

Logols Learning

WEEKEND WEB DEVELOPMENT BOOT CAMP

TRAINING: DATA MODELING

Entities

► Types of data

- Exist physically or logically
- Think of nouns

► Physical Ex.

- Customer
- Employee

► Logical Ex.

- Transaction
- Bill of Materials

► Shape: Rectangle

Entity

- Entities are the different types of data that you are trying to map.
- It is anything that you want to hold data for.
- They are things that exist either physically or logically.
- They can be thought of as nouns normally – person, place, or thing
- By existing physically I mean a physical object or person like a table, a chair, a customer, an employee, a product, a user
- By existing logically I mean any thing that exists that is not physical.
 - Examples include transactions, bill of materials, or different types of events like text message received or server restarted.
- An entity is represented as a rectangle with the name of the entity written in the middle of the shape.

Attributes

- ▶ Fields within an Entity ▶ Shape: Oval
- ▶ Ex. A Person Entity has the following attributes:
 - ▶ Name
 - ▶ Phone Number
 - ▶ Address
 - ▶ Social Security Number



Attribute

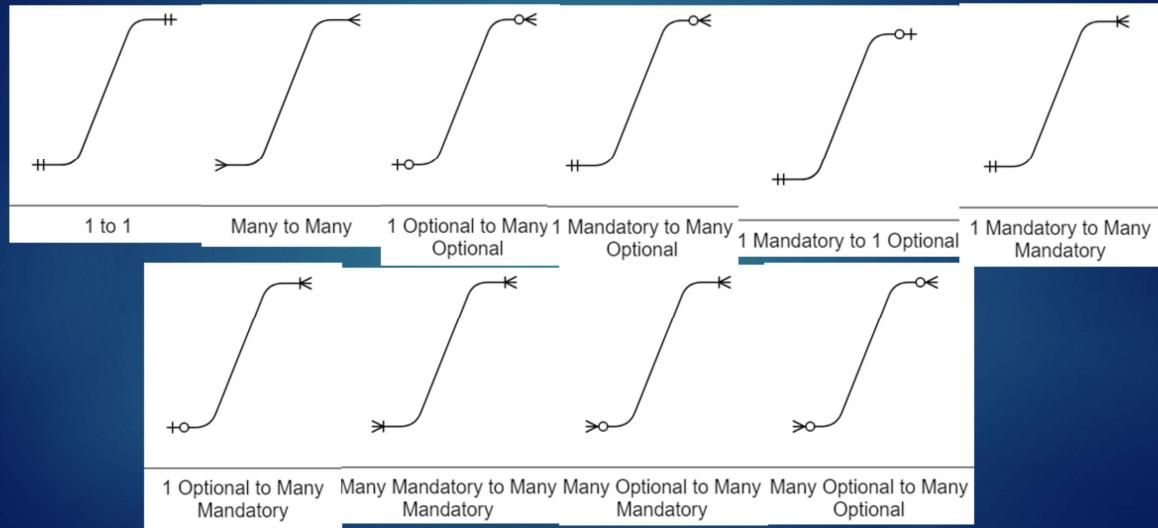
- Attributes are fields or pieces of data within an entity.
- They are represented by an oval shape
- Examples would be name, phone number, address, and social security

Relationships

- ▶ Which entities are related
- ▶ Think of verbs
- ▶ How entities are related - Cardinality
 - ▶ One to One
 - ▶ One to Many
 - ▶ Many to Many
- ▶ Shape: Line
- ▶ Has different notations at the ends of the lines
- ▶ Describe how entities are related

- Relationships show which entities are related.
- These are usually verbs, like a person owns a house or a manager manages an employee
- The relationships also show how entities are related. Possible options are:
 - One to One
 - One to Many
 - Many to Many
- An example of a one to one relationship is, one employee has one desk
- An example of a one to many relationship is, one manager has many employees
- An example of a many to many relationship is, many employees work on many projects.
- This is called the cardinality of the relationship
- The relationship defines the minimum and maximum cardinality of the relationship
- The minimum can be as low as 0
- The maximum can be as high as many – which means infinite.
- The only other option is 1

Crowfoot Notation



- These are different permutations of the crowfoot notation.
- As you can see there are two parts
- The end of the line shows that max number and next to it there is a definition of the min number
- The options are 0, one, or many
- The max number has to at least be one.
- This is shown for each side of the relation.

