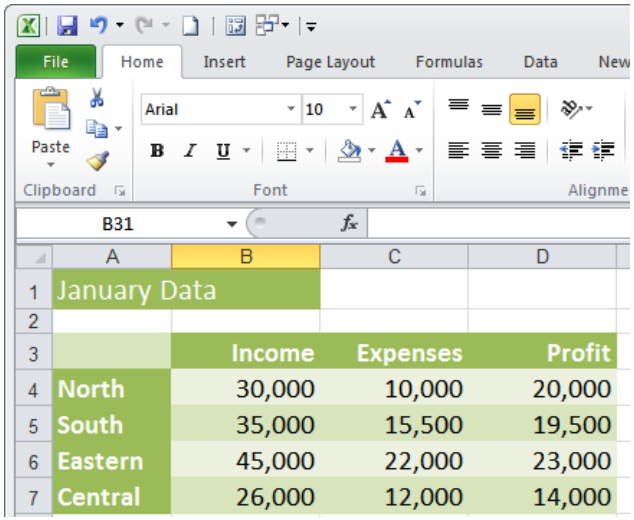
Introduction to Relational Databases, Data Modeling and Database Management

ALY 6030- Week2

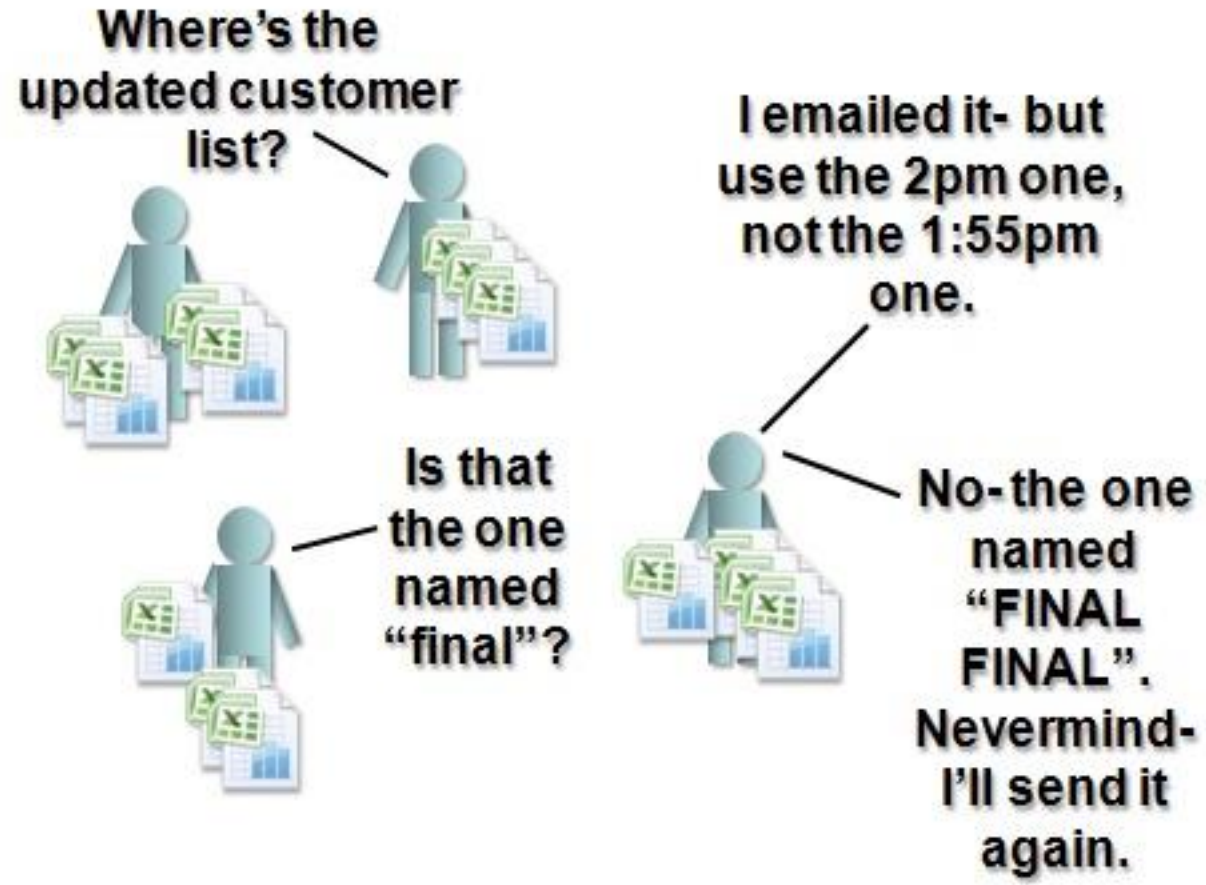
Dr. Behboudi –Northeastern University

Past → Present → Future



	A	B	C	D
1	January Data			
2				
3		Income	Expenses	Profit
4	North	30,000	10,000	20,000
5	South	35,000	15,500	19,500
6	Eastern	45,000	22,000	23,000
7	Central	26,000	12,000	14,000

Have you experienced this?



http://www.datamartist.com/product/team-collaboration-with-datamartist

Motivating questions

- Why use relational databases to store data?
- What is the best way to store our data within a relational database?

Example: Excel spreadsheet

- Consider the spreadsheet, Buildings.xlsx:

ID	Nickname	FullName	StreetNumber	StreetName
368	Kendall	Red Kendall Building	100	Main St.
379	Central	Green Central Building	200	Main St.
346	Porter	Purple Porter Building	103	Center Ave.
352	Davis	Blue Davis Building	105	Center Ave.
489	Alewife	Orange Alewife Building	569	Broadway Ave.
412	Park	Yellow Park Building	613	Broadway Ave.

