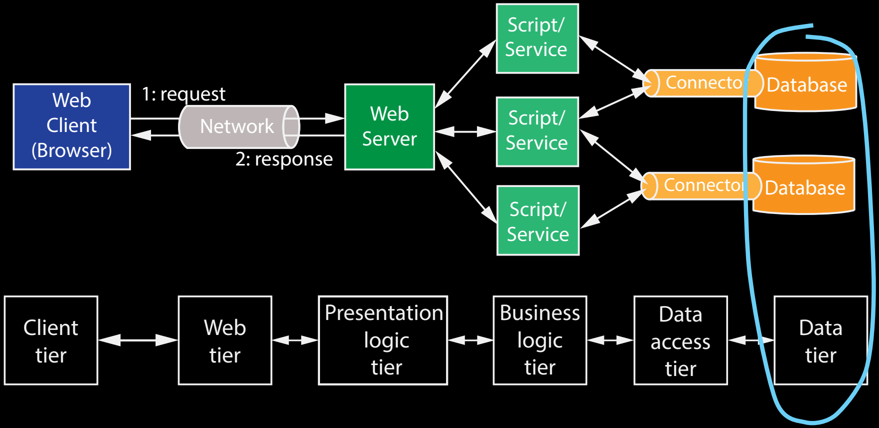
Web Application Architectures

Module 3: Database Interactions
Lecture 1: Relational Databases

THE UNIVERSITY *of* NEW MEXICO

Focus of our Attention



Relational Databases

- Relational databases are the most common way to persistently store data in web applications.
- A relational database is used to store a collection of relations. This involves storing "records" in tables.
- Each row in a table (or entity) corresponds to one record, and the columns correspond to fields (or attributes) of the record.

Ex.

people

id	first_name	last_name	address	phone
1	Frank	Furter	1 Wiener Way	212-555-1234
2	John	Doe	30 Smith PL	505-555-1234
3	Jane	Doe	30 Smith PL	505-463-4321
4	John	Doe	30 Smith PL	505-463-1234

Relational Databases

- What's the point of the `id` field in the previous table? It is used to form relationships to other tables. It's referred to as the primary key of the table.
- For example, `John Doe` appeared in the previous table twice. Why? Because he has two phones. More specifically, there is a one-to-many relationship between people and phones — one person can have many phones.
- We can normalize the database by creating two tables, one for people and a separate table for phones:
  - Each record in the `phone` table will hold the `id` of a person.
  - In the `phone` table, this `person_id` is referred to as a foreign key.
  - Given the `id` of a `person`, we can now search the `phone` table for all of the phones that belong to a person. This is typically done using the structured query language (SQL), but in Rails we'll by-pass this using the methods provided with Active Records.

Relational Databases – Normalization

Ex. A one-to-many relationship between tables:

people

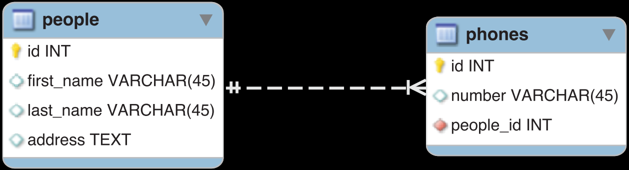
id	first_name	last_name	address
1	Frank	Furter	1 Wiener Way
2	John	Doe	30 Smith PL
3	Jane	Doe	30 Smith PL

phones

id	person_id	number
1	1	212-555-1234
2	2	505-555-1234
3	3	505-463-4321
4	2	505-463-1234

Schema and Entity-Relationship Models

- The structure/organization of the tables in a database is referred to as a schema.
- An entity-relationship model is a common way of abstractly capturing a database schema.

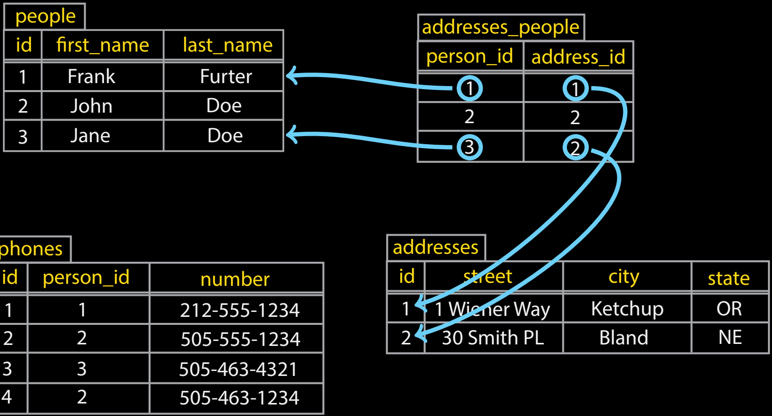


Relational Databases

- Notice that we could further normalize the database by creating an `address` table.
- However, in this case, the one-to-many relationship is in the other direction, i.e., we have one address for many people in the table.
- You can imagine a situation where one person also has many addresses, e.g., one for work, one for home, etc.
- Thus, we really need to create a many-to-many relationship between `people` and `addresses`.
- This is done by creating a join table — it's called this because it "joins" the `people` and `addresses` tables.
- The join table in the following example is called `addresses_people`. Notice that it only stores foreign keys, and has no primary keys.

Relational Databases – Join Tables

Ex. A many-to-many relationship using a join table:



people

id	first_name	last_name
1	Frank	Furter
2	John	Doe
3	Jane	Doe

addresses_people

person_id	address_id
1	1
2	2
3	2

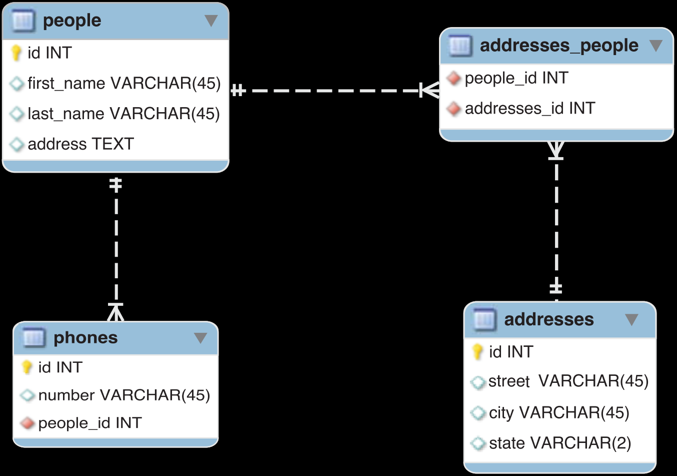
phones

id	person_id	number
1	1	212-555-1234
2	2	505-555-1234
3	3	505-463-4321
4	2	505-463-1234

addresses

id	street	city	state
1	1 Wiener Way	Ketchup	OR
2	30 Smith PL	Bland	NE

Entity-Relationship Model


people
id INT
first_name VARCHAR(45)
last_name VARCHAR(45)
address TEXT
addresses_people
people_id INT
addresses_id INT
phones
id INT
number VARCHAR(45)
people_id INT
addresses
id INT
street VARCHAR(45)
city VARCHAR(45)
state VARCHAR(2)