Data Modeling

FinTech

Lesson 7.3

Class Objectives

By the end of today's class, you will be able to:

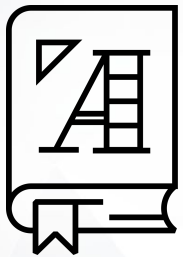
Apply data modeling techniques to database design.

Normalize data.

Identify data relationships.

Create visual representations of a database through entity relationship diagrams.

Data Normalization



Data normalization is the process of restructuring data to a set of defined "normal forms."

The process of data normalization **eliminates data redundancy and inconsistencies**.

Data Normalization

- The process of restructuring data to a set of “normal forms.”
- Reduces and eliminates data redundancy and inconsistencies
- Three most common forms:
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)
- There are even more levels!

First Normal Form (1NF)

Each field in a table row should contain a single value.

Each row is unique.

- Rows can have fields that repeat.
- Whole rows do not fully match.

Raw Data

family	children
Smith	Chris, Abby, Susy
Jones	Steve, Mary, Dillion

Normalization

First Normal Form

family	children
Smith	Abby
Smith	Susy
Jones	Mary
Smith	Chris
Jones	Dillion
Jones	Mary

Second Normal Form (2NF)

Must be in first normal form

Single column primary key

- Primary key
- Identifies the table and row uniquely

Generally, there could be a need to create a new table.

Data in 1NF

family	children
Smith	Abby
Smith	Susy
Jones	Mary
Smith	Chris
Jones	Dillion
Jones	Mary

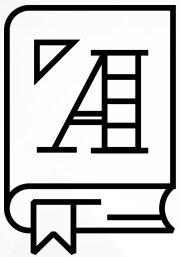
2NF Normalization

Family Table

family_id	family
1	Smith
2	Jones

Child Table

child_id	family_id	children
11	1	Chris
22	1	Abby
33	1	Susy
44	2	Steve
55	2	Mary
66	2	Dillion



Transitive dependency is the reliance of a column’s value on another column through a third column.

Transitive Dependency

Transitive	If X > Y and Y > Z, then X > Z.
Dependency	• One value relies on another. • Examples: city relies on postal code; age relies on birthday.
For example	• Say you have three columns: `StoreName`, `OwnerAddress`, `OwnerName`. • `OwnerName` and `OwnerAddress` rely on `StoreName`. • `OwnerAddress` also depends on `OwnerName`. • Therefore, `OwnerAddress` is transitively dependent on `StoreName` through `OwnerName`.

Third Normal Form (3NF)

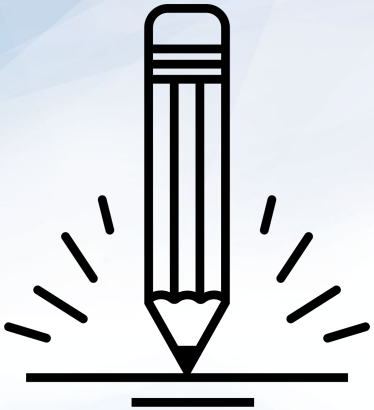
- Must be in second normal form
- Contains non-transitively dependent columns

owner_id	owner_name	owner_address	store_name
11	Marshall	123, Fake Street	Soups and Stuff
22	Susan	44, New Drive	Sink Emporium
33	Molly	99, Old Lane	Tasty Burgers

3NF Normalization

owner_id	owner_name	owner_address
11	Marshall	123, Fake Street
22	Susan	44, New Drive
33	Molly	99, Old Lane

store_id	store_name	Owner_id (fk)
1	Soups and Stuff	11
2	Sink Emporium	22
3	Tasty Burgers	33



Activity: Employee Normalizer

In this activity, you will be organizing improperly stored employee data into the three normal forms.

(Instructions sent via Slack)

Suggested Time:
15 minutes

Time’s Up! Let’s Review.

