

# CS411 - The Relational Model



# Review

- What is a DBMS?
- What makes studying them different from other CS topics?
- What data model are most DBMSs based on?
- Who invented it?



# Review

- What is the difference between a DML and a DDL?
- What does “buffer management” refer to in the context of a DBMS?
- What does “ACID” stand for?
- Why is logging so important for a DBMS?



# Data Models

- A notation for describing data:
  1. structure of the data
  2. operations on the data
  3. constraints on the data



# The Relational Model

- structure - relations
- operations - relational algebra
- constraints - expressed in relational algebra and set theory



# Example Relation

First Name	Last Name	Phone	Email
Holden	Caufield	(217)-555-3251	nophoney@hotmail.com
Richard	Parker	(217)-555-1212	pi_delicious@gmail.com
Hucklberry	Finn	(217)-555-8519	raftboy@hotmail.com
Luke	Skywalker	(217)-555-2917	wompratbullseye@gmail.com
Bella	Swan	(217)-555-6666	edwardsgrl04@aol.com
Marty	McFly	(217)-555-1987	delorian88@gmail.com



# Example Operation (query)

- “Find the phone numbers of people with a gmail account.”



First Name	Last Name	Phone	Email	Phone
Holden	Caufield	(217)-555-3251	nophoney@hotmail.com	(217)-555-1212
Richard	Parker	(217)-555-1212	pi_delicious@gmail.com	(217)-555-2917
Huckleberry	Finn	(217)-555-8519	raftboy@hotmail.com	(217)-555-1987
Luke	Skywalker	(217)-555-2917	wompratbullseye@gmail.com	
Bella	Swan	(217)-555-6666	edwardsgrl04@aol.com	
Marty	McFly	(217)-555-1987	delorian88@gmail.com	



# Example constraints

- “Hugo Reyes” is not a valid phone number
- “4 , 8, 15, 16, 23, 42” is not a valid name
- Can’t have two people named “Ben Linus”



# The Semistructured Model

- We'll periodically revisit this model
- Presented here mainly as a contrast



# The Semistructured Model

- structure - hierarchical tags (XML)
- operations - tag traversal
- constraints - associated with tags



# Example XML

```
<People>
  <Person name="Han Solo">
    <Email>nerfherder@gmail.com</Email>
    <Phone>(217)555-8781</Phone>
    <Phone>(217)555-1444</Phone>
  </Person>
  <Person name="John McClane">
    <Phone>(217)555-1345</Phone>
    <Email>yippiekiyay@aol.com</Email>
  </Person>
</People>
```



# Example Operation

- “Find name of people in 217 area code.”

<People>

  <Person name=“Han Solo”>

    <Email>nerfherder@gmail.com</Email>

    <Phone>(217)555-8781</Phone>

    <Phone>(217)555-1444</Phone>

  </Person>

  <Person name=“John McClane”>

    <Phone>(217)555-1345</Phone>

    <Email>yippiekiyay@aol.com</Email>

  </Person>



</People>

# Example Constraints

- Values associated with “Email” must have realistic format
- A person cannot have more than 5 phone numbers
- Two people cannot have the same name



# Back to the Relational Model

- Clearly less expressive than semistructured model
- Why use it?
  - efficient storage representation
  - high level, easy to learn operators
  - optimization of operations can be automated

