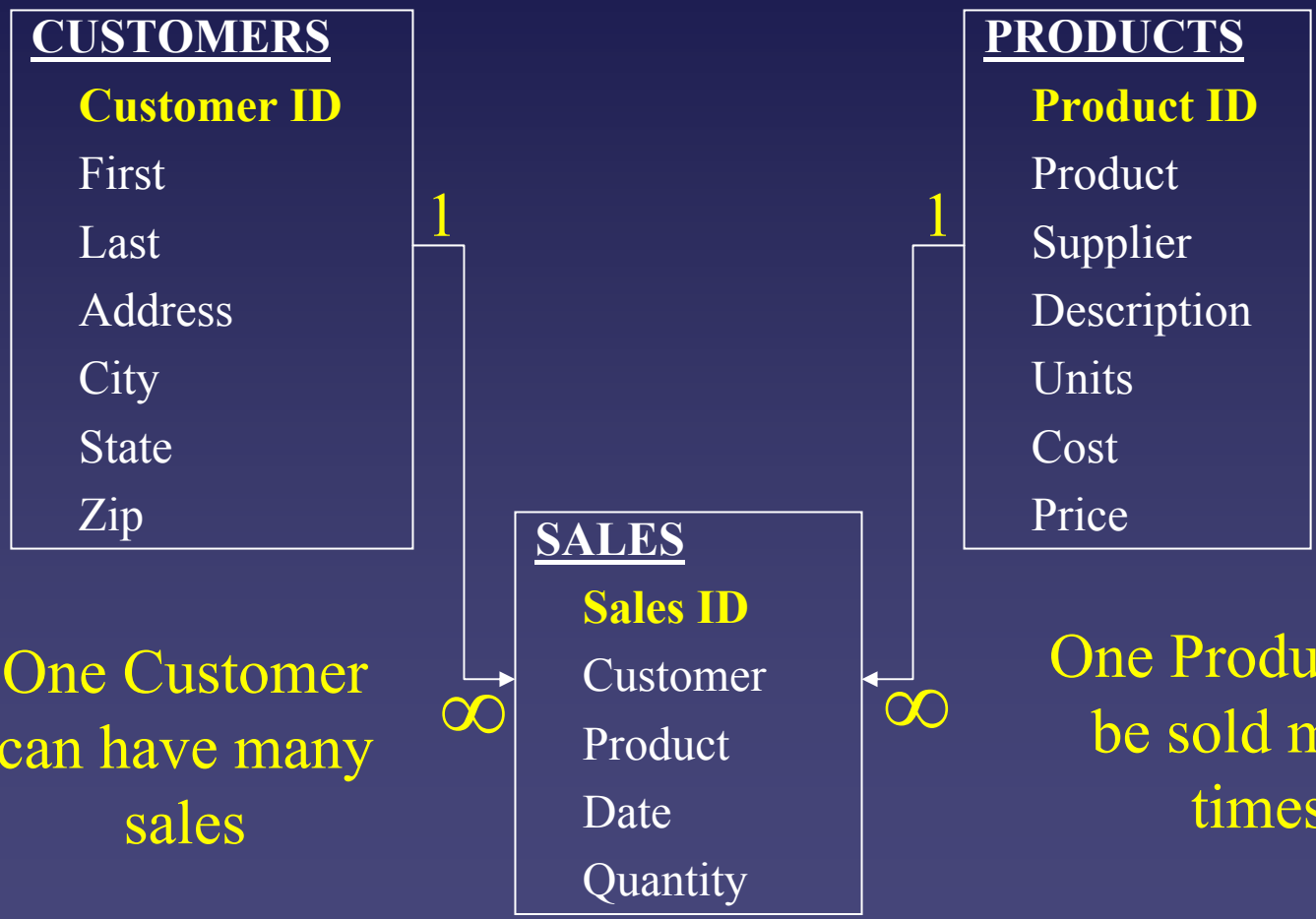
- **Sales ID**
- Customer ID
- Product ID
- Date
- Quantity

