

Slide 1: Data Models Part 3: Relational vs. Transactional Models

Data Models Part 3: Relational vs. Transactional Models

Sadie St. Lawrence, MSA
Data Scientist, VSP

UC DAVIS
EXTENSION

Slide 2: Learning Objectives

Learning Objectives

Define and describe both relational and transactional database models

Define entities, attributes, and relationships

Describe and explain the differences between a one-one, one-many, and many-many relationships

Slide 3: Learning Objectives

Learning Objectives

Describe primary keys in a database

Explain how ER diagrams are used to document and illustrate relationships

Slide 4: Relational vs. Transactional Model

Relational vs. Transactional Model

Relational Model

Allows for easy querying and data manipulation in an easy, logical and intuitive way

Transactional Model

Operational database – insurance claims within a healthcare database

Slide 5: Data Model Building Blocks

Data Model Building Blocks

Entity:

Person, place thing or event
Distinguishable, unique, and distinct

Attribute:

A characteristic of an entity

Relationship:

Describes association
among entities

- One-to-many
- Many-to-many
- One-to-one

Slide 6: Data Model Building Blocks

Data Model Building Blocks

One-to-many: customer to invoices

Many-to-many: student to classes

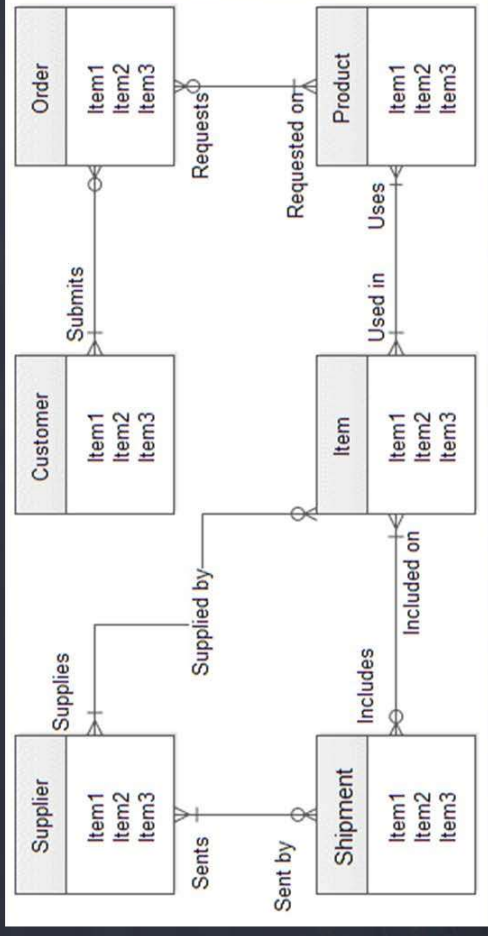
One-to-one: manager to store

Slide 7: ER Diagrams

ER Diagrams

ER model

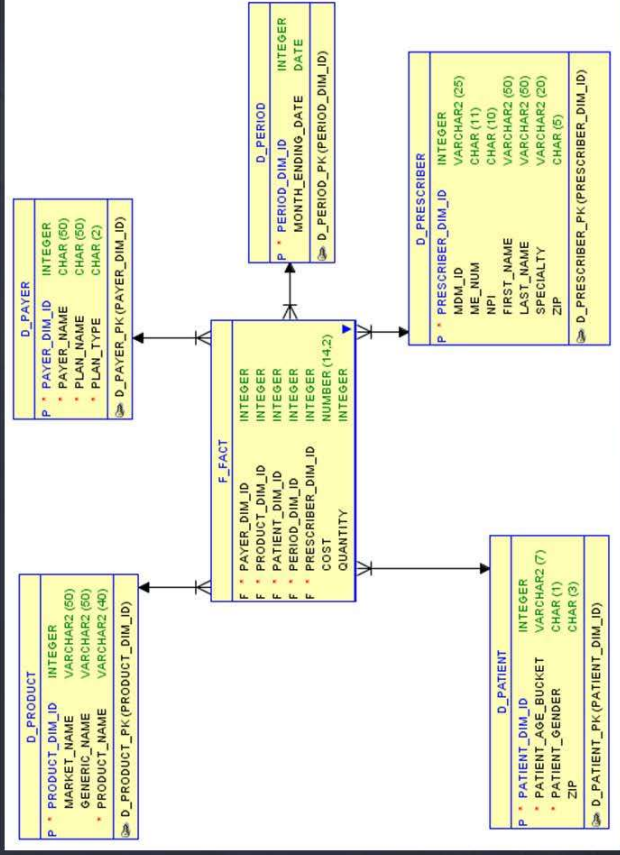
Is composed of entity types and specifies relationships that can exist between instances of those entity types



