Data Modeling

- A technique used in Database Design
- Like an architect's drawings of a house being built
  - The db architect draws the model of the database
  - The technicians (DBAs for example) then build the database according to the architect's design
- emphasizes on what data is needed and how it should be organized instead of what operations need to be performed.
- helps to visualize database structure and ensure default values, columns keys, constraints, indexes, triggers, etc.

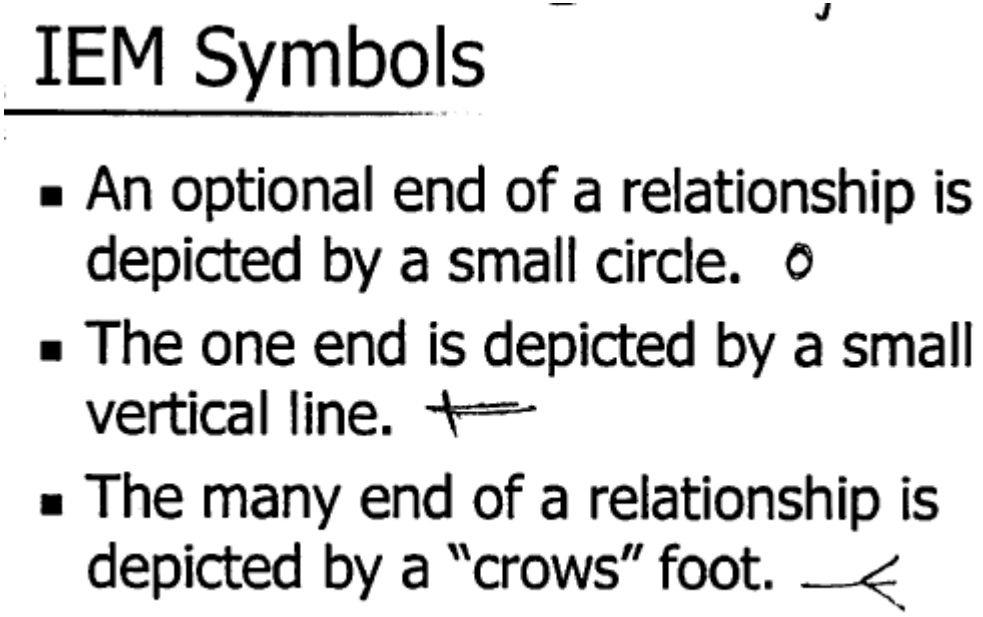
Data Modeling

- **Different techniques**

– iDEF1X

– Crows' Feet (IEM – Information Engineering Methodology)

- Chen



Data Modeling

- Types of DB Design

  Logical Database design

  – Implementation Independent

  – Establishes the entities, fields and relationships between fields

HOW the system should be implemented regardless of the DBMS.
--> typically created by Data Architects and Business System Analysts.
--> main purpose is to developed technical mapping of rules and data structures.

PhysicalDatabase design

– Refine the Logical design in terms of the constraints and characteristics of DBMS

HOW the system will be implemented using a specific DBMS system.
--> typically created by DBA and developers.
--> main purpose is the actual implementation of the database.

Data Modeling

Purpose of Design Types

- The main reason for having two separate stages in the design process is so that the designer can focus on two separate issues:
- In the Logical step, the focus is on the business process and requirements
- In the Physical step, the focus is on the technical requirements

Data Modeling

- Correct Database design is concerned with:
  - Avoidance of data redundancy.
  - Application performance.
  - Data Independence.
  - Data Security.
  - Application Development.

Data Modeling

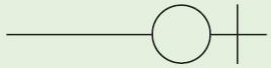
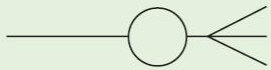
- Process
  1. Gather all data requirements
  2. Normalize the data to 3rd Normal Form
  3. From 3NF schemas, draw the Data Model (ERD)
  4. Review the ERD with your customers verifying it against the requirements
  5. Upon signoff, generate DDL

Data Modeling

- Symbols you will use in data modeling

Entity (rectangle) with an entity name, Primary Key, and Attributes listed

Cardinality & Optionality Symbols

Symbol	Meaning
	One—Mandatory
	Many—Mandatory
	One—Optional
	Many—Optional

Data Modeling

- Cardinality
  - How many of THESE are related to how many of THESE
  - Typically: zero, one, or many
  - On both ends of the relationship

- Optionality
  - Is the relationship **mandatory** (one or more) or **optional** (zero)

Data Modeling

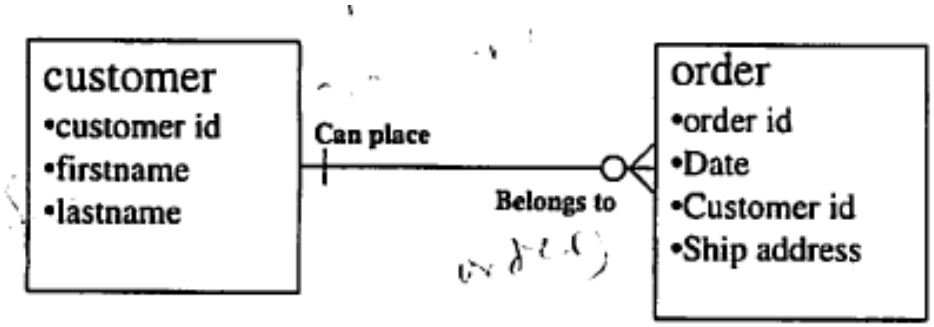
A Mandatory relationship is where there must be at least one matching record in each entity.

