
MONTANA
STATE UNIVERSITY

The Relational Model

...

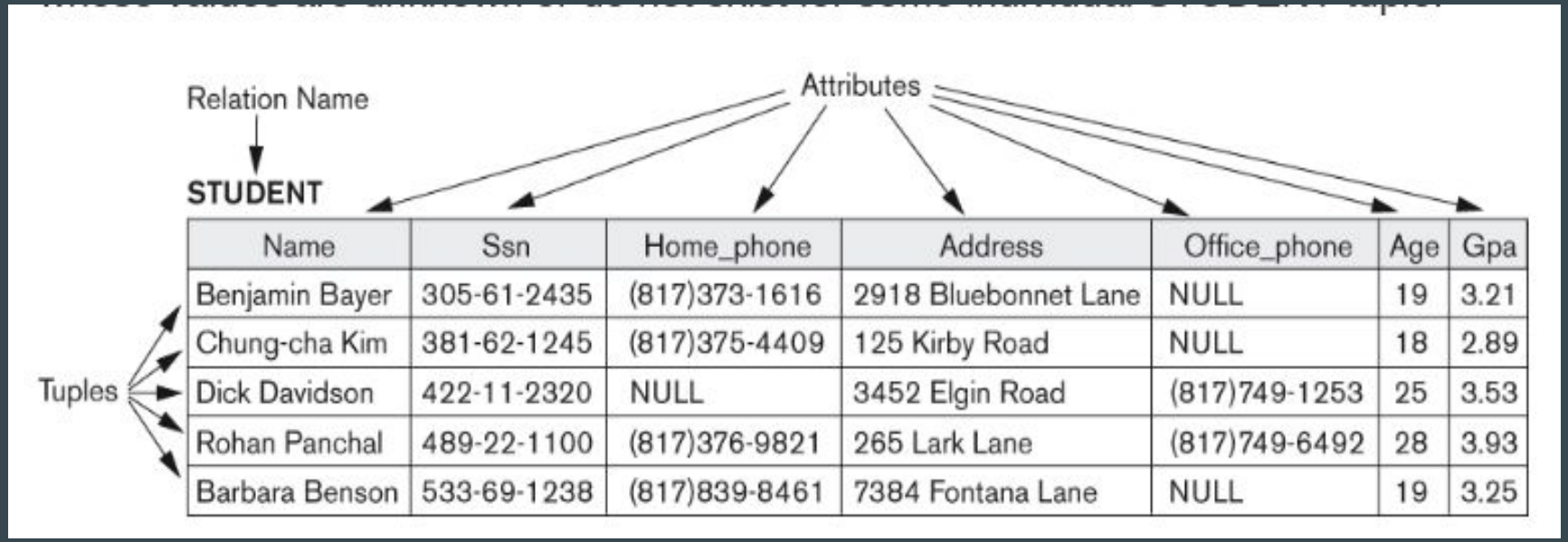
Getting Closer To The Real World...

The Relational Model

- A *Relation* is a set of *Tuples* that follow a *Relational Schema*
- Sounds fancy, but think of Tuples as rows
- Think of the schema as the columns
- The *Domain* of an attribute is the set of all possible values for that attribute
- Example Relational Schema:

  STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age)