- This sheet stores information about the ***concept*** of a building within some kind of company
- This would be a ***table*** in a relational database

Relational model definitions

- **Entity**: object, concept or event
- **Attribute** (column): a characteristic of an entity
- **Record** (row): the specific characteristics or attribute values for one example of an entity
- **Entry**: the value of an attribute for a specific record
- **Table**: a collection of records
- **Database**: a collection of tables

Single table example

Entity: Buildings

Table: A collection of records about the entity (buildings)

Buildings

Record: Information about building 346

ID	Nickname	FullName	StreetNumber	StreetName
368	Kendall	Red Kendall Building	100	Main St.
379	Central	Green Central Building	200	Main St.
346	Porter	Purple Porter Building	103	Center Ave.
352	Davis	Blue Davis Building	105	Center Ave.
489	Alewife	Orange Alewife Building	569	Broadway Ave.
412	Park	Yellow Park Building	613	Broadway Ave.

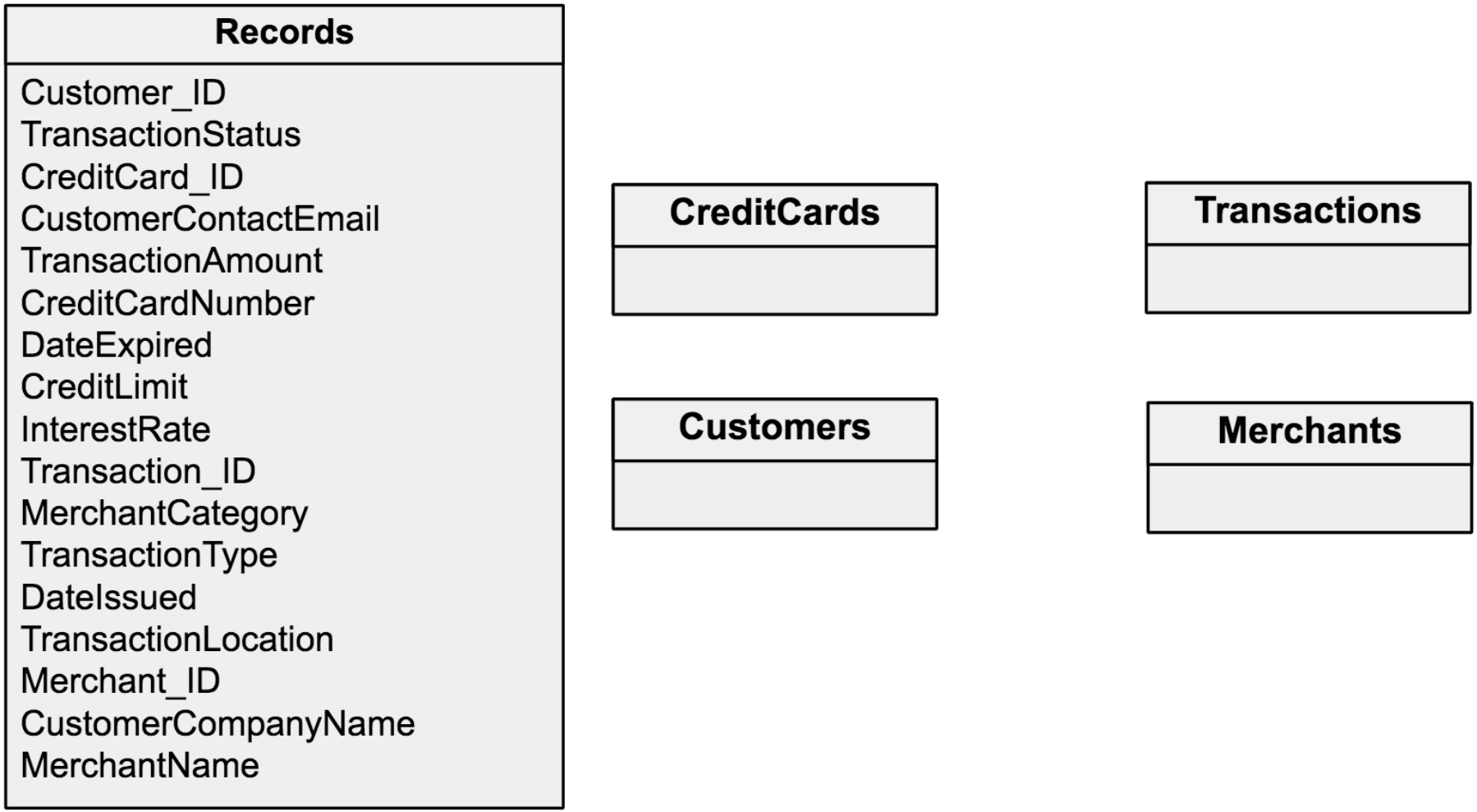
Entry: Value of ID for the Park building

Attribute: FullName– the full names of the buildings

Database: dataMedical, includes tables such as: Employees, Buildings, Departments, Patients

Check your knowledge
Credit card database

What are the entities in this credit card database?



Check your knowledge
Medical database

Identify which are entities and which are attributes:

- Appointments
- Patient First Name
- Employee Email Address
- Patient Number
- Prescriptions
- Street Name
- Type of Department
- Department Floor
- Building Number
- Medical Notes
- Drug Count
- Healthcare Plans
- Prescription Number
- Employee Phone Extension
- Department Number
- Appointment Cost
- Refills
- Drug Strength Unit
- Hospital Center ID
- Patient Last Name
- Department Name
- Town
- Department Status
- State
- Building Name
- Street Number
- Appointment Room
- Employees
- Out Of Pocket To Date
- Job Title
- Drug Cost
- Patients
- Drug Copay
- Building Nickname
- HSA eligible
- Department
- Appointment Number
- Text of notes
- Employee First Name
- Employee ID
- Center Name
- Employee Last Name
- Zip Code
- Healthcare Plan Number
- Healthcare Plan Tier
- Note Number
- Employee Status
- Center Nickname
- Hospital Centers
- Buildings
- Max Out Of Pocket
- Appointment Length
- Drug Strength
- Appointment Copay
- Appointment Date and Time
- Drug Name
- Drug Unit

Check your knowledge
Medical database

Identify which are entities and which are attributes:

- Appointments (E)
- Patient First Name
- Employee Email Address
- Patient Number
- Prescriptions (E)
- Street Name
- Type of Department
- Department Floor
- Building Number
- Medical Notes (E)
- Drug Count
- Healthcare Plans (E)
- Prescription Number
- Employee Phone Extension
- Department Number
- Appointment Cost
- Refills
- Drug Strength Unit
- Hospital Center ID
- Patient Last Name
- Department Name
- Town
- Department Status
- State
- Building Name
- Street Number
- Appointment Room
- Employees (E)
- Out Of Pocket To Date
- Job Title
- Drug Cost
- Patients (E)
- Drug Copay
- Building Nickname
- HSA eligible
- Departments (E)
- Appointment Number
- Text of notes
- Employee First Name
- Employee ID
- Center Name
- Employee Last Name
- Zip Code
- Healthcare Plan Number
- Healthcare Plan Tier
- Note Number
- Employee Status
- Center Nickname
- Hospital Centers (E)
- Buildings (E)
- Max Out Of Pocket
- Appointment Length
- Drug Strength
- Appointment Copay
- Appointment Date and Time
- Drug Name
- Drug Unit

Client-Server Architecture

- Client is for the user
  - User (analyst or database administrator) opens an application to run some queries or interact with the database
- Server is managed by the administrator
  - Stores the data
  - Manages connections to the data
    - Multiple clients can log into server and access data, server determines level of access
- Abstraction, performance, consistency

Why do we need data models to design a database?

- Data models help specify each entity in a table **in a standardized way**
- Data models allow administrator to impose rules, constraints, and **relationships** on the data that are stored
  - Enables users to understand business rules and effectively process and analyze data
- Acts as a **schematic** for building the database

Rules of the relational data model

- Each **attribute** (column) has a unique name within a **table**
- All **entries** or **values** in the **attribute** are examples of that **attribute**
- Each **record** (row) is unique in a good database

Buildings

ID	Nickname	FullName	StreetNumber	StreetName
368	Kendall	Red Kendall Building	100	Main St.
379	Central	Green Central Building	200	Main St.
346	Porter	Purple Porter Building	103	Center Ave.
352	Davis	Blue Davis Building	105	Center Ave.
489	Alewife	Orange Alewife Building	569	Broadway Ave.
412	Park	Yellow Park Building	613	Broadway Ave.

What makes a good data model?

- **Complete**: Is all necessary data represented?
- **No redundancy**: Is the same fact recorded more than once?
- **Enforcement of rules**: How accurately does it enforce business rules?

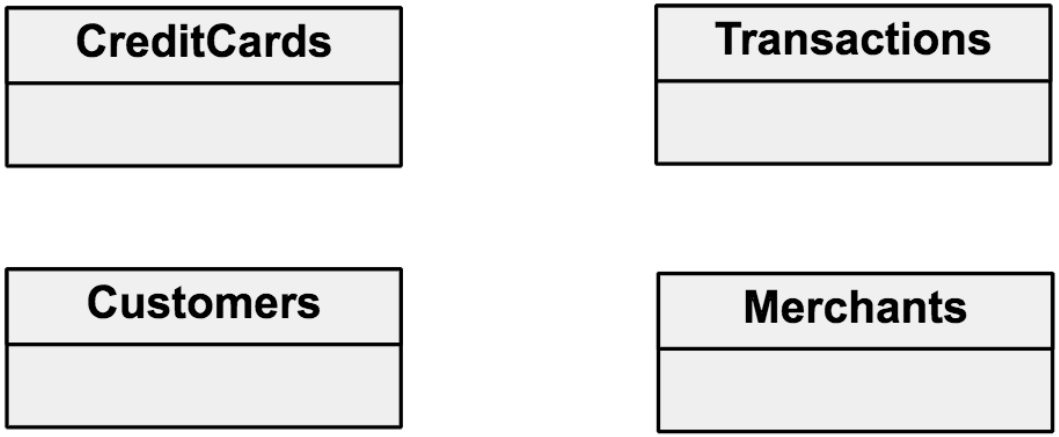
How to draw an entity-relationship diagram (ERD)

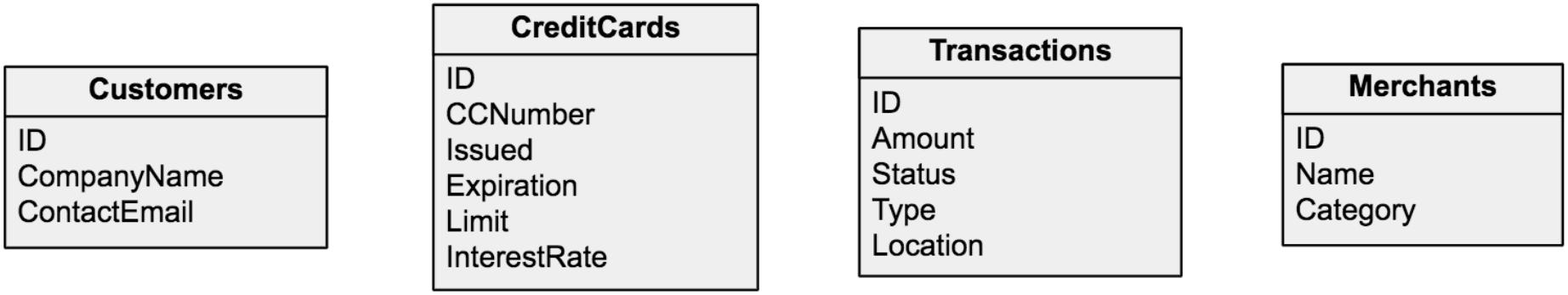
- **ERD** or **entity-relationship diagram** is a schematic of the database
- **Entities** are drawn as boxes
- **Relationships** between entities are indicated by lines between these entities
- **Cardinality** describes the expected number related occurrences between the two entities in a relationship and is shown using **crow's foot notation**

Relationships + cardinality = business rules

Check your knowledge
Credit card database

Which attributes are associated with each entity?