An Optional relationship is where there may or may not be a matching record in each entity.

e.g**. Customer and Order table:**

- Customer is the parent; customers can exist whether they buy a product or not. (Independent entity)
- Order, is the child record, Fk migrates from the Customer. It is mandatory that Order has a CustomerID. (Dependent entity)
- an order cannot exist without a customer.

Data Modeling



Data Modeling

- Prepare for Data Modeling
  - Once data requirements are clear, then
  - Decide the **Business Area** you are modeling
    - NOTE: Failure to restrict your model to a single business area will make the data modeling process much more complex
  - Organize all the data items into **ENTITIES** and **ATTRIBUTES**
  - Determine an attribute that can serve as a **PRIMARY KEY** for each entity
  - If no appropriate candidate keys exist, then plan to create a **SURROGATE** key

Data Modeling

Begin the process of Data Modeling

1. Draw a rectangle to represent each **ENTITY**
2. Write the **NAME** of the entity above the rectangle (entity names should be singular)
3. Draw a **RELATIONSHIP** line between each related entity
4. Draw **CARDINALITY** and **OPTIONALITY** symbols on both ends of each relationship line
5. Resolve any many-to-many relationships by creating an **ASSOCIATION** ("child") entity between the two "parent" entities

Data Modeling

1. Draw a horizontal line across each entity rectangle, and enter the name of the **primary key** attribute above the line
    1. NOTE: As you define primary keys and group the attributes within entities, you will **NORMALIZE** the data
2. Then list all the remaining **attributes** within the rectangle below the line
3. Identify any **foreign key** attributes with an "(FK)"
4. Walk through the model with your customers

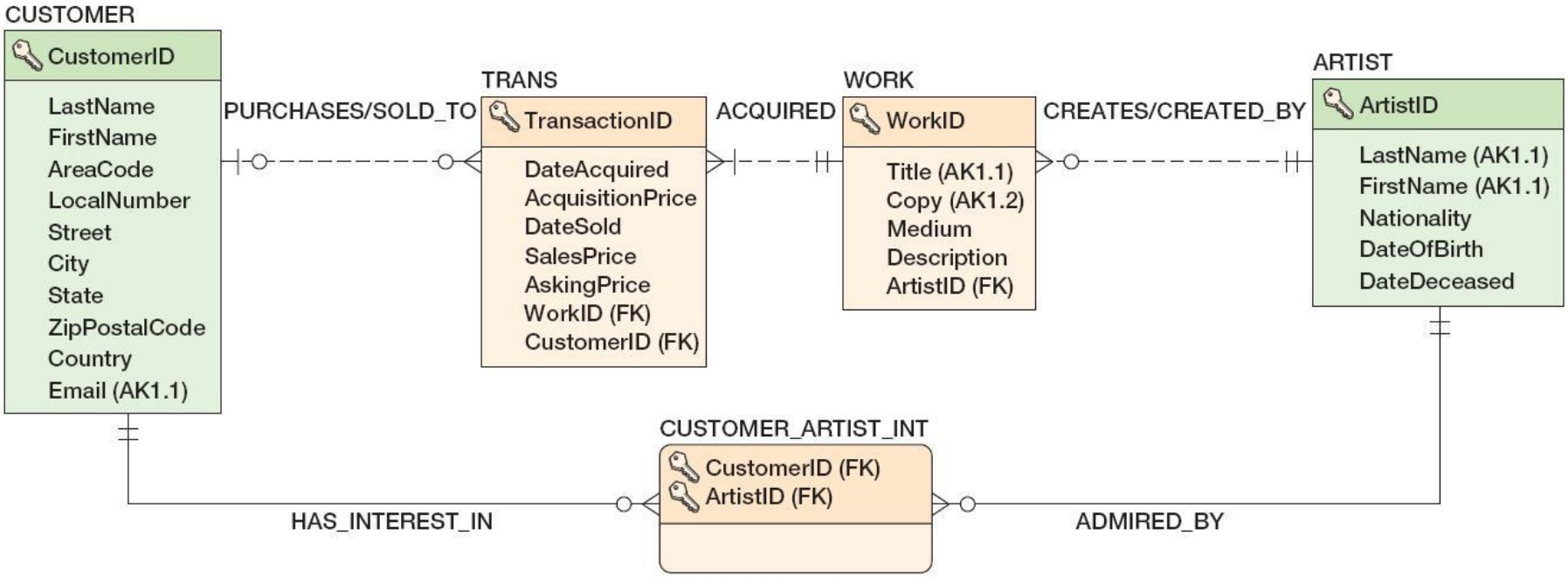
Data Modeling

"Reading the Relationships"

- An order belongs to one customer
- A customer can place zero, one or many orders
- An order may contain one or many OrderDetails
- A product may be purchased on zero, one or many OrderDetails
- Read relationship clockwise

Data Modeling

- An example:



Look at this data model, then consider the questions on the next slide.