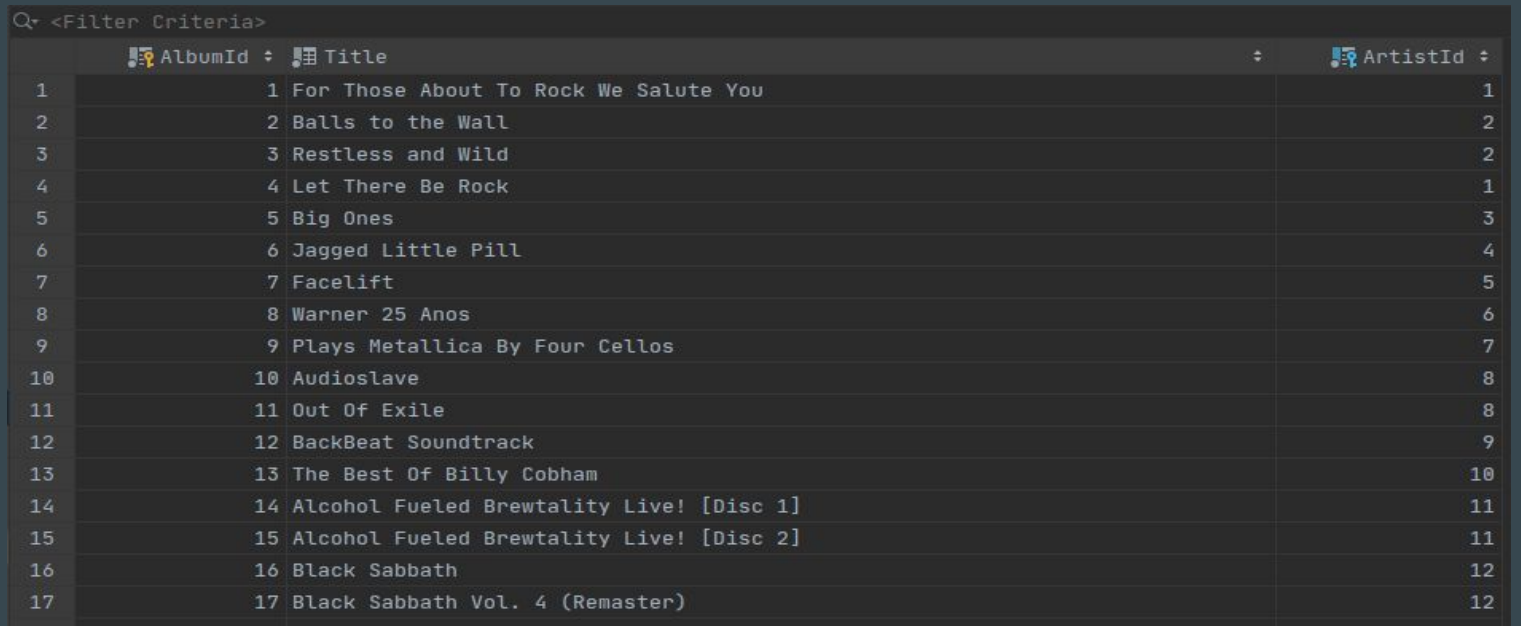
The Relational Model



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

The Relational Model

- Maybe easier to think of a relation as spreadsheet, but with typed & named columns, rather than free-form cells