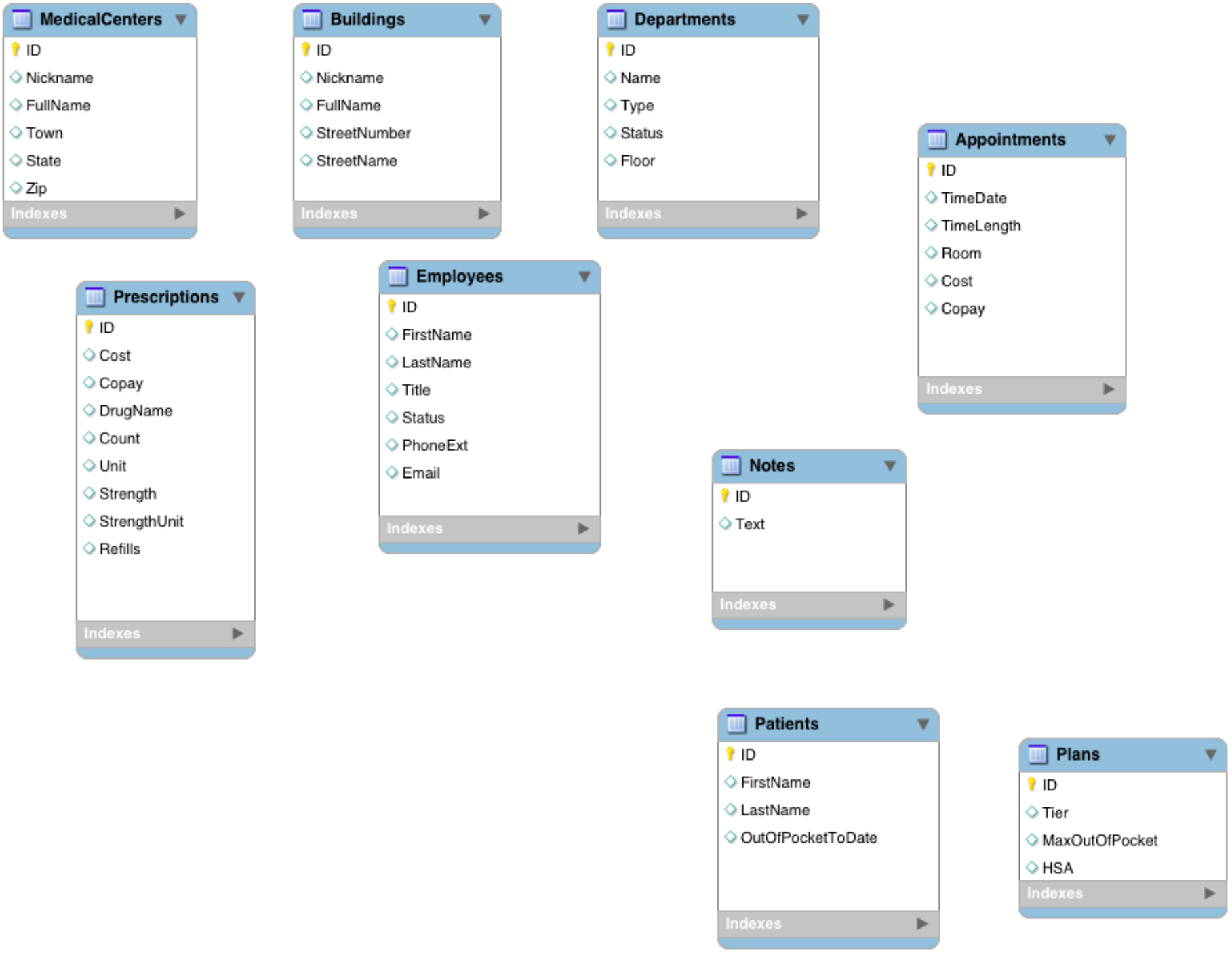




Check your knowledge

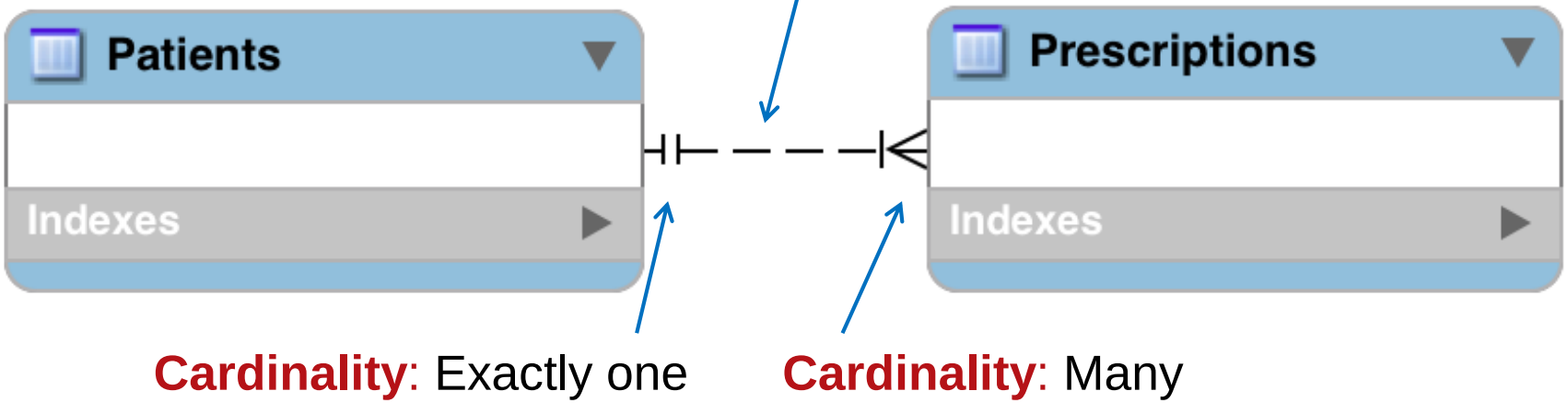
Identify which attributes belong with which entity