Data Modeling

- What are the names of the five entities?
- What is the primary key of each entity?
- Which one is an association entity?
- Why are some of the relationship lines dashed, and some are solid?
- Why do 4 of the entities have square corners and one has rounded corners?
- Which entity has a composite (or “concatenated”) key?
- Relationship descriptions should be read clockwise: a customer purchases a work; a work is sold to a customer. Which entity represents the fact that a customer purchased a work?

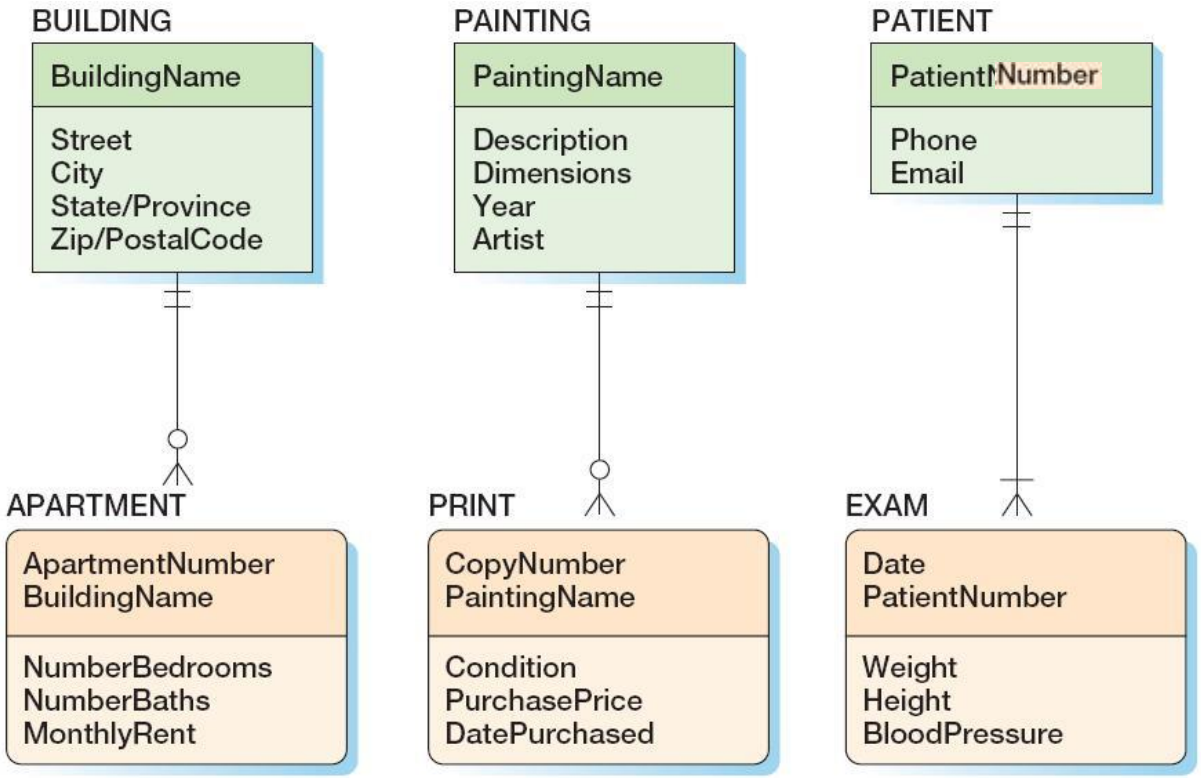
Data Modeling

More Model Constructs

- Square Edge = “Independent Entities”
- Round Edge = “Dependent Entities” (“weak”)
- Dashed Line = a dependent entity where the parent’s key does not migrate to primary
- Solid Line = a dependent entity where the parent’s key migrates to primary

Dependent Entities

- ID Dependent Entities - An ID-dependent entity is an entity whose identifier (key) includes the identifier of another entity.