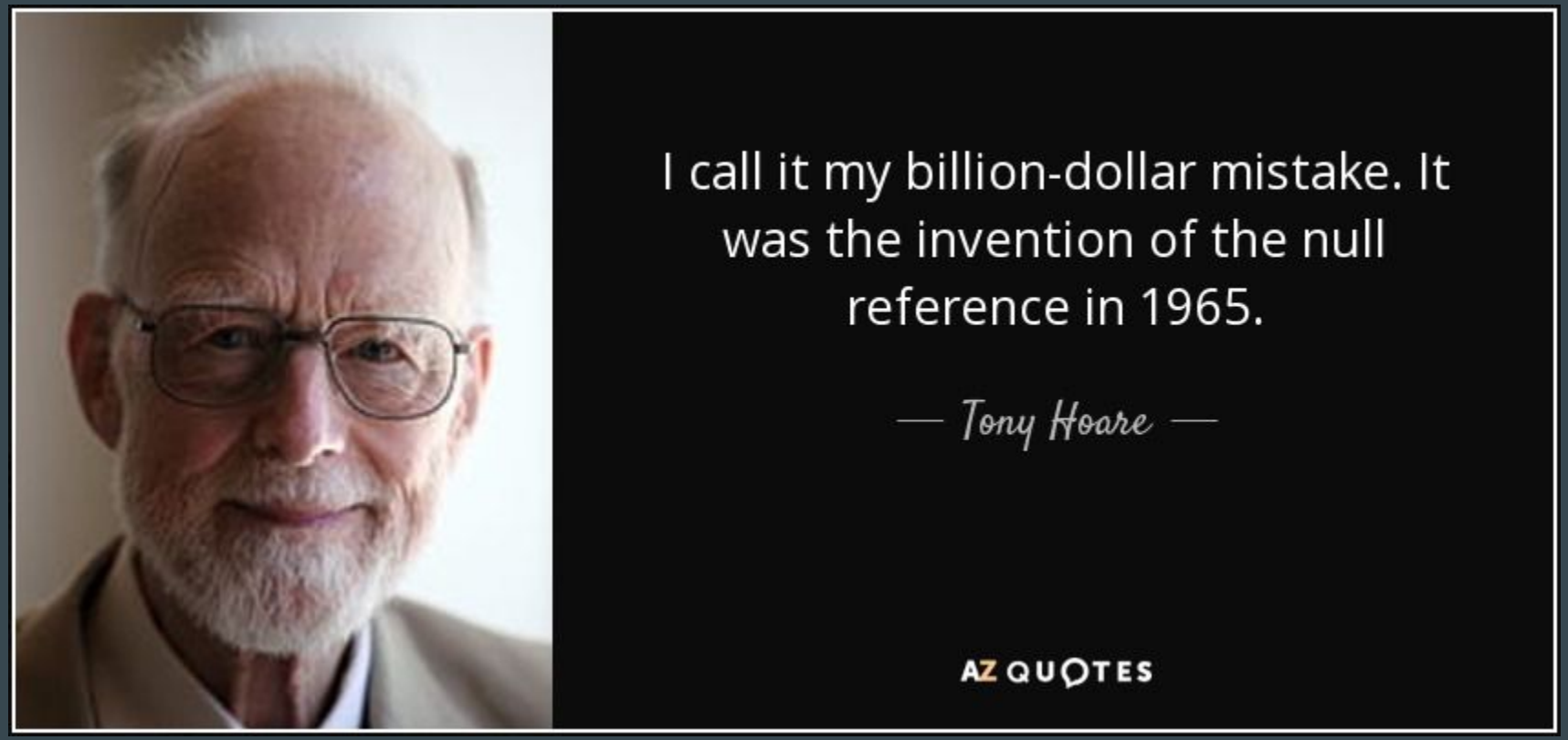
<Filter Criteria>

	AlbumId	Title	ArtistId
1	1	For Those About To Rock We Salute You	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cellos	7
10	10	Audioslave	8
11	11	Out Of Exile	8
12	12	BackBeat Soundtrack	9
13	13	The Best Of Billy Cobham	10
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11
16	16	Black Sabbath	12
17	17	Black Sabbath Vol. 4 (Remaster)	12

NULL Values

- Null values are controversial in both the relational model world as well as the programming world
- Is NULL part of the domain of values?
- Java primitives: no! Java objects: yes!
- Most databases allow null values for a column, even if that value maps to a primitive
    - What to do?

NULL Values


I call it my billion-dollar mistake. It was the invention of the null reference in 1965.
— Tony Hoare —
AZ QUOTES

NULL Values

- SELECT email FROM accounts WHERE balance > 100000
- Some SQL to find the big rollers in your online casino to give them a nice, fat coupon
  - Don’t worry about the details right now
- What about accounts with NULL for balance?
- This is how you lose money with nulls :)

Constraints In The Relational Model

- Recall that the set of constraints in the E/R diagram world was very limited
- The Relational Model formalizes constraints to a much larger extent
- We'll begin by looking at keys
  - Super Key - any combination of attributes for which two distinct tuples will have distinct values
  - Minimal Super Key - A super key from which no attributes can be removed and still be a super key
  - Candidate Key - A minimal super key
  - Primary Key - The Candidate key chose as the official key for the table