- Appointments (E)
- Patient First Name
- Employee Email Address
- Patient Number
- Prescriptions (E)
- Street Name
- Type of Department
- Department Floor
- Building Number
- Medical Notes (E)
- Drug Count
- Healthcare Plans (E)
- Prescription Number
- Employee Phone Extension
- Department Number
- Appointment Cost
- Refills
- Drug Strength Unit
- Hospital Center ID
- Patient Last Name
- Department Name
- Town
- Department Status
- State
- Building Name
- Street Number
- Appointment Room
- Employees (E)
- Out Of Pocket To Date
- Job Title
- Drug Cost
- Patients (E)
- Drug Copay
- Building Nickname
- HSA eligible
- Departments (E)
- Appointment Number
- Text of notes
- Employee First Name
- Employee ID
- Center Name
- Employee Last Name
- Zip Code
- Healthcare Plan Number
- Healthcare Plan Tier
- Note Number
- Employee Status
- Center Nickname
- Hospital Centers (E)
- Buildings (E)
- Max Out Of Pocket
- Appointment Length
- Drug Strength
- Appointment Copay
- Appointment Date and Time
- Drug Name
- Drug Unit


MedicalCenters
ID
Nickname
FullName
Town
State
Zip
Indexes
Buildings
ID
Nickname
FullName
StreetNumber
StreetName
Indexes
Departments
ID
Name
Type
Status
Floor
Indexes
Appointments
ID
TimeDate
TimeLength
Room
Cost
Copay
Indexes
Prescriptions
ID
Cost
Copay
DrugName
Count
Unit
Strength
StrengthUnit
Refills
Indexes
Employees
ID
FirstName
LastName
Title
Status
PhoneExt
Email
Indexes
Notes
ID
Text
Indexes
Patients
ID
FirstName
LastName
OutOfPocketToDate
Indexes
Plans
ID
Tier
MaxOutOfPocket
HSA
Indexes

ERD for Patients and Prescriptions

Relationship: There is a relationship between Patients and Prescriptions



Cardinality: Exactly one

Cardinality: Many

- **Business rules** defined through **relationships** and **cardinality**:
  - There is **exactly one** patient for each prescription
  - Each patient may have zero, one or many prescriptions (shortened to **zero or many**)

Cardinality – crow's foot notation

- General meanings:

One

Many

- For reference in formal ERDs, mandatory vs. optional:

One (and only one)

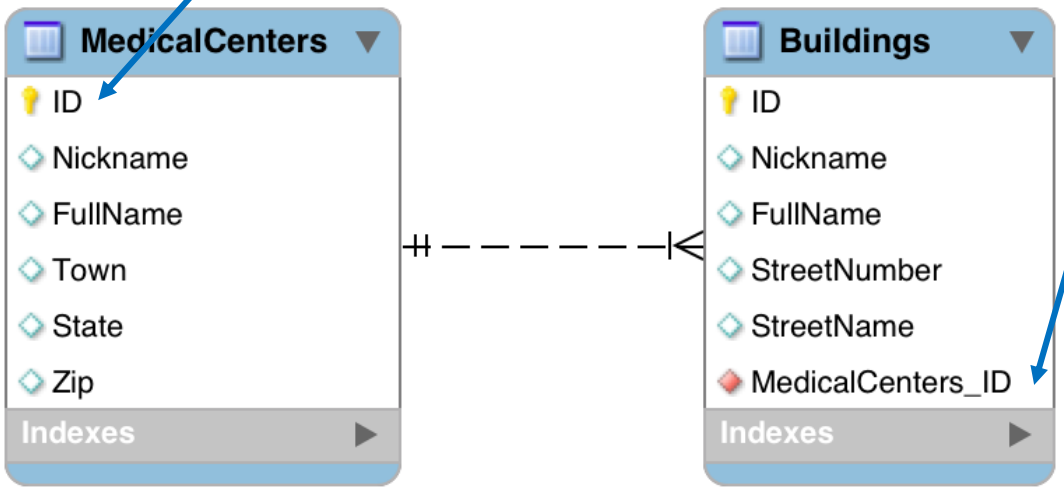
Zero or one

One or many

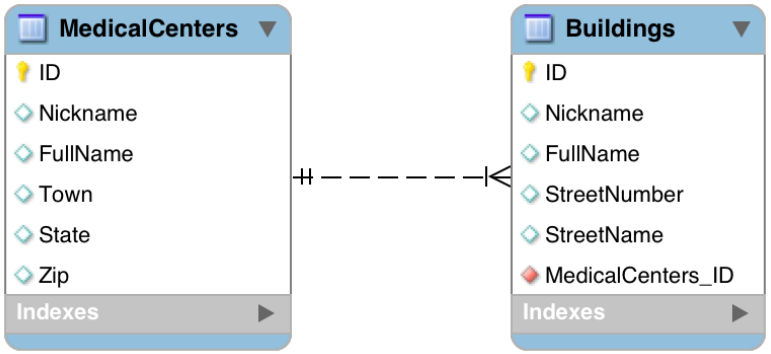
Zero or many

Primary and foreign keys

- **Primary key**: one or more attributes that uniquely identify a record – Buildings.ID and MedicalCenters.ID
- Primary key of the **independent** or parent entity type is maintained as a non-key attribute in the **related**, **dependent** or child entity type, this is known as the **foreign key**