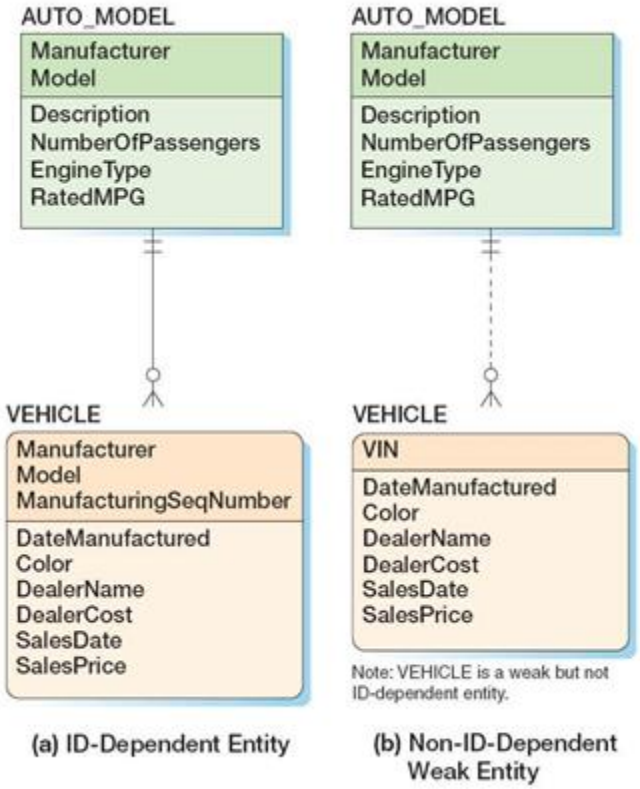
(a) APARTMENT is ID-Dependent on BUILDING

(b) PRINT is ID-Dependent on PAINTING

(c) EXAM is ID-Dependent on PATIENT

Dependent Entities

- Non-ID Dependent Weak Entities

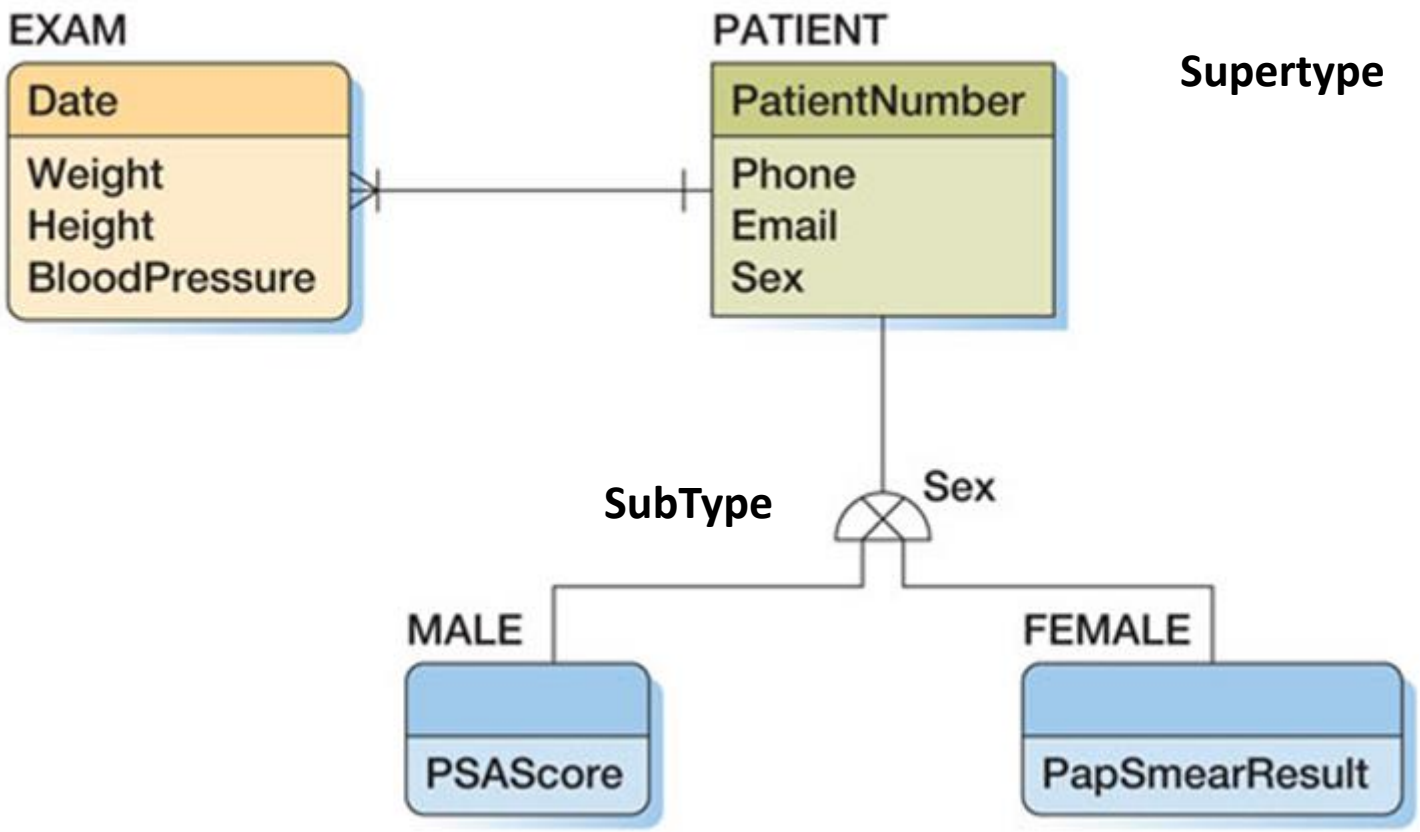


Data Modeling

More Model Constructs

- Subtype/Supertype
  - The supertype contains all common attributes, while the subtypes contain specific attributes.
  - The supertype may have a **discriminator** attribute which indicates the subtype.