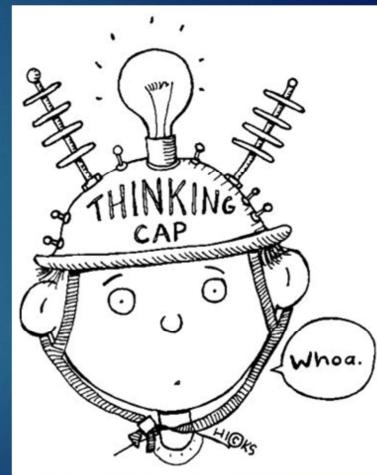


EXAMPLE

ENTITIES & RELATIONSHIPS

ASSESSMENT

ENTITIES & RELATIONSHIPS



- What shape is used to model Entities?
- What shape is used to model attributes of an entity?
- What are the 3 different ways that entities are related?
- What are the symbols that represent the 3 ways that entities are related?



TEAM PROJECT

ENTITIES AND RELATIONSHIPS

Normalization

- ▶ Unbundles overlapping entities
- ▶ Data Integrity
 - ▶ Minimize Duplication of data
- ▶ Referential Integrity
 - ▶ Make a change only in one place
- ▶ Keyed Data Access
 - ▶ Access and manipulate data quickly
- ▶ Avoid Anomalies
 - ▶ Insert, Update, Delete

- Normalization unbundles overlapping entities, meaning entities with the same attributes associated with them.
- This is important for a few reasons.
 - First we want to maintain data integrity.
 - This means minimizing the duplication of data.
 - Second we want to maintain referential integrity.
 - With this we want to ensure all updates are done in one place.
 - Third we want keyed data access.
 - This allows for quick data selection and updates
 - Overall, we are trying to remove anomalies
 - These could be insert, update, and delete anomalies.
 - An insert anomaly would occur if you are not able to insert data that needs to be inserted because a dependency on other data does not exist yet.
 - An update anomaly occurs if the same data exists in multiple locations and only one of those locations is updated.
 - A delete anomaly would occur if you have to delete data you don't want to delete in order to delete data you do want to delete.
- To normalize is to go through a number of steps.
- After each step you end at a different normal form.

- We are only go through third normal form even though there are further forms.

First Normal Form

- ▶ Ensure Data is Atomic
 - ▶ Having no repeating groups (array of the same value)
- ▶ Attribute cannot hold multiple values.
- ▶ Define a primary or candidate key

- First normal form is ensuring there are no repeating groups and an attribute holds only one value.
- For example address includes the number, street name, city, state and zip code.
- We call of this the address, but it can all be broken down as well.
- So we could put the number and street name in a separate column, city in a separate column, state in a separate column and zip in a separate column
- We also want to define a primary key.



EXAMPLE

FIRST NORMAL FORM

Second Normal Form

- ▶ Table is in 1NF
- ▶ No non-prime attribute is dependent on the proper subset of any candidate key of table.
- ▶ An attribute that is not part of any candidate key is known as a non-prime attribute
- ▶ In other words remove functional dependencies

- Wow that sounds confusing doesn't it?
- It really just means that we need to look for any other subsets of data that are dependent on only part of the candidate key.
- For example you have been given data about a family that contains their names, ages, gender, address, and classes that they signed up for at the Local YMCA that includes the name, room, and time of the class.
- The candidate key would likely be made up of the persons social security number or person id along with the class id.
- The class info is only partially dependent on the full key of the person id and class id because it can live on its own only being dependent on the class id.
- So we would break this out as a separate class entity.

EXAMPLE

SECOND NORMAL FORM

Third Normal Form

- ▶ Table is in 2NF
- ▶ Transitive functional dependency of non-prime attribute on any super key should be removed.
- ▶ A transitive dependency is a dependency between two nonkey attributes

- This is very similar to what we did with second normal form.
- The difference here is that transitive dependencies are dependent on non key attributes
- From our previous example about our family data
- They all live at the same house so if we put the data in a table like this, the address would be repeated multiple times and cause anomalies.
- Second normal form tells us to take address and create a new entity out of it because it can live on its own with its own candidate key.

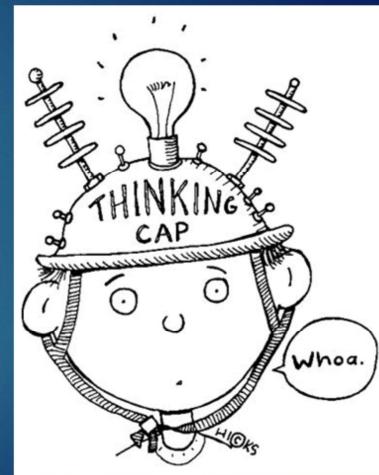


EXAMPLE

THIRD NORMAL FORM

ASSESSMENT

NORMALIZATION



- Name at least two reasons for normalization.
- How many values should an attribute hold?
- What should be done if columns are dependent on an attribute that is not or only part of the candidate key?