Slide 8: ER Diagrams

ER Diagrams

- Show relationships
- Business process
- Represented visually
- Show links (primary keys)



Slide 9: Primary Keys and Foreign Keys

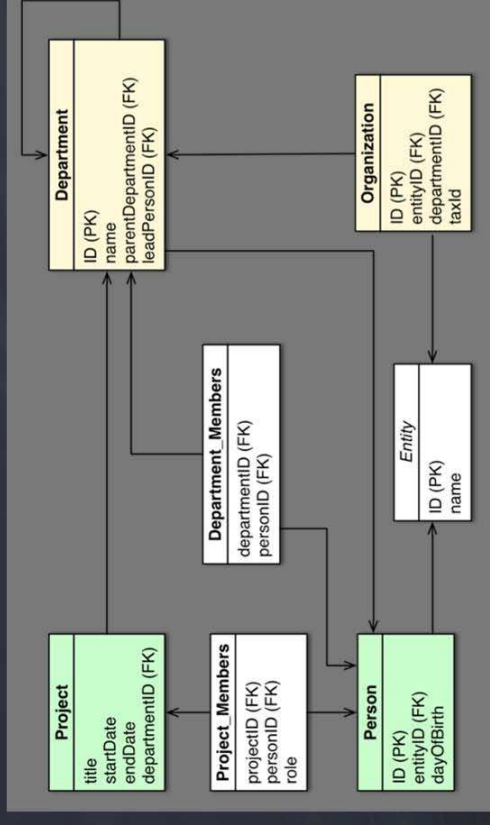
Primary Keys and Foreign Keys

Primary Key

A column (or set of columns) whose values uniquely identify every row in a table

Foreign Key

One or more columns that can be used together to identify a single row in another table



Slide 10: ER Diagram Notation

ER Diagram Notation

Chen Notation

Crow's Foot Notation

UML Class Diagram Notation

Slide 11: Chen Notation

Chen Notation

A **One-to-Many (1:M) Relationship**: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A **Many-to-Many (M:N) Relationship**: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A **One-to-One (1:1) Relationship**: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Slide 12: Crow's Foot Notation

Crow's Foot Notation

A **One-to-Many (1:M) Relationship**: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A **Many-to-Many (M:N) Relationship**: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



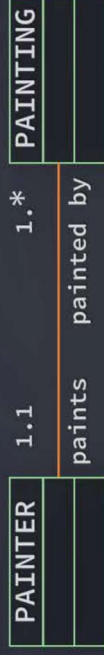
A **One-to-One (1:1) Relationship**: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



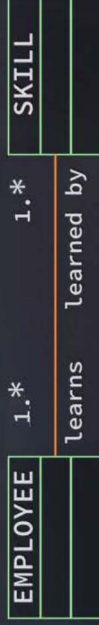
Slide 13: UML Class Diagram Notation

UML Class Diagram Notation

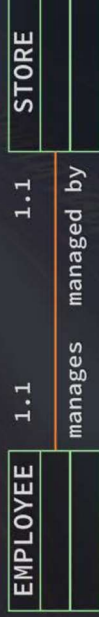
A **One-to-Many (1:M) Relationship**: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A **Many-to-Many (M:N) Relationship**: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A **One-to-One (1:1) Relationship**: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Slide 14: ER Diagram Notation

ER Diagram Notation

Chen Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Crow's Foot Notation



UML Class Diagram Notation