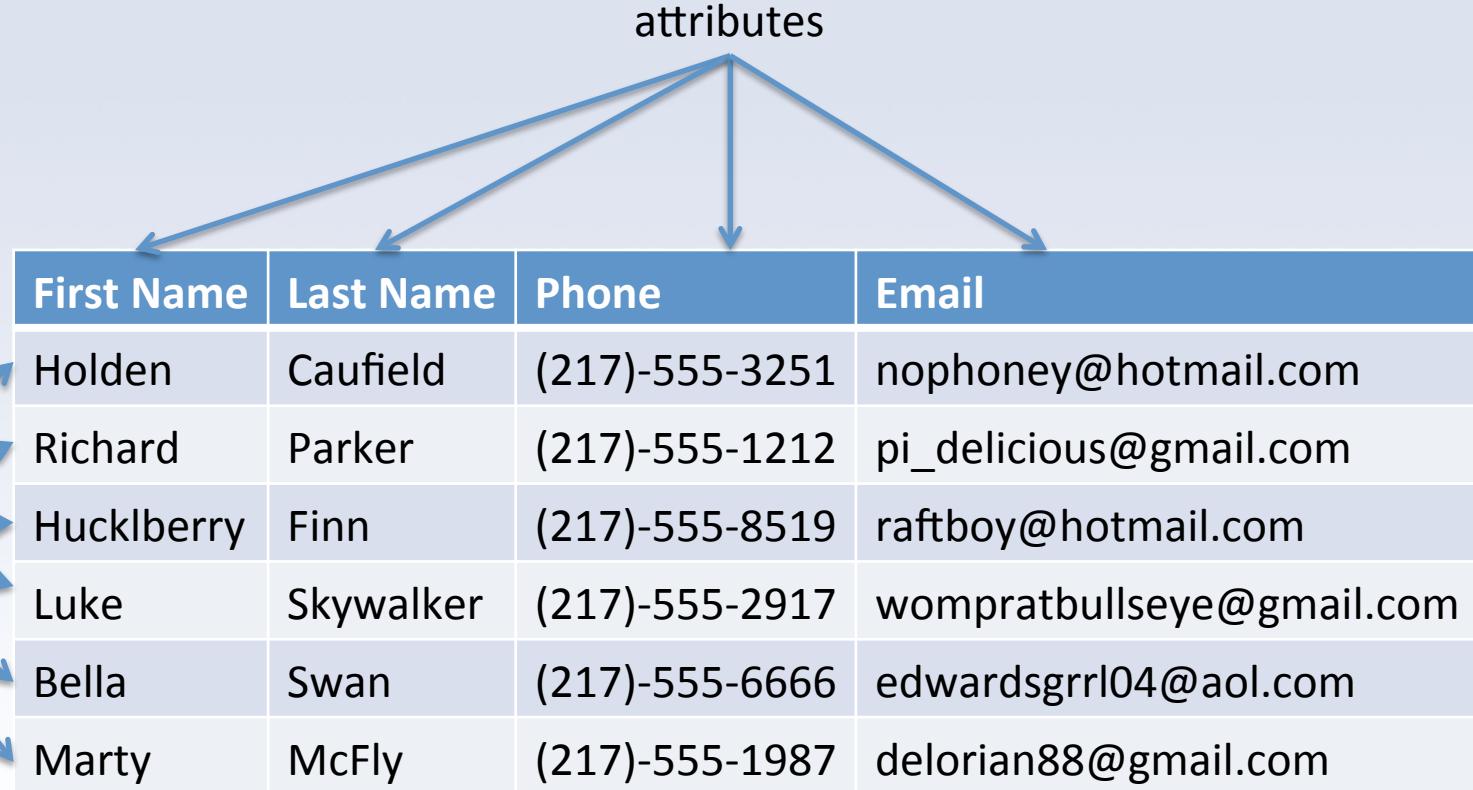


# Back to the Relational Model

- Relation - a table representing some data
- Attributes - the columns of the table
- Tuples - the rows of the table



# Back to the Relational Model





# Order doesn't matter

- This is the same relation:

Email	First Name	Phone	Last Name
pi_delicious@gmail.com	Richard	(217)-555-1212	Parker
wompratbullseye@gmail.com	Luke	(217)-555-2917	Skywalker
nophoney@hotmail.com	Holden	(217)-555-3251	Caufield
edwardsgrrl04@aol.com	Bella	(217)-555-6666	Swan
delorian88@gmail.com	Marty	(217)-555-1987	McFly
raftboy@hotmail.com	Huckleberry	(217)-555-8519	Finn



# Back to the Relational Model

- Schema for a relation
  - the name of the relation and its attributes
  - Represented like this:

Person(FirstName,LastName,Phone,Email)