Main Table

PRODUCTS

- **Product ID**
- Product
- Supplier
- Description
- Units
- Cost
- Price

Sales Database



Sales Database

Products by Customer

First _____ Last _____

Address _____

City _____ State ____ Zip Code _____

Product	Date	Qty

Sales Database

Customers by Products

Product _____

Supplier _____

Description _____

Units _____ Cost _____ Price _____

Customer	Date	Qty

Examples

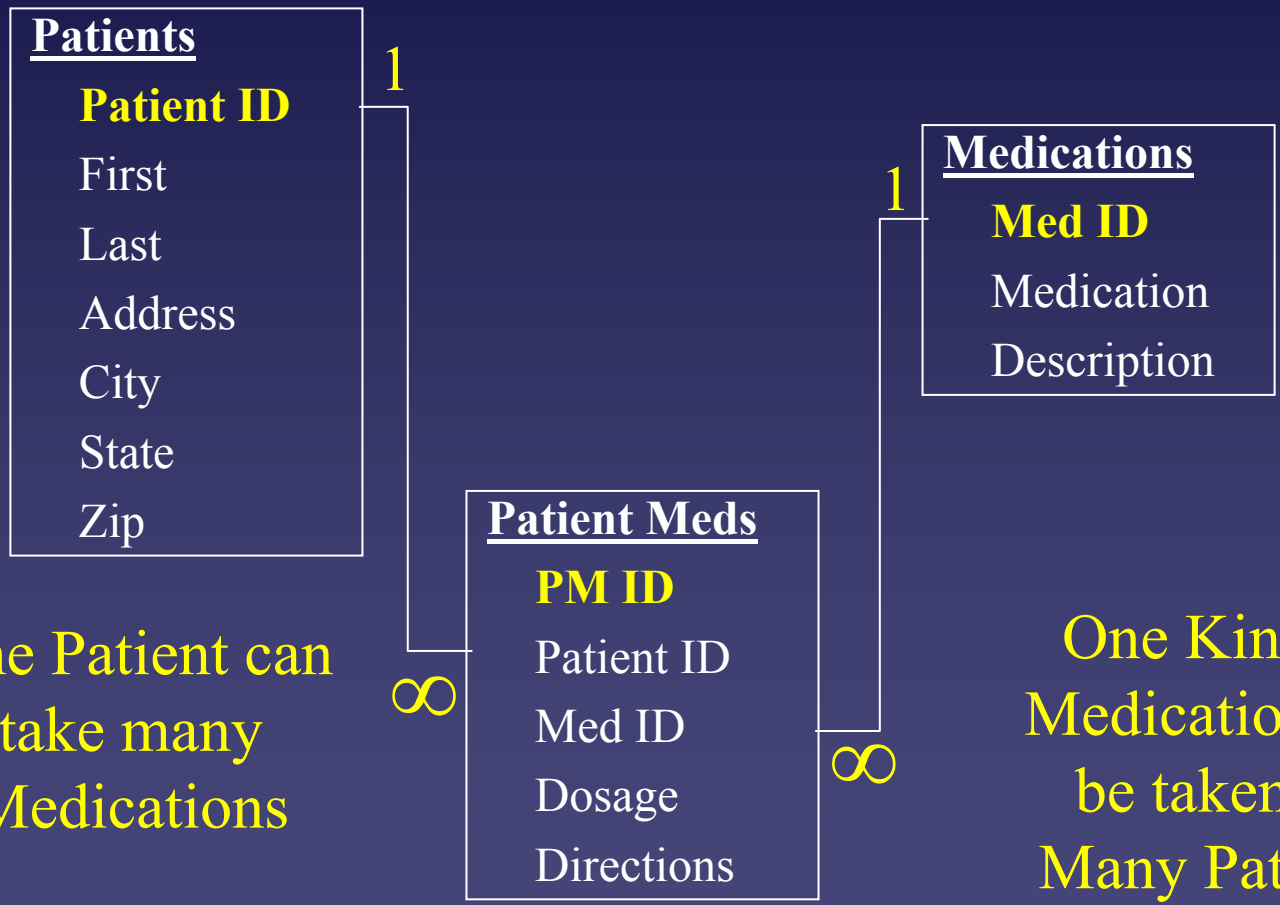
Patients

- **Patient ID**
- First
- Last
- Address
- City
- State
- Zip

Medications

- **Med ID**
- Medication
- Description

Examples


Patients
Patient ID
First
Last
Address
City
State
Zip
1
Medications
Med ID
Medication
Description
1
Patient Meds
PM ID
Patient ID
Med ID
Dosage
Directions
∞
∞
One Patient can take many Medications
One Kind of Medication can be taken by Many Patients