TEAM PROJECT

NORMALIZATION

Relational Database Terminology

- ▶ Table - Relation
 - ▶ Column – Relation Header Attribute
 - ▶ Row – Relation Body
- ▶ Constraint – Predicate on a column
- ▶ Primary Key – Uniquely Defines the table
- ▶ Foreign Key – Relates to another table
- ▶ Index – Provides quicker data access on one or a set of columns
- ▶ SQL – Structured Query Language – Query

- A table is for the most part a relation. It is the definition of what can be stored.
 - A table has columns which are the attributes that define the parts of a table like first name, last name, social security, etc. for an employee table
 - A table also has rows, which is the data itself, like John Smith, Tom Jones for an employee
- A constraint is a predicate that can be applied to a column.
- A Primary key is one or multiple columns that uniquely defines each row in the table. This cannot be repeated in any other row of the table.
 - For an employee table this might be the social security number or an employee id. We will discuss best practices on this later.
- A foreign key is a column or multiple columns that relate to a column or columns in another table to relate data between tables.
 - For example we may have an employee table with an employee id and we may have a company table with a company id. To relate these two tables together. We may add the company id to the employee table. The company id in the employee table would be considered the foreign key because it is the column that exists in both tables and links the 2 tables together.
- An index is a definition for one or multiple columns for the database to create a tree structure so data can easily be search by the one or set of columns.
- SQL stands for structure query language. It is a standard language that is used between

different relational databases.

- The language is used to insert, update, delete, or select data as well as to define the structure of tables.
- The language is standard across different relational databases, but most relational databases have their own flavor of the language that can only be used with that relational database. For example, Microsoft Sql Server has T-Sql and Oracle has PL-SQL. They are based on standard ANSI sql but are different.
- Standard ANSI sql is recommended to be used.

Primary Keys

- ▶ Primary Key
 - ▶ Uniquely defines a record in a table
 - ▶ Types:
 - ▶ Composite
 - ▶ Surrogate
- ▶ Foreign Key
 - ▶ Columns that uniquely relate to another table

- A primary key uniquely defines a record in a table
- Table searches perform a whole lot better when there are primary keys / clustered indexes on them.
- There can only be one on every table.
- Once the primary key is placed on the table the database management system can use that to form a tree, which makes operations faster for the database to perform.
- Multiple columns can make up a primary key – this is called a composite key
 - For example, we have a class at our local YMCA could be defined with a primary key of the class name, teacher, room, and time
- This is quite long though, we normally don't want our keys to be more than 3 columns, so instead we might make up a unique id called ClassId. This would be called a surrogate key because we are using it in place of the composite key.
 - I generally use surrogate keys for almost every table. This is fairly common practice.
- A foreign key is used to relate one table to another.
- Usually this is done by placing the primary key of one table into another table as a foreign key.
- For example We have a class table with a primary key of classId and a teacher table with a primary key or teacherId.
 - Because this is a one to many relationship with one teacher having one to many

classes we will add the teacherId to the class table as a foreign key.

- When you add a foreign key you have to option to enforce constraints and cascade updates and deletes
- Enforcing constraints is generally suggested as you want to ensure that what is being inserted into the foreign key column does exist in the table that it is related to.
- Cascading updates and deletes can be dangerous because data could be lost unexpectedly.
 - Many systems prefer to do soft deletes because of the possible loss of data so cascading deletes become less important
 - Also, many systems use surrogate keys and prefer to soft delete a record than update a primary key which can make cascading updates less important.



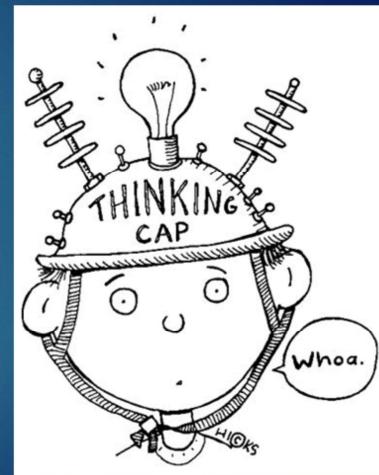
EXAMPLE

PHYSICAL DATA MODEL

- Let's create a physical data model for our assessment system from our logical data model

ASSESSMENT

PHYSICAL DATA MODEL



- Data is stored in a what?
- What uniquely defines the table?
- What relates to another table?
- Create a physical data model for a teacher, class, and students



TEAM PROJECT

PHYSICAL DATA MODEL

QUICK REVIEW

DATA MODELING



- What are the 3 different ways that entities are related?
- Name at least two reasons for normalization.
- How many values should an attribute hold?
- What uniquely defines the table?
- What relates to another table?
- Create a physical data model for a teacher, class, and students

Additional Resources

- ▶ Data Modeling 101
 - ▶ <http://www.agiledata.org/essays/dataModeling101.htm>
- ▶ Normalization Videos:
 - ▶ <https://www.youtube.com/watch?v=NScuEk7CSNo>
 - ▶ https://www.youtube.com/watch?v=0suZ8H_bDgY&t=487s