Foreign keys



MedicalCenters

ID	Nickname
10	River North
14	River South

Buildings

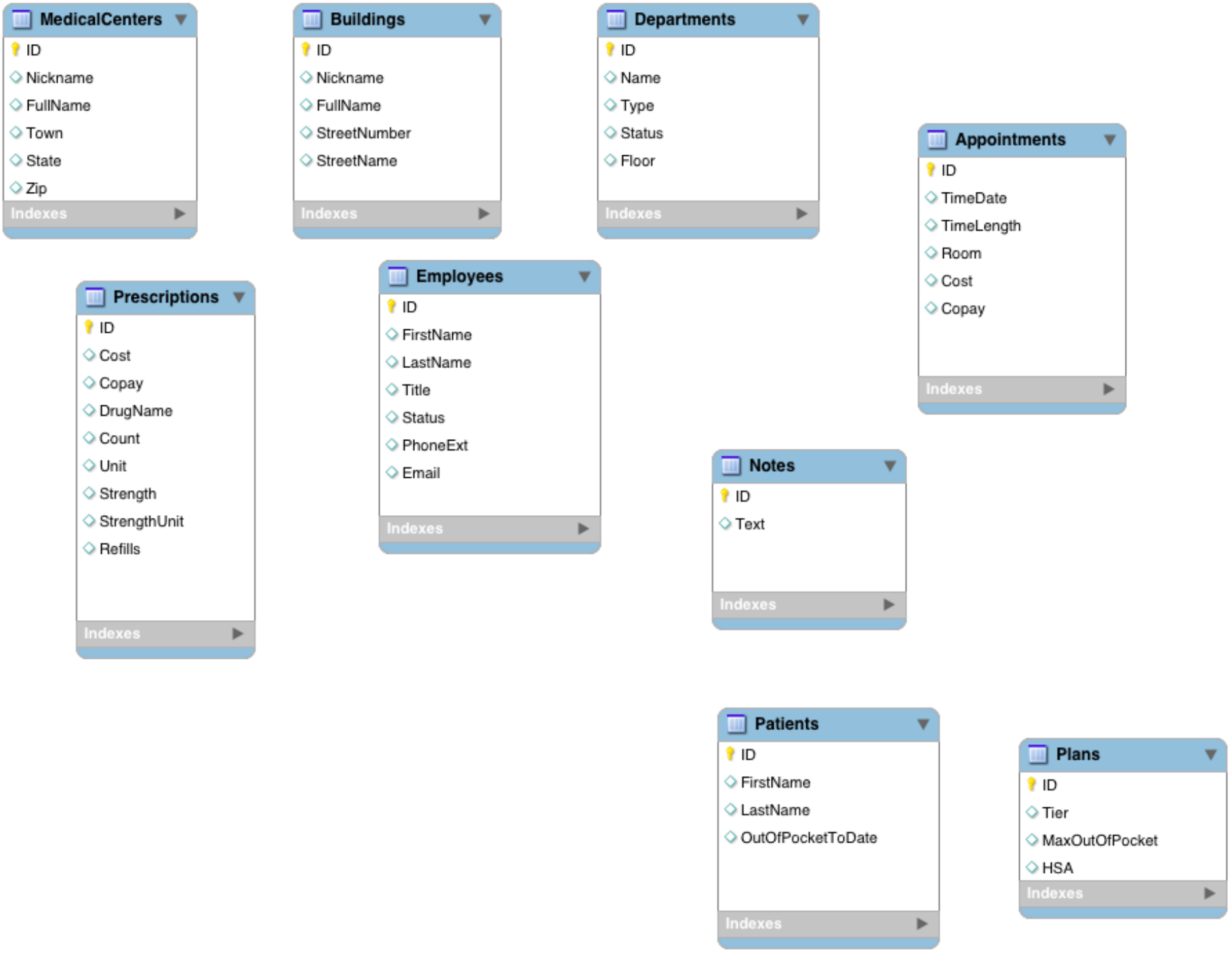
ID	FullName	MedicalCenters_ID
368	Red Kendall Building	10
379	Green Central Building	10
346	Purple Porter Building	10
352	Blue Davis Building	10
489	Orange Alewife Building	14
412	Yellow Park Building	14

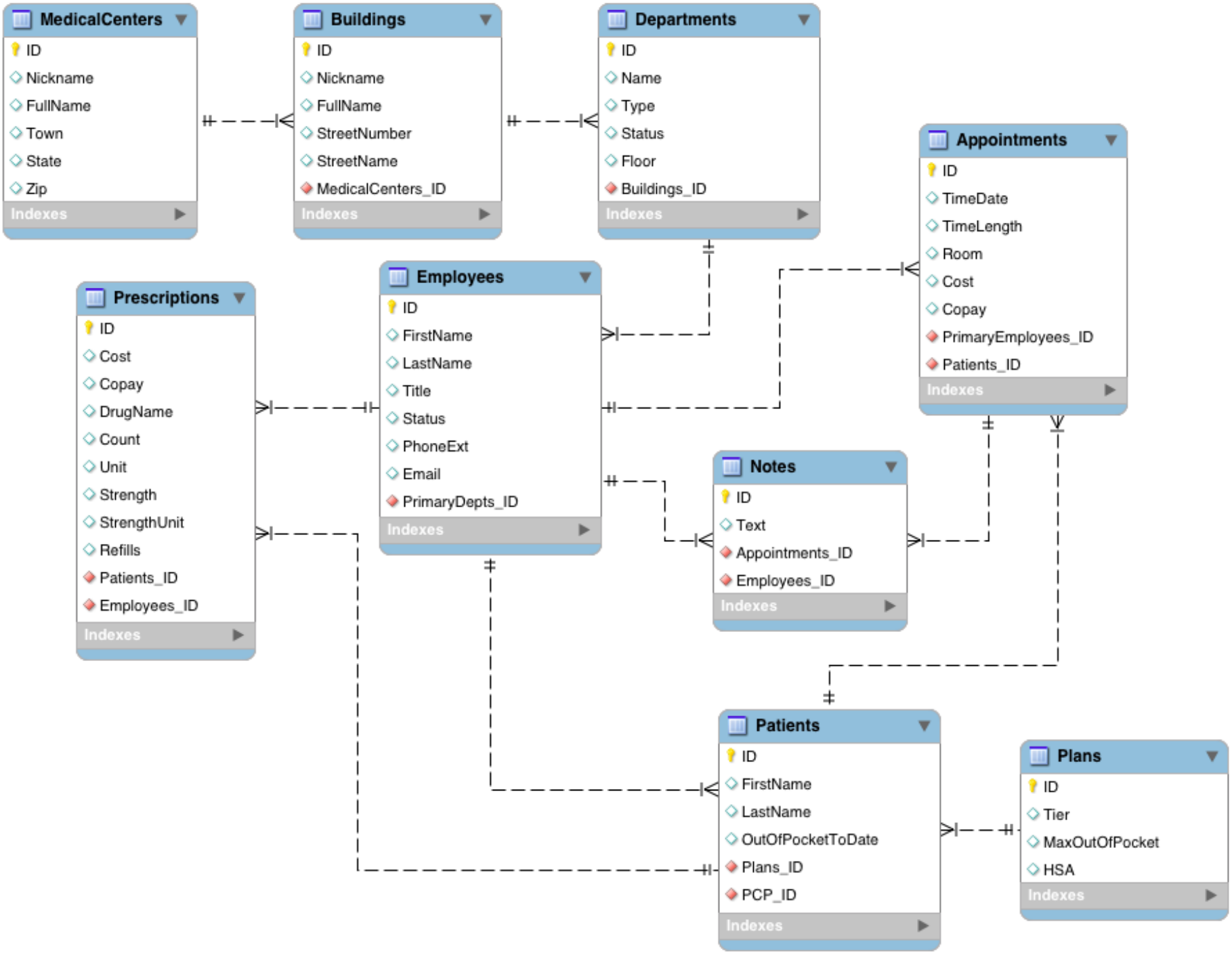
- Database requires a valid Medical Center ID when Building is added.
- ID is the unique identifier of Buildings; MedicalCenters_ID is not needed as part of the Buildings primary key