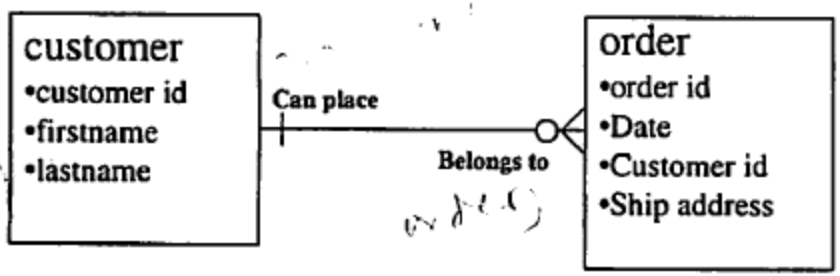
Data Modeling



What is Relationship?

- Relationships are associations between entities.
  - Generally a verb connecting two or more entities.
    - An Employee **has** an Address.
    - A Manager **is** an Employee.
    - A Manager **manages** a Department.

Relationship Example



Relationship Types

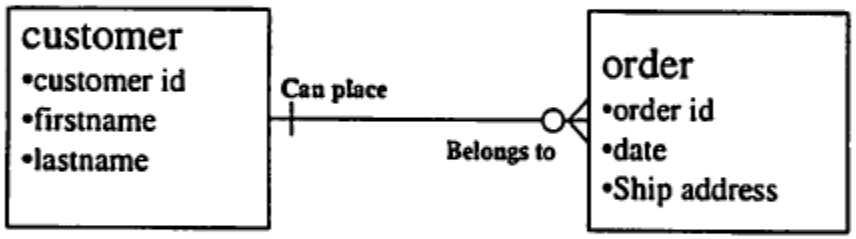
- One-to-One
  - Each record in entity A is related the only ONE record in entity B. Each record in entity B is related the only ONE record in entity A.
- One-to-Many
  - Each record in entity A is related to zero, one or more records in entity B. Each record in entity B is related to only ONE record in entity A.
- Many-to-Many
  - Each record in entity A is related to zero, one or more records in entity B. Each record in entity B is related to zero, one or more records in entity A.

One-To-One Relationship

Each employee would have ONE record in each table.

other examples, Driver and Driver licenses;
Person and Address Entities, etc.

One-To-Many Relationship



A customer may place one or more orders. Every order must belong to a customer.

Many-To-Many Relationship

An employee may have one or more skills. Each skill may belong to one or more employees.

Resolving a Many-to-Many Relationship

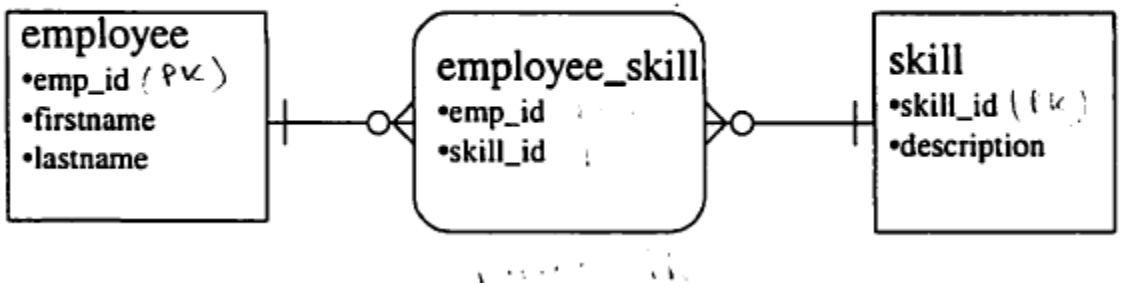
- What is it?
- Why do I have to Resolve it?
- How do I resolve it?

Resolving a Many-to-Many Relationship

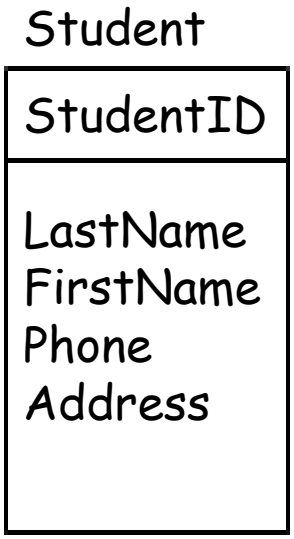
- Many to Many relationships can exist in the logical design but not in the physical design.
- To resolve a Many to Many relationship:
  - Create an intersection entity
  - Create a Unique ID for the new entity which consists of a combination of the unique IDs of the two original entities

Resolving Many to Many...