# Back to relational model

- We differentiate between
  - schema
  - instance
  - current instance



# Back to the Relational Model

schema of relation → Person(FirstName,LastName,Phone,Email)

instance of relation



First Name	Last Name	Phone	Email
Holden	Caufield	(217)-555-3251	nophoney@hotmail.com
Richard	Parker	(217)-555-1212	pi_delicious@gmail.com
Huckleberry	Finn	(217)-555-8519	raftboy@hotmail.com
Luke	Skywalker	(217)-555-2917	wompratbullseye@gmail.com
Bella	Swan	(217)-555-6666	edwardsgr04@aol.com
Marty	McFly	(217)-555-1987	delorian88@gmail.com



# Back to the Relational Model

“Delete all the gmail users”

current instance of relation



First Name	Last Name	Phone	Email
Holden	Caufield	(217)-555-3251	nophoney@hotmail.com
Hucklberry	Finn	(217)-555-8519	raftboy@hotmail.com
Bella	Swan	(217)-555-6666	edwardsgrl04@aol.com



# Back to the Relational Model

- A *database* consists of many relations
- The set of schemas for all the relations is the *database schema*

Person(FirstName,LastName,Phone,Email)

Job(FirstName,LastName,Company,Title,Salary)

Company(Name,Industry,StockSymbol)



# Domains

- Each attribute must have an elementary type (e.g. string, integer)
- Sometimes we specify this explicitly:

Person(FirstName:string,LastName:string,Phone:integer,Email:string)



# Keys

- An important type of constraint
  - A set of attributes for a relation
  - No two tuples can share the same values for these attributes
  - Essentially provides a unique identifier for a specific tuple



# Keys

- We specify keys by underlining the attributes:

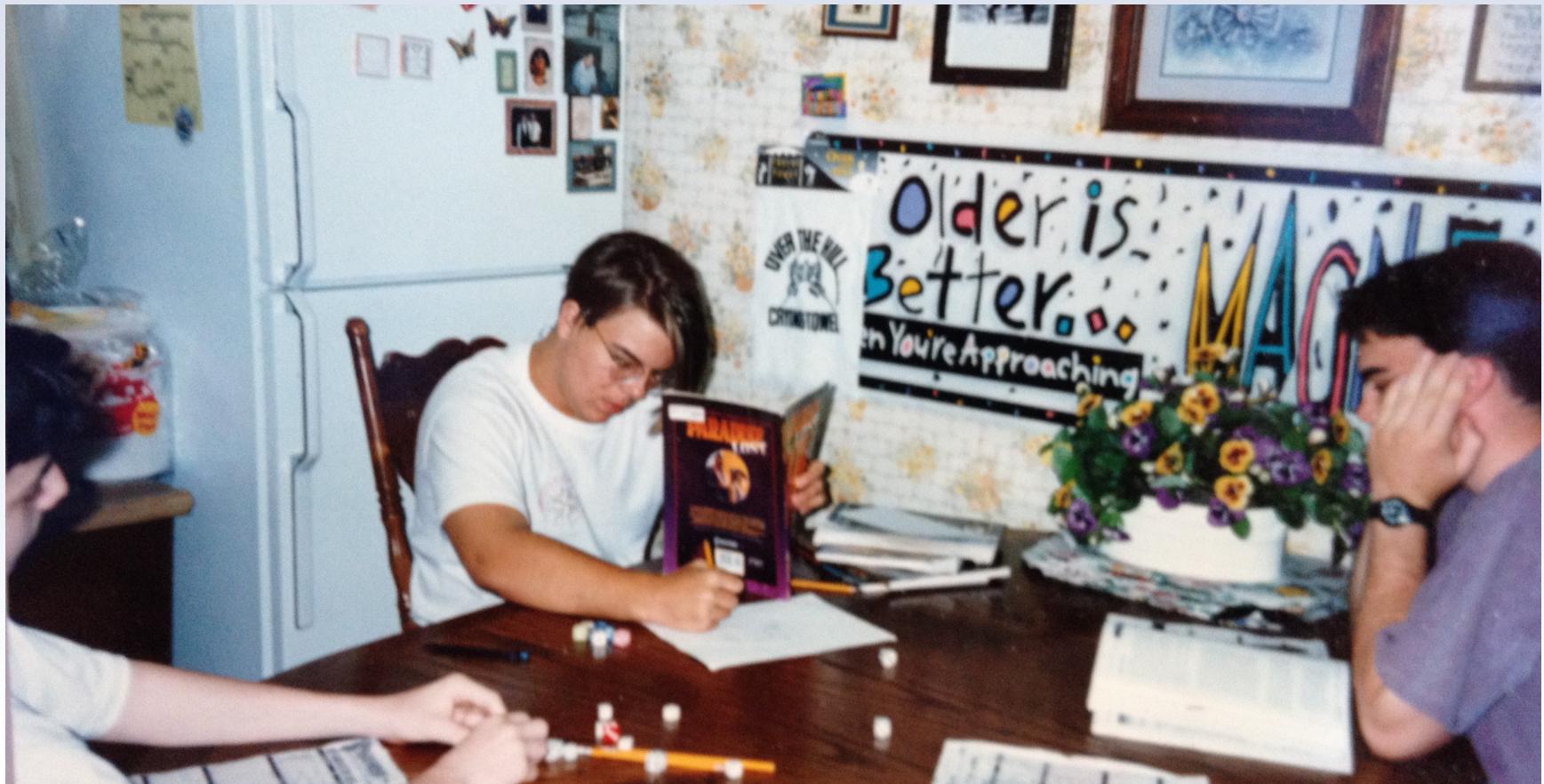
Person(FirstName,LastName,Phone,Email)