dataMedical cardinality and foreign keys

- On the next slide there is a data model for the dataMedical group

- Draw relationships, indicating cardinality – and select the appropriate foreign keys to:
  - Captures underlying **rules** or **logic** of the business
  - Provides information about how the database should be structured


MedicalCenters
ID
Nickname
FullName
Town
State
Zip
Indexes
Buildings
ID
Nickname
FullName
StreetNumber
StreetName
Indexes
Departments
ID
Name
Type
Status
Floor
Indexes
Appointments
ID
TimeDate
TimeLength
Room
Cost
Copay
Indexes
Prescriptions
ID
Cost
Copay
DrugName
Count
Unit
Strength
StrengthUnit
Refills
Indexes
Employees
ID
FirstName
LastName
Title
Status
PhoneExt
Email
Indexes
Notes
ID
Text
Indexes
Patients
ID
FirstName
LastName
OutOfPocketToDate
Indexes
Plans
ID
Tier
MaxOutOfPocket
HSA
Indexes


MedicalCenters
ID
Nickname
FullName
Town
State
Zip
Indexes
Buildings
ID
Nickname
FullName
StreetNumber
StreetName
MedicalCenters_ID
Indexes
Departments
ID
Name
Type
Status
Floor
Buildings_ID
Indexes
Appointments
ID
TimeDate
TimeLength
Room
Cost
Copay
PrimaryEmployees_ID
Patients_ID
Indexes
Prescriptions
ID
Cost
Copay
DrugName
Count
Unit
Strength
StrengthUnit
Refills
Patients_ID
Employees_ID
Indexes
Employees
ID
FirstName
LastName
Title
Status
PhoneExt
Email
PrimaryDepts_ID
Indexes
Notes
ID
Text
Appointments_ID
Employees_ID
Indexes
Patients
ID
FirstName
LastName
OutOfPocketToDate
Plans_ID
PCP_ID
Indexes
Plans
ID
Tier
MaxOutOfPocket
HSA
Indexes

Domain validation entities

- Also called **pick lists** or **validation lists**
- Used to standardize data in a database

Employees			
ID	**FirstName**	**LastName**	**Status**
94165	Linda	Marshall	Current
94312	Timothy	Brown	Current
94323	Diana	West	Current
94122	Courtney	Ford	Current
94324	Dale	Thompson	Previous

Domain validation entity

ValidEmpStatus
Current
Previous

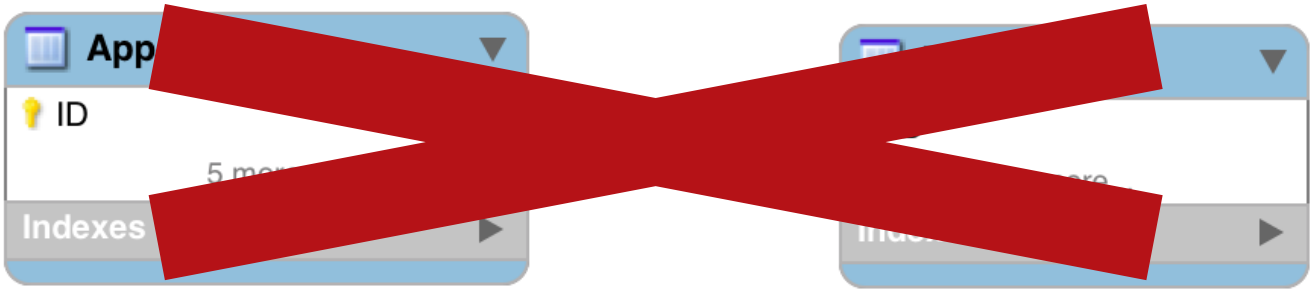
- **Domain validation entity**: table with a single attribute, enforces values of attribute in related table
- Requires that any employee status type must be on a list of existing valid employee statuses in the table "ValidEmpStatus"

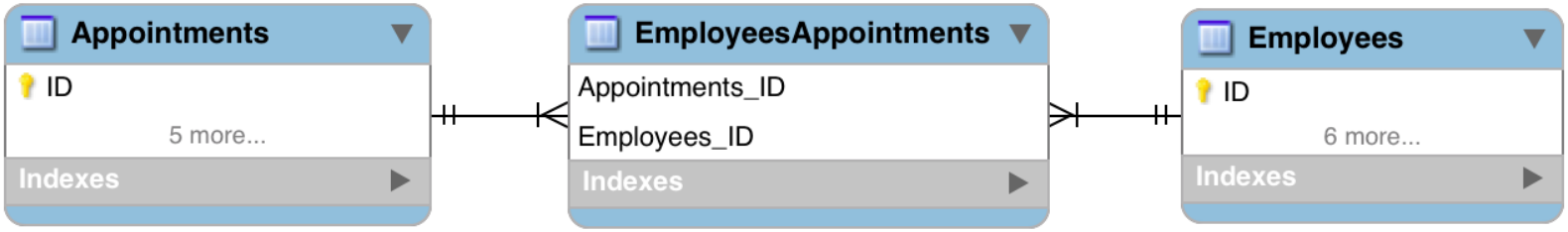
Is there always only one solution for a data model?