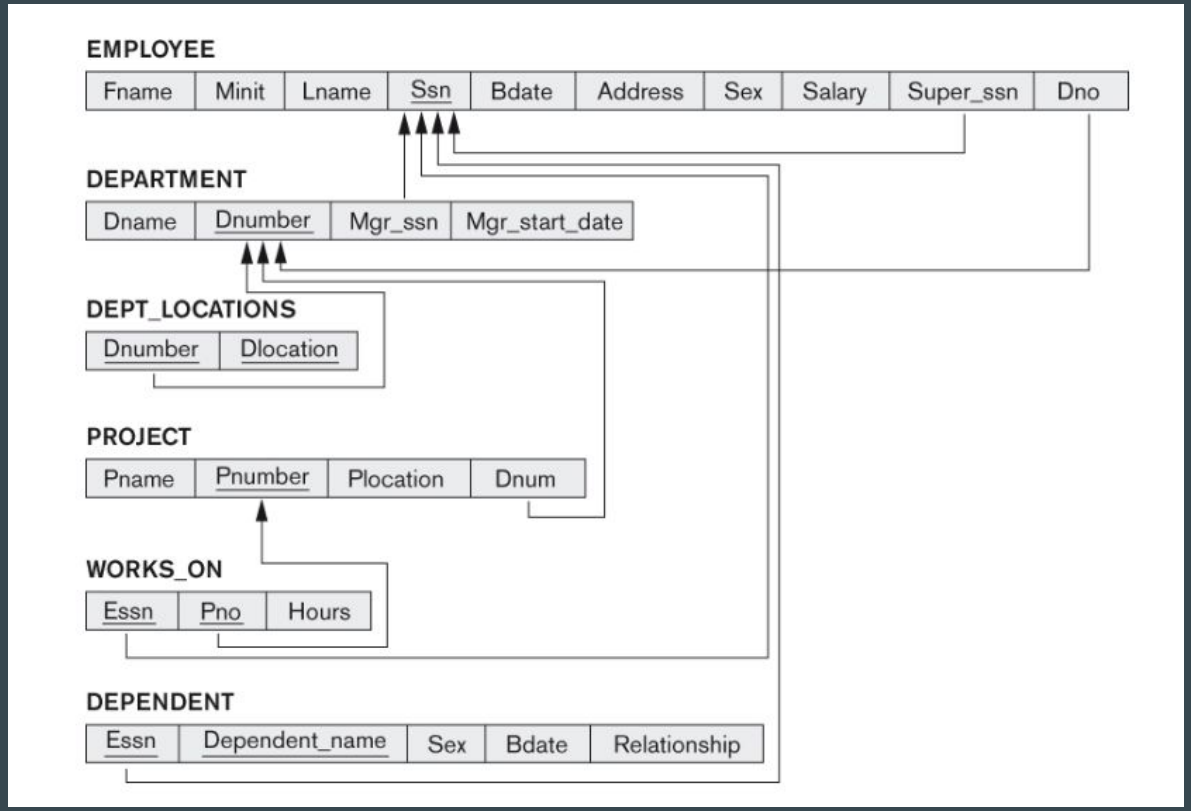
Keys

- In the real world, almost all database tables will have what is known as a *synthetic key*
- A synthetic key is a key that is (usually) auto-generated by the DBMS for a relation when it is inserted
- Very convenient
- *Problem: when I insert a new row in a DBMS, how do I know what the generated key was?!?*
    - Recall UUIDs: one nice feature of them is that you can generate them “up front” without relying on the database to do it for you

Constraints In The Relational Model

- What about *references* to key in another table?
  - A reference to another table is known as a *Foreign Key*
- This is known as *Referential Integrity*
- *Referential Integrity Constraints* are placed on a relational model (and on database tables) to ensure that Foreign Keys point to actual tuples or rows in other relations or tables

Constraints In The Relational Model



Constraints In The Relational Model

- In DBMS systems this are often referred to as *Foreign Key Constraints*
- In general, these are expensive
- Some systems drop them to make insert performance faster
  - Ruby on Rails doesn’t use them
  - I have rarely seen an issue without them
  - Please don’t tell Database Administrators about this!

Constraints In The Relational Model

- Additional Constraints
  - A string must be a valid email
  - A string must be from a set of choices (e.g. State)
    - AKA an enum
  - A column must be greater than or equal to another column
- The relational model and practical databases are often not good at expressing additional constraints
- These types of constraints are typically enforced at the application level
  - What's the problem with this?