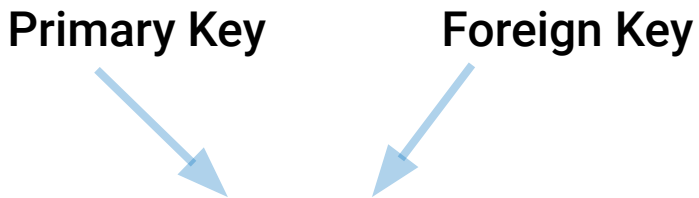
Foreign Keys

Foreign Keys

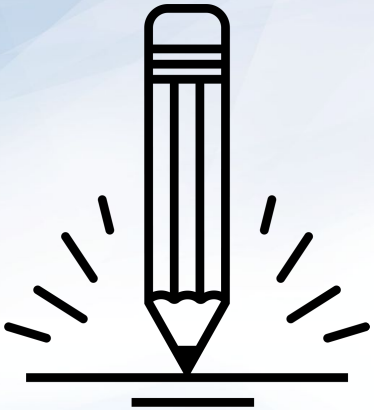
Foreign keys reference the primary key of another table.

- Can have a different name
- Do not need to be unique

family_id	family
1	Smith
2	Jones



child_id	family_id	children
11	1	Chris
22	1	Abby
33	1	Susy
44	2	Steve
55	2	Mary
66	2	Dillion



Activity: Foreign Keys

In this activity, you will create tables with foreign keys.

(Instructions sent via Slack)

Suggested Time:
15 minutes

Time’s Up! Let’s Review.

Data Relationships

Data Relationships

01

One-to-One

02

One-to-Many

03

Many-to-Many

One-to-One Relationship

ID	Name	Social Security
1	Homer	111111111
2	Marge	222222222
3	Lisa	333333333
4	Bart	444444444
5	Maggie	555555555

- Each item in one column is linked to only one other item from the other column.
- Here, each person in the Simpson family can have only one Social Security number.
- Each Social Security number can be assigned to only one person.

One-to-Many Relationship

ID	Address
11	742 Evergreen Terrace
12	221B Baker Street

ID	Name	Social Security	AddressID
1	Homer	111111111	11
2	Marge	222222222	11
3	Lisa	333333333	11
4	Bart	444444444	11
5	Maggie	555555555	11
6	Sherlock	112233445	12
7	Watson	223344556	12

- Two tables: one for people, another for addresses.
- Each person has only one address.
- But each address can be associated with multiple people.

One-to-Many Relationship

ID	Address
11	742 Evergreen Terrace
12	221B Baker Street

ID	Name	Social Security	AddressID
1	Homer	111111111	11
2	Marge	222222222	11
3	Lisa	333333333	11
4	Bart	444444444	11
5	Maggie	555555555	11
6	Sherlock	112233445	12
7	Watson	223344556	12

- The two tables, joined, would look like this.
- Each person has an address.
- Each address can be associated with multiple people.

Many-to-Many Relationship

ID	Child		ID	Parent
1	Bart		11	Homer
2	Lisa		12	Marge
3	Maggie			

Each child can have more than one parent.

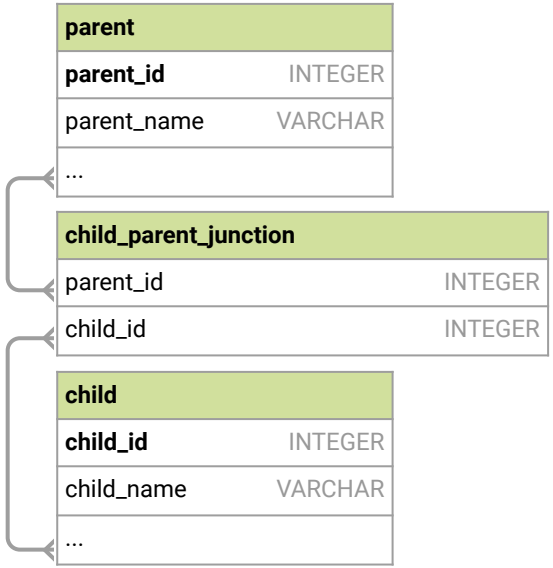
Each parent can have more than one child.

Many-to-Many Relationship

ChildID	Child	ParentID	Parent
1	Bart	11	Homer
1	Bart	12	Marge
2	Lisa	11	Homer
2	Lisa	12	Marge
3	Maggie	11	Homer
3	Maggie	12	Marge

- Each child can have more than one parent.
- Each parent can have more than one child.
- The two tables are joined in a **junction table**.