Examples

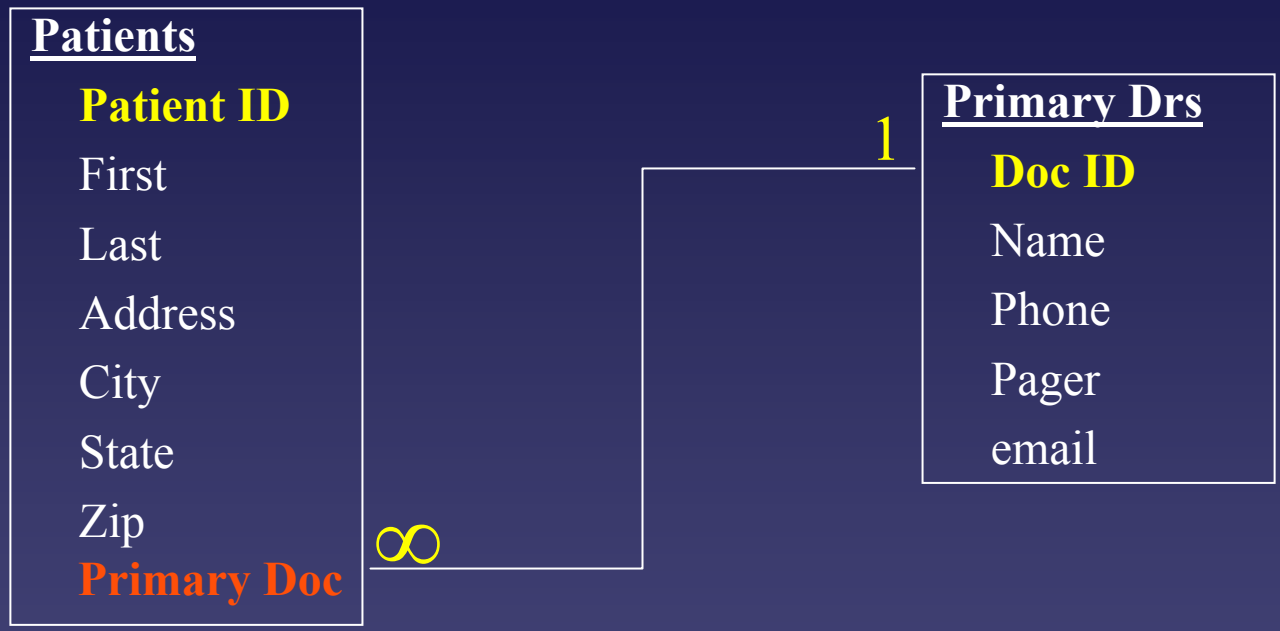
Patients

- **Patient ID**
- First
- Last
- Address
- City
- State
- Zip

Primary Drs

- **Doc ID**
- Name
- Phone
- Pager
- email

Examples



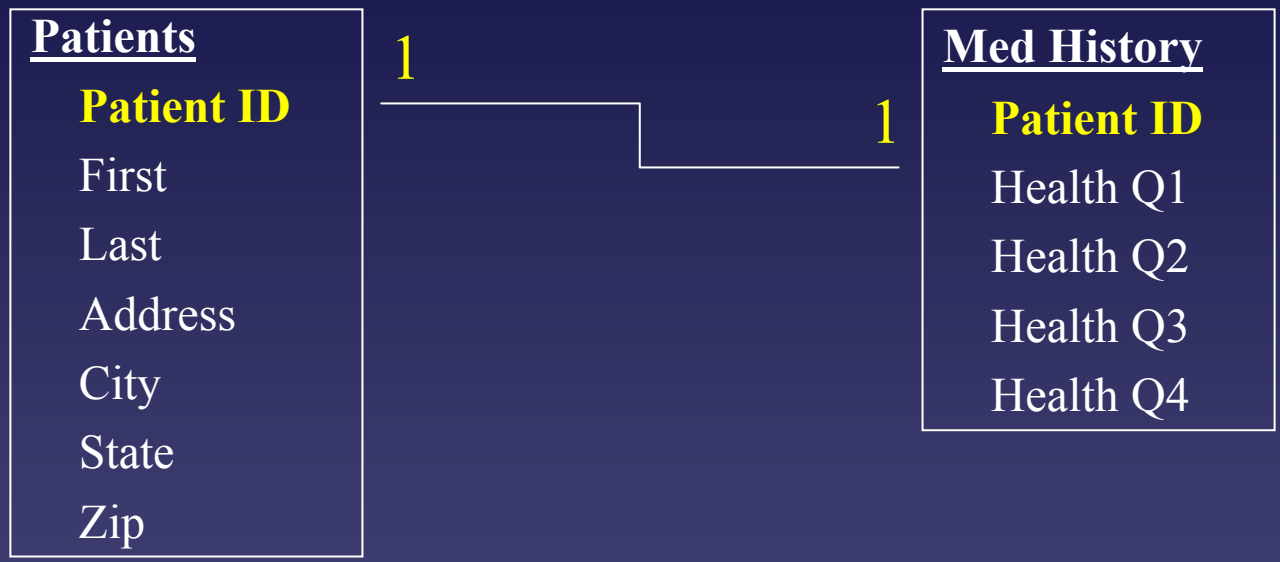
One Patient will have only ONE primary Doctor.
One Primary Doctor can have MANY patients.

Examples

Patients
Patient ID
First
Last
Address
City
State
Zip

Med History
Patient ID
Health Q1
Health Q2
Health Q3
Health Q4

Examples



One Patient will have only ONE Medical History.
Each Medical History will belong to only ONE patient.

Let's Practice…