Operations on Relations

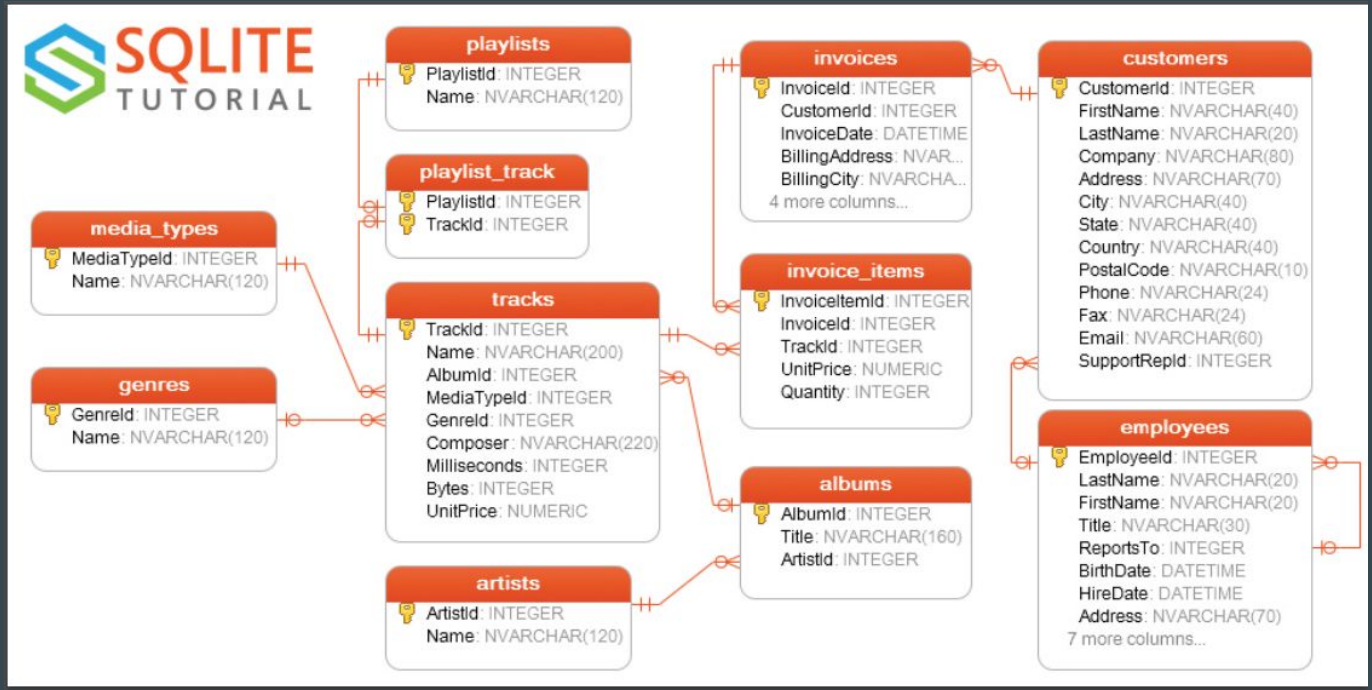
- Obviously you have retrievals: reading tuples from the relation
  - A large part of this class is going to be teaching you to be a SELECT ninja
  - SELECT is the SQL way to retrieve data
- You also have Update Operations
  - *Insert*
  - *Delete*
  - *Update*
- Also associated with these operations is the notion of a *Transaction*
  - an atomic unit of work against the database

Operations on Relations

- You will hear these operations referred to as CRUD
  - C - Create
  - R - Read
  - U - Update
  - D - Delete
- Simple web applications are often referred to as “crud front ends on a database”
- We will be building one such CRUD-dy app