Junction Table

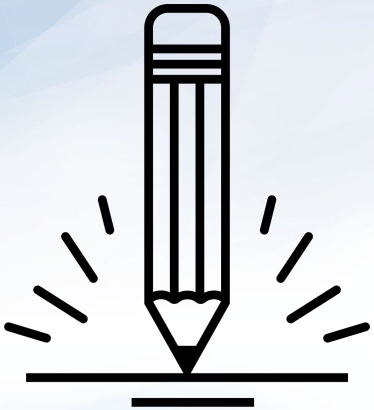


	parent_id integer	child_id integer
1	11	1
2	11	2
3	11	3
4	12	1
5	12	2
6	12	3

Join child and parent table to junction table

	parent_name character varying (255)	child_name character varying (255)
1	Homer	Bart
2	Homer	Lisa
3	Homer	Maggie
4	Marge	Bart
5	Marge	Lisa
6	Marge	Maggie

The junction table contains many parent_ids and many child_ids.



Activity: Data Relationships

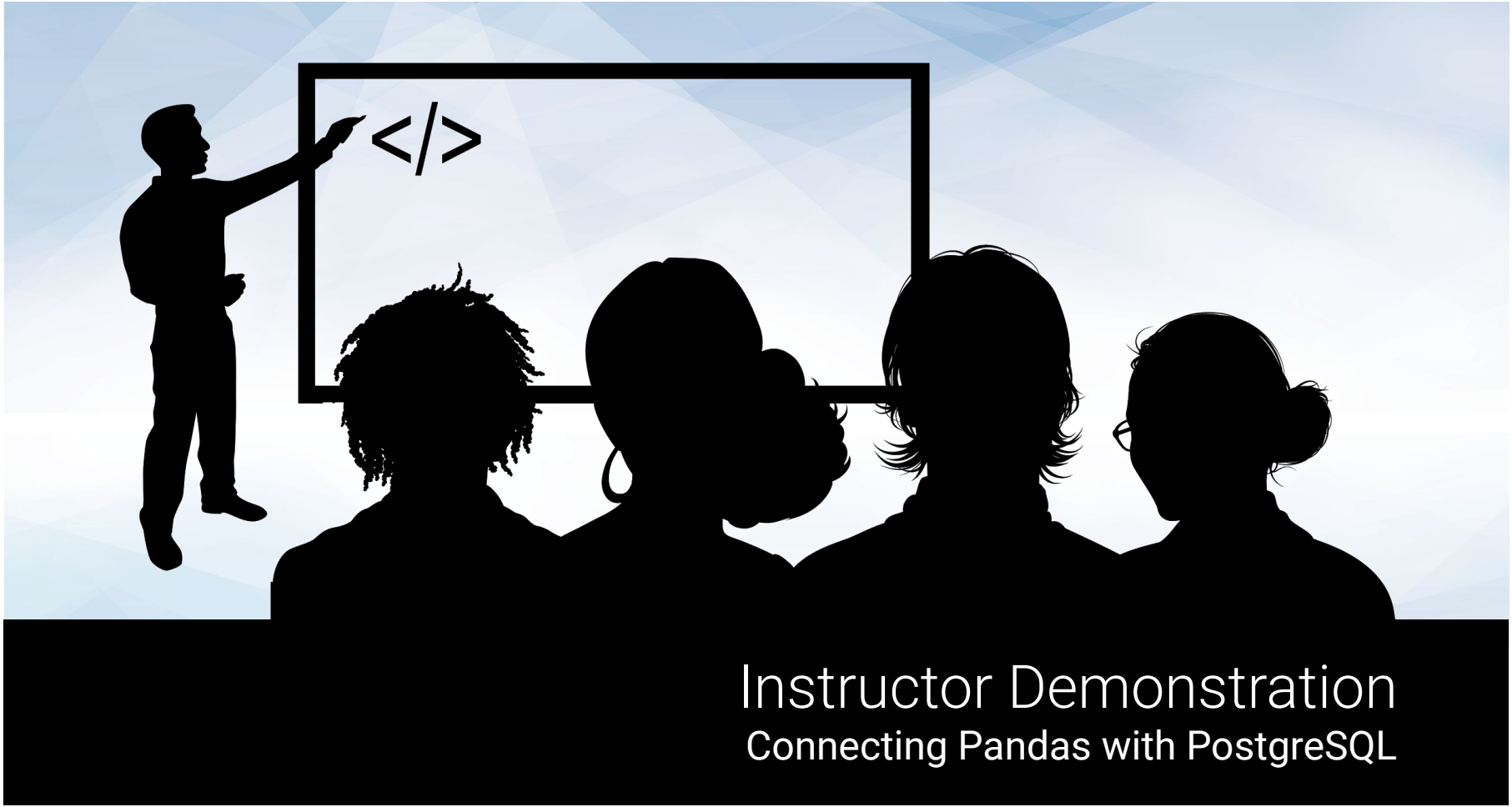
In this activity, students will create table schemata for agents and regions, and then create a junction table to display all regions assigned to agents.