Job(FirstName,LastName,Company,Title,Salary)

Company(Name,Industry ,StockSymbol)



# Let's Role Play!



# Let's Role Play!

- We've been hired to build a music database
  - e.g iTunes, Media Player, Spotify, or Pandora



# Let's Role Play!

- What relations might we need?
- What attributes might they have?
- What would be the domains of the attributes?
- What would the keys be for each relation?



# Music database relations

- Band
- Musician
- Album
- Song



# What about attributes?

Band(Name,City,Genre,YearFormed,YearEnded,Label)

Musician(FirstName,LastName,Band,Instrument)

Album(Title,Band,DateReleased)

Song(Title,Album,Length)



# Band Relation

Name	City	Genre	Formed	Ended	Label
Nirvana	Seattle	Alt Rock	1987	1994	DCG
Talking Heads	New York	Post Punk	1975	1991	EMI
St. Vincent	Tulsa	Art Rock	2003	Present	4AD
Led Zeppelin	London	Hard Rock	1968	1980	Atlantic
Dinosaur Jr.	Amherst	Alt Rock	1984	1997	Merge
The Who	London	Art Rock	1964	1982	Warner Bros
Mastodon	Atlanta	Metal	2000	Present	Roadrunner
Fleetwood Mac	London	Pop Rock	1967	1995	Warner Bros
Joy Division	Salford	Post Punk	1976	1980	Factory
Kiss	New York	Hard Rock	1973	Present	Roadrunner
The Beatles	Manchester	Pop Rock	1960	1970	Capitol



# Musician Relation

First Name	Last Name	Band	Instrument
Stevie	Nicks	Fleetwood Mac	Vocals
Kurt	Cobain	Nirvana	Vocals, Guitar
J	Mascis	Dinosaur Jr.	Vocals, Guitar
Anne	Clark	St. Vincent	Vocals, Guitar
Gene	Simmons	Kiss	Vocals, Bass
Paul	Stanley	Kiss	Guitar
Paul	McCartney	The Beatles	Vocals, Bass
John	Lennon	The Beatles	Vocals, Guitar
Pete	Townsend	The Who	Vocals, Guitar
Keith	Moon	The Who	Drummer
David	Byrne	Talking Heads	Vocals, Guitar
Ian	Curtis	Joy Division	Vocals, Guitar



# Album Relation

Title	Band	DateReleased
Nevermind	Nirvana	09/24/1991
Rumors	Fleetwood Mac	02/04/1977
Revolver	The Beatles	08/05/1966
Abbey Road	The Beatles	09/26/1969
Tommy	The Who	05/28/1969
IV	Led Zeppelin	11/08/1971
Leviathan	Mastodon	08/31/2004
Remain in Light	Talking Heads	10/08/1980
Dressed to Kill	Kiss	03/19/1975
Strange Mercy	St. Vincent	09/13/2011
Without a Sound	Dinosaur Jr.	08/23/1994
Closer	Joy Division	07/18/1980



