9.3 Lesson Summary - Data Modeling

The data that we have worked with up to this point in the course has been structured simply enough that it can be expressed in a flat table of rows and columns. There are many real-world scenarios that require a more complex data structure. How all of your data is organized is referred to as the data model. A simple data example doesn't require much attention to the data model. If we want to store information about a group of people, their age, and if they like ice cream or not we can imagine this as a table with 3 columns; a name, age, and likes\_ice\_cream column. If we want to record more information about these people our data model can get more complex quite quickly. What happens if we want to record an address for our people? What if we want to be able to have more than one address per person? What happens if we do not have an address for a person? To properly answer these questions, we need to think about how our data should be modeled. When you are collecting and organizing data it is often important to think about your data model, and how to best organize your databases.

Concept: **Data Normalization** is the process of reorganizing your data to reduce duplicates, reduce large sections of empty fields, and organize your data more efficiently. Normalizing your data often involves moving it from a single table into multiple related tables.

* Activity: 01-Ins\_Data\_Normalization

Concept: There are different levels of **Data Normalization**. In **first normal form** each row in a table should contain only a single value and each row should be unique. **Second normal form** requires your data to be in first normal form, you must have a single column for your primary key, and each table is uniquely identified. Third normal form has data organized in second normal form and contains no transitively dependent columns.

* Activity: 01-Ins\_Data\_Normalization, 02-