- Several solutions may exist
- These often depend on the application requirements or business needs

Many to many relationships

- What if many employees can be associated with each appointment
- How can we get information a given appointment from the database?





Independent Dependent Independent

- **Associative table** (entity), aka **junction table**
- Primary key of **parent** is used in foreign and primary key of **child**

SQL (Structured Query Language) Overview

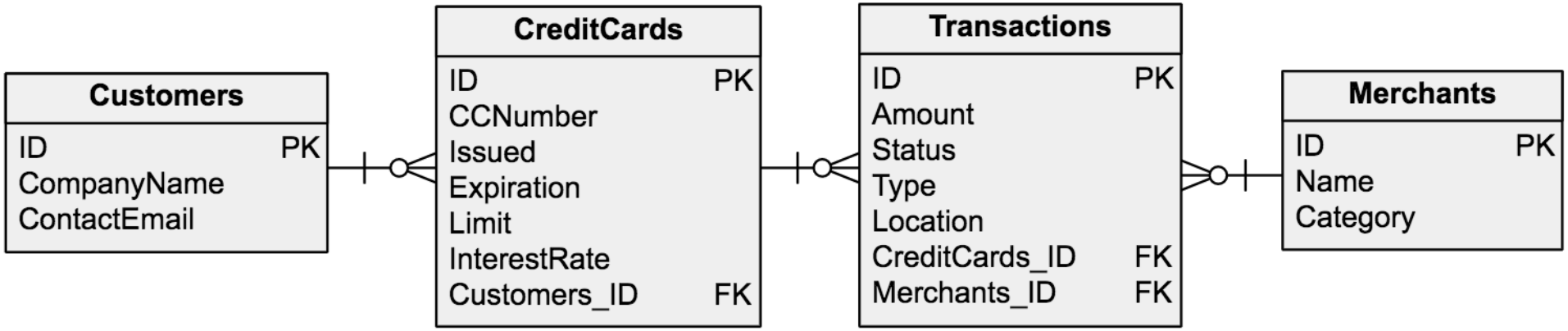
- Not a complete language like Java or C++
  - SQL is sub-language of about 30 statements
  - SQL has several inconsistencies; NULLs are problematic

- Variations among SQL distributions
  - Error codes
  - Data types supported
  - Joins
  - Syntax

Database creation workflow

1. Create the data model.
2. Create a new database using the selected database software.
3. Create the tables within the new database.
4. Insert data into the database.
5. Write queries to retrieve a subset of data from the database.

Creating the credit card database



- Step 1 is complete
- Step 2: Create a new database using the selected database software. (name the database with your username first, if using the Level Server)

```
DROP DATABASE IF EXISTS username_creditcardco;
CREATE DATABASE username_creditcardco;
USE username_creditcard;
```

Numeric data types in SQL

Numeric Data Types	Description
INT	4-Byte integer
BIGINT	8-Byte integer
DECIMAL	Number with fixed number of digits before and after the decimal point e.g. DECIMAL(7,2): -99999.99 to 99999.99
FLOAT	Single-precision floating-point number
DOUBLE	Double-precision floating-point number
BIT	Bit value storage

Reference: https://dev.mysql.com/doc/refman/5.7/en/numeric-types.html

String/text data types in SQL

String/Text Data Types	Description
CHAR	Fixed-length string
VARCHAR	Variable-length string
MEDIUMTEXT **LONGTEXT**	Variable-length strings, with greater max length than VARCHAR MEDIUMTEXT - maximum: 16MB LONGTEXT - maximum size: 4GB

Reference: https://dev.mysql.com/doc/refman/5.7/en/string-types.html

Date data types in SQL

Date and Time Data Types	Description
DATE	A date stored in 'YYYY-MM-DD' format
TIME	A time stored in 'HH:MM:SS' format, can be used for elapsed time as well as time of day data; put another way, this data type is not limited to the 24-hour cycle
DATETIME	A date and time stored in 'YYYY-MM-DD HH:MM:SS' format
TIMESTAMP	A date and time stored in 'YYYY-MM-DD HH:MM:SS' UTC format
YEAR	A year stored in 'YYYY' format

Reference: https://dev.mysql.com/doc/refman/5.7/en/date-and-time-

Creating the customers table

3. Create the tables within the new database.
4. Insert data into the database.

```
USE creditcardco;

CREATE TABLE Customers ( ID INT,
CompanyName VARCHAR(255),
ContactEmail VARCHAR(255));

ALTER TABLE Customers
ADD PRIMARY KEY (ID);

-- OR

DROP TABLE IF EXISTS Customers;

CREATE TABLE Customers ( ID INT NOT NULL,
CompanyName VARCHAR(255) NOT NULL,
ContactEmail VARCHAR(255),
PRIMARY KEY (ID));
```

Walk through the CREATE and INSERT script together

- Check the database name in the script, and modify it to username_databasename

- Review the following:
  - INSERT
  - UPDATE
  - DELETE
  - ALTER
  - CREATION OF PRIMARY KEYS
  - CREATION OF FOREING KEYS (Script vs. Workbench GUI)

Summary

- The **data model** describes the data that is stored in the database and how to access it
- Each **record** is unique in a good database
- **Data models** enable users to understand **business rules** and effectively process and analyze data
- **Business rules** are imposed on the database through **relationships** and **cardinality**

Introduction to Relational Databases and Data Modeling

- How can these skills help with data analysis?