- This diagram illustrates that the employee entity and the skill entity both link to an entity called employee_skill



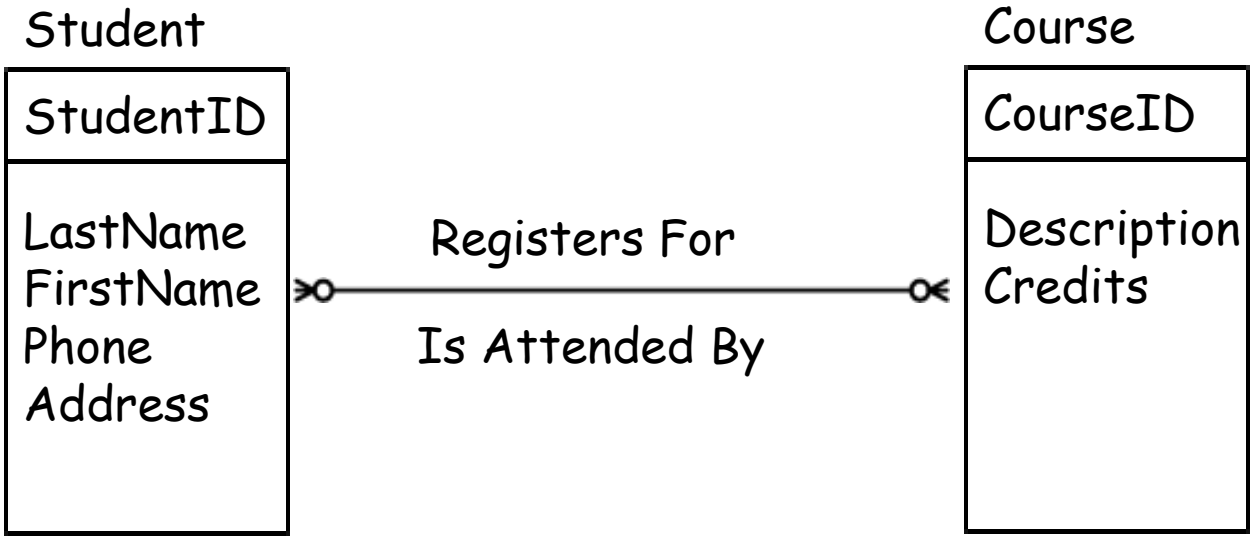
Resolving a Many-to-Many Relationship
Step by step




Student
StudentID
LastName
FirstName
Phone
Address
Course
CourseID
Description
Credits


Student
StudentID
LastName
FirstName
Phone
Address
Course
CourseID
Description
Credits


Student
StudentID
LastName
FirstName
Phone
Address
Registers For
Course
CourseID
Description
Credits


Student
StudentID
LastName
FirstName
Phone
Address
Registers For
Is Attended By
Course
CourseID
Description
Credits

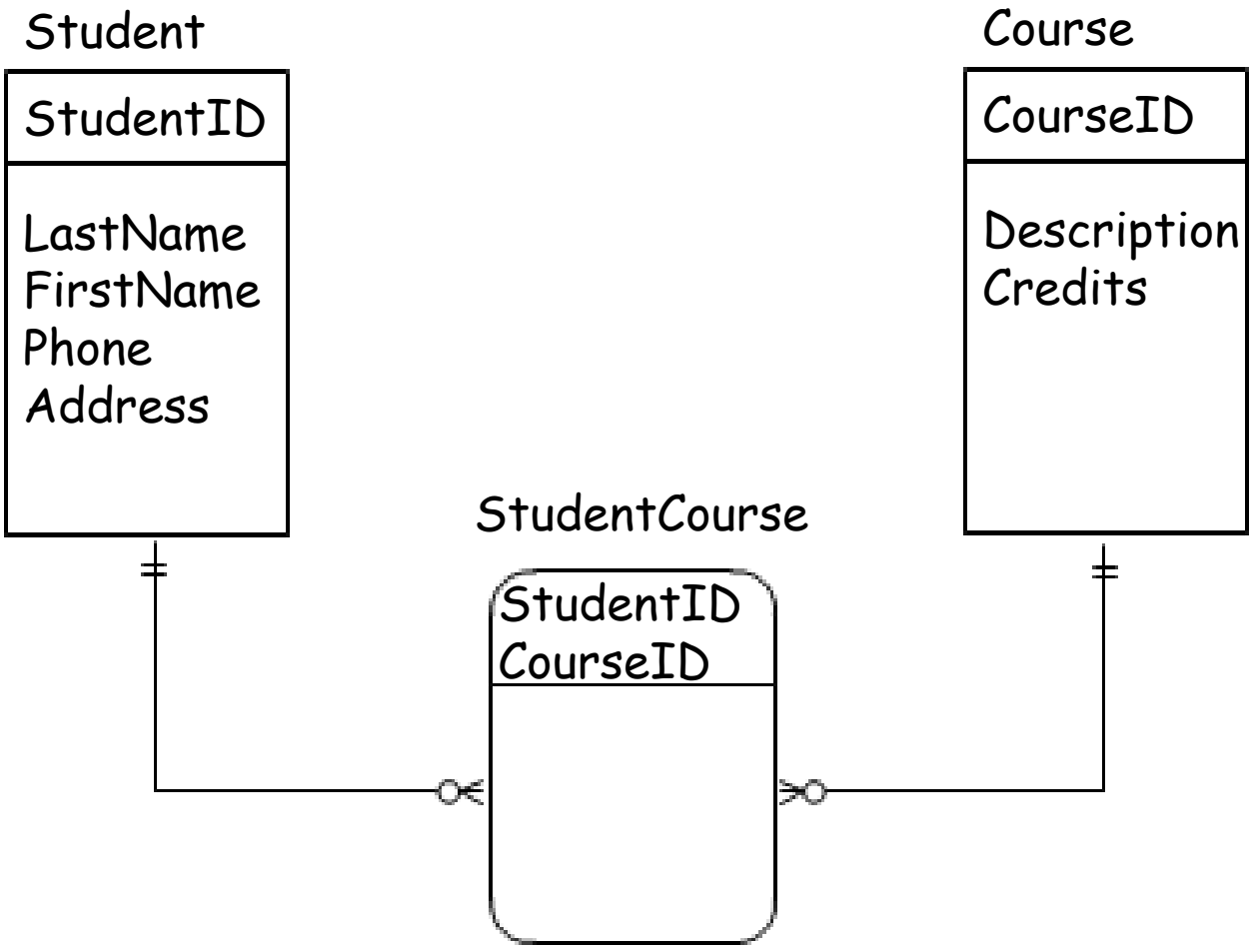
This is a Many-to-Many relationship. As such, it cannot be Implemented in a relational database. Why is that?