(Instructions sent via Slack)

Suggested Time:
15 minutes

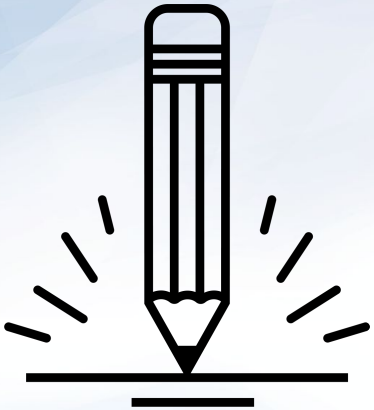
Time’s Up! Let’s Review.


Break



Instructor Demonstration

Connecting Pandas with PostgreSQL



Activity: Feeding Pandas with SQL

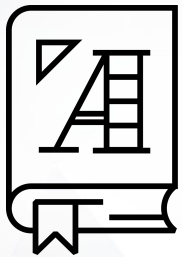
In this activity, students will read data into a Pandas DataFrame from a PostgreSQL database using SQLAlchemy.

(Instructions sent via Slack)

Suggested Time:
15 minutes

Time’s Up! Let’s Review.

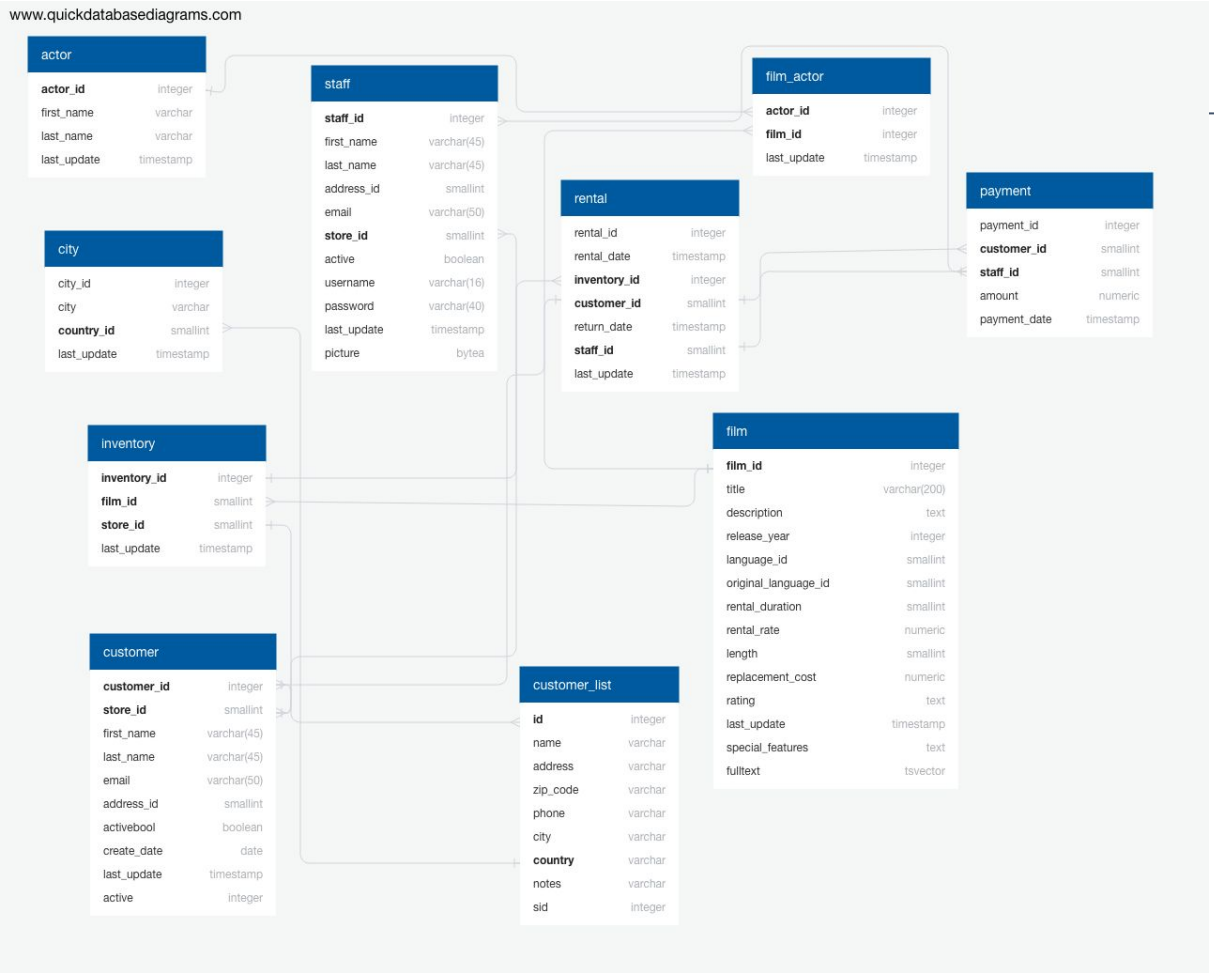
Entity Relationship Diagrams



An **entity relationship diagram,** or **ERD**, is a visual representation of entity relationships within a database.

ERDs

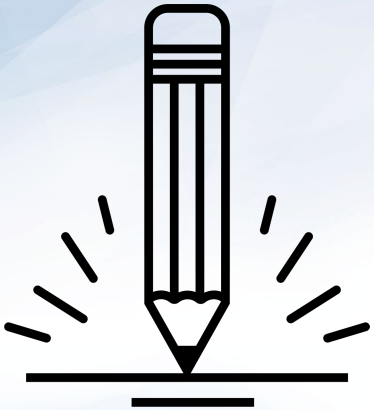
Entities, their data types, and relationships are all illustrated in the diagram.



ERDs

There are three types of ERDs, or data models, used when creating diagrams:

Conceptual	Basic information containing table and column names.
Logical	Slightly more complex than conceptual models with primary & foreign keys defined.
Physical	Blueprint of the database, reflecting relationships, attributes and data types.



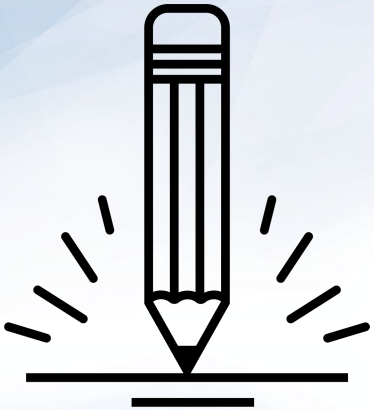
Activity: Designing an ERD, Part 1

In this activity, you will work with a partner to create a mortgage lending database model.

(Instructions sent via Slack)

Suggested Time:
15 Minutes

Time’s Up! Let’s Review.



Activity: Designing an ERD, Part 2

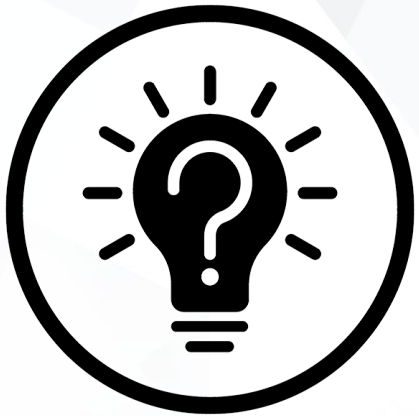
In this activity, you and your partner will continue designing a database model for the mortgage lending database by transitioning your logical ERD created in the previous activity to a physical ERD.

(Instructions sent via Slack)

Suggested Time:
15 minutes

Time’s Up! Let’s Review.

Structured Review



QUESTIONS


The End