# Song Relation

Title	Album	Length
Isolation	Closer	2:53
Smells Like Teen Spirit	Nevermind	5:01
Come Together	Abbey Road	4:18
Feel The Pain	Without a Sound	4:18
Go Your Own Way	Rumors	3:38
Pinball Wizard	Tommy	3:01
Eleanor Rigby	Revolver	2:08
Stairway to Heaven	IV	8:02
Once in a Lifetime	Remain in Light	4:19
I Am Ahab	Leviathan	2:45
Cruel	Strange Mercy	3:35
Rock and Roll All Night	Dressed to Kill	2:34



# Is this a good design?

- Can two musicians share the same name?
- Can a musician be in two bands?
- What about the country and province for each band?
- What about the track number for each song?
- Can bands have multiple labels?



# Is this a good design?

- Probably not
- But we'll learn more about designing database schemas later



# SQL

- Both a DDL and DML
- Three kinds of relations:
  - Tables
  - Views
  - Temporary internal relations



# Common SQL Data Types

- CHAR(n), VARCHAR
- BIT(n)
- BOOLEAN
- INTEGER
- FLOAT, DOUBLE, DECIMAL(m,n)
- DATE, TIME



# Creating SQL Tables

Band(Name,City,Genre,YearFormed,YearEnded,Label)

```
CREATE TABLE Band(  
    name          CHAR(100) PRIMARY KEY,  
    city          CHAR(100),  
    genre         CHAR(100),  
    yearFormed   DATE,  
    yearEnded    DATE,  
    label         CHAR(100)  
);
```



# Creating SQL Tables

Musician(FirstName,LastName,Band,Instrument)

```
CREATE TABLE Musician(  
    firstName          CHAR(100),  
    lastName          CHAR(100),  
    band              INTERVAL,  
    instrument        VARCHAR,  
    PRIMARY KEY        (firstName,lastName)  
);
```



# Creating SQL Tables

Album (Title,Band,DateReleased)

```
CREATE TABLE Album(  
    title          CHAR(100),  
    band          CHAR(100),  
    dateReleased  DATE,  
    PRIMARY KEY    (title)  
);
```



# Creating SQL Tables

`Song(Title,Album,Length)`

```
CREATE TABLE Song(  
    title          CHAR(100),  
    album          CHAR(100),  
    length         INTERVAL,  
    PRIMARY KEY    (title)  
);
```



# Modifying a Table

Add a birthday attribute for Musicians:

```
ALTER TABLE Musician ADD birthday DATE;
```

```
Musician(FirstName,LastName,Band,Birthday,Instrument)
```



# Modifying a Table

Add a price attribute for Albums:

```
ALTER TABLE Album ADD price DECIMAL(5,2);
```

Album(Title,Band,DateReleased,Price)



# Modifying a Table

Remove instrument attribute for Musicians:

ALTER TABLE Musician DROP instrument

Musician(FirstName,LastName,Band,Birthday)



# Deleting SQL Tables

- `DROP TABLE Album;`

`Band(Name,City,Genre,YearFormed,YearEnded,Label)`

`Musician(FirstName,LastName,Band,Birthday)`

`Song(Title,Album,Length)`



# Two Kinds of “Key” Constraints

- **UNIQUE**
  - The attributes must be unique
- **PRIMARY KEY**
  - Only one allowed per relation
  - Attributes must be unique *and* not null



# Other SQL constraints

- Attributes can also be NOT NULL
  - Exactly what it sounds like
- DEFAULT <value>
  - If the value for a tuple is unspecified, it will automatically be filled in with the default value



???

- What's this all about?

$$\pi_{Song.name}(\sigma_{dateReleased < 01-01-01}(Album) \bowtie_{Album.name = Song.album} \sigma_{length > 3:00}(Song))$$


# Next Lecture

- Relational algebra
  - Abstract representation of queries