How would you set up the **foreign key relationships** necessary to link a student and his/her courses, or a course and its students?

You must follow the rules for **First Normal Form**. You cannot add multiple "CourseID" attributes to the "Student" entity. You cannot add multiple "StudentID" attributes to the "Course" entity. Doing so would violate **First Normal Form**.

How to resolve this?

Create an "**Association**" or "**Intersection**" Entity.


Student
StudentID
LastName
FirstName
Phone
Address
Course
CourseID
Description
Credits
StudentCourse
StudentID
CourseID

Association Entity

- Created to resolve a Many-to-Many relationship
- A dependent Child between two independent Parents
- Relationship to parents is mandatory
- A composite primary key, a portion inherited from each parent
- Rounded corners indicate dependency

Data Modeling Tools

Drawing your data models

- Reasonable Free **Tools**:
  - MySQLWorkbench
  - Lucidchart (free trial)
  - TOADModeler (free trial)
  - Oracle SQL DEVELOPER
  - ErWin

Data Modeling Software

- Drawing a Model
  - Entities, attributes, relationships, cardinality, optionality
  - Physical characteristics like data type & length
  - Constraints like PK, FK, Nullable

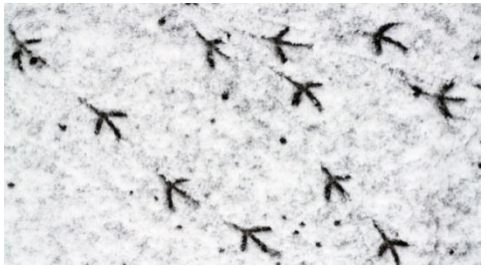
- Different tools support different capabilities
  - Some can only DRAW
  - Some capture the intelligence behind the drawing
  - Some can generate SQL DDL
    - ***Forward and reverse engineering***

Data Modeling Software

- Different tools have different licensing models
  - Industrial Strength = Expensive
    - PowerDesigner from SAP
    - ERWin
    - ER/Studio from Idera
    - SPARX Enterprise Architect
    - IBM InfoSphere
    - TOAD Data Modeler
    - Visio Professional
- Let's take a look
  - https://www.datasciencecentral.com/profiles/blogs/top-6-data-modeling-tools
  - http://www.databaseanswers.org/modelling_tools.htm

Data Modeling Software

- Different tools have different licensing
  - Free = less capable (mostly)
    - LucidChart – lovely drawings, but no intelligence
  - Free Trial – 15 or 30 days, then you pay
- Other concerns
  - Windows, Mac, Linux
  - Local install or cloud
  - Drawing Technique: CROWS FOOT (IE) versus IDEF1X versus UML
  - Data Modeling versus full blown Database Administration Console versus SQL Editor



Data Modeling Software

Drawing your data models

- Reasonable Free Tools:
  - MySQLWorkbench Free, local install
    - Full capabilities
  - LucidChart – Free Trial, cloud
    - Drawing only, limited time
  - TOAD Modeler – Free Trial
    - Full capabilities, limited time, local install
  - Oracle SQL Developer – Free, local executable
    - Full capabilities (* but you need a database connection)
  - Astah ? Good reviews, free to students

Data Modeling Software

Drawing your data models

- Recommendation:
  - MySQL Workbench Free, local install
    - Full capabilities (for MySQL)

Let's download .... attachments are in canvas.

Link:

https://www.mysql.com/products/workbench/

Data Modeling Software

MySQL Workbench Prerequisites

To be able to install and run MySQL Workbench on Windows your system needs to have libraries listed below installed. The listed items are provided as links to the corresponding download pages where you can fetch the necessary files.