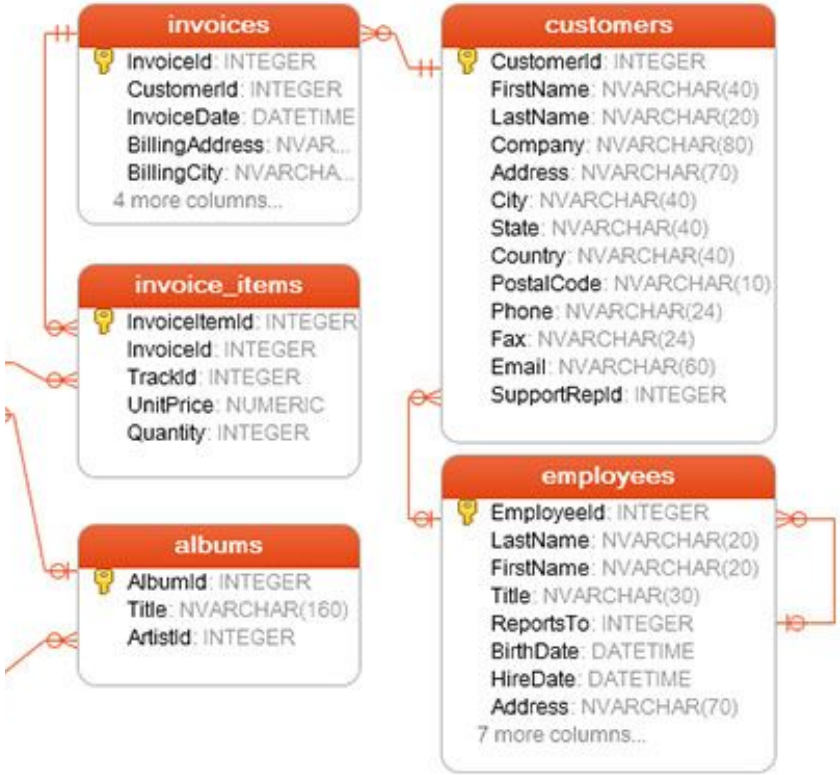
Visualizing A Relational Model

- chinookdb, the sample database we will use in the class



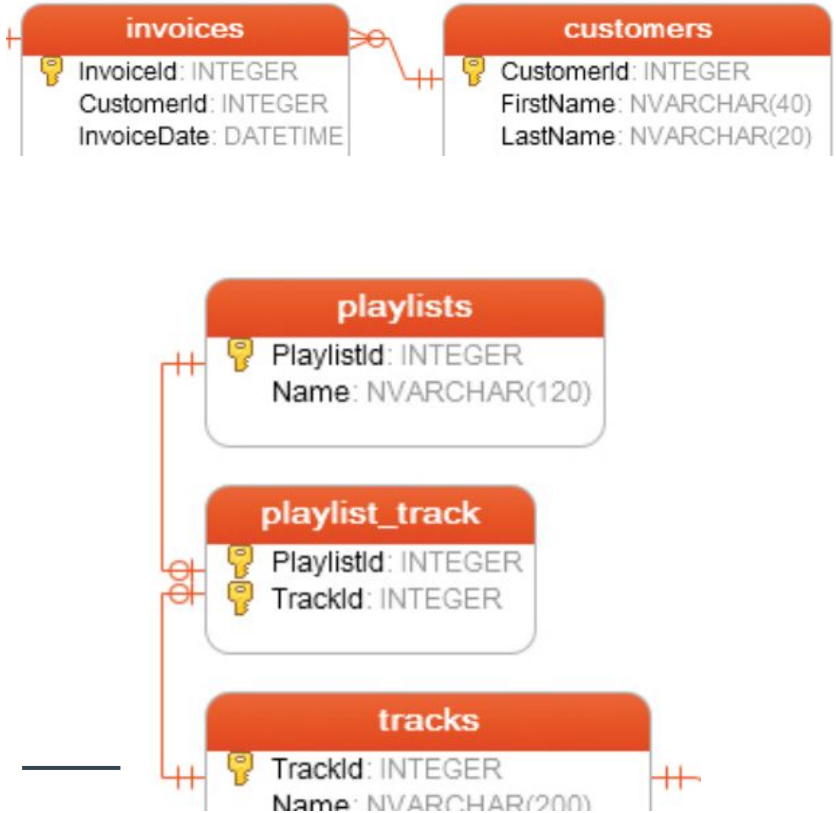
The Relational Model

- Notice the universality of INTEGER Ids
- Note the Foreign Key references
  - `invoices.CustomerId`
  - This is how relationships are encoded in the tables
- Notice that Employees has a self-referential key
  - This encodes a tree



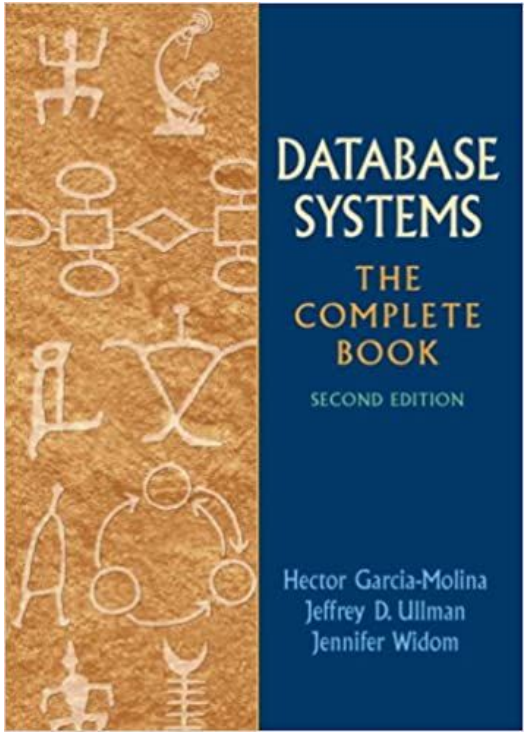
The Relational Model

- 1-N Relationship
  - FK is in N table
- N-N Relationship
  - Done with a join table with FK of both tables
- 1-1 Relationship
  - FK can be in either table



The Book

- The book is not great on this topic
- Dated material & overly academic language
- Luckly, we have other options for learning all this stuff



IntelliJ DB Demo

...


MONTANA
STATE UNIVERSITY