- Microsoft .NET Framework 4.5
- Visual C++ Redistributable for Visual Studio 2015

After MySQL workbench download:

Go to Edit --> Preferences.

--> SQL Editor and uncheck "Safe Updates" check box.

--> Reconnect to Server

--> Modeling --> Diagram --> check Show captions // logout and then login.

Data Modeling Software

Video overview of how to download and install MySQL and MySQL Workbench on your MAC OS and Windows computer.

How To Install MySQL on Mac OS
https://youtu.be/UcpHkYfWarM

How to Install MySQL Workbench on Windows 10
https://youtu.be/Z0ZcCmt7pd0

You can also search in YouTube or Google it, you may be able to find better resources.

Data Modeling Software

Step-By-Step

1. Draw each TABLE
2. Double-Click to bring up COLUMN dialog
3. Add each column, data type, constraints
4. Create relationships
   - Click on Child Key, then on Parent Key
   - This duplicates keys in the model
   - Modify cardinality / optionality, if needed.
5. Export DDL

   Database - Forward Engineer (creates DDL)

   - Reverse Engineer (created DataModel)

   Or, File - Import / Export

- 11.1: When a model is exported using the main menu item **File**, **Export**, **Forward Engineer SQL CREATE Script**, some server variables are temporarily set to enable faster SQL import by the server. The statements added at the start of the code are:

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';
```

These statements function as follows:

- `SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;` : Determines if an InnoDB engine performs duplicate key checks. Import is much faster for large data sets if this check is not performed.
- `SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;` : Determines if the server should check that a referenced table exists when defining a foreign key. Due to potential circular references, this check must be turned off for the duration of the import, to allow defining foreign keys.
- `SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';` : Sets `SQL_MODE` to `TRADITIONAL`, causing the server to operate in a more restrictive mode.

These server variables are then reset at the end of the script using the following statements:

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Data Modeling

Let's Practice using the schemas from the chair company.

EZ Chair Company

UNNORMALIZED	FIRST NORMAL FORM	SECOND NORMAL FORM	THIRD NORMAL FORM
Customer Order	Customer Order	Customer Order	Order
Order Number	Order Number	Order Number	Order Number
Order Date	Order Date	Order Date	Order Date
Delivery Date	Delivery Date	Delivery Date	Delivery Date
Customer Discount	Customer Discount	Customer Discount	discount amount
discount amount	discount amount	discount amount	invoiced amount
invoiced amount	invoiced amount	invoiced amount	customer number
customer number	customer number	customer number	order total
customer name	customer name	customer name	
Contact	Contact	Contact	Customer
ContactType	ContactType	ContactType	customer number
bill to address	bill to address	bill to address	customer name
bill to city	bill to city	bill to city	Contact
bill to state	bill to state	bill to state	ContactType
bill to zip	bill to zip	bill to zip	bill to address
ship to address	ship to address	ship to address	bill to city
ship to city	ship to city	ship to city	bill to state
ship to state	ship to state	ship to state	bill to zip
ship to zip	ship to zip	ship to zip	ship to address
--- Product Number	order total	order total	ship to city
| Description			ship to state
| quantity ordered	OrderDeail	OrderDeail	ship to zip
--- unit price	Order number	Order number	
order total	Product Number	Product Number	OrderDeail
	Product Description	Quantity	Order number
	Quantity	total	Product Number
	unit price		Quantity
		Product	total
		Product Number	
		Product Description	Product
		unit price	Product Number
			Product Description
			unit price

Data Modeling Software

This is a link to a YouTube video recording of the class demonstration of MySQL Workbench data modeling software from class on Monday, February 12, 2018.

https://youtu.be/W66W-1zJtm8

Or,

https://www.youtube.com/watch?v=W66W-1zJtm8&feature=youtu.be

You can also search in YouTube or Google it, you may be able to find better resources.

Practice creating Tables and DDL using MySQL Workbench

Project Code	Project Title	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Pensions System	M Phillips	24500	S10001	A Smith	L004	IT	22.00
PC010	Pensions System	M Phillips	24500	S10030	L Jones	L023	Pensions	18.50
PC010	Pensions System	M Phillips	24500	S21010	P Lewis	L004	IT	21.00
PC045	Salaries System	H Martin	17400	S10010	B Jones	L004	IT	21.75
PC045	Salaries System	H Martin	17400	S10001	A Smith	L004	IT	18.00
PC045	Salaries System	H Martin	17400	S31002	T Gilbert	L028	Database	25.50
PC045	Salaries System	H Martin	17400	S13210	W Richards	L008	Salary	17.00
PC064	HR System	K Lewis	12250	S31002	T Gilbert	L028	Database	23.25
PC064	HR System	K Lewis	12250	S21010	P Lewis	L004	IT	17.50
PC064	HR System	K Lewis	12250	S10034	B James	L009	HR	16.50

project (project_code, project_title, project_manager, project_budget)

project employee (project_code, employee_no, Hourly_rate)

employee (Employee_no, Employee_name, Department_No)

Department (Department_no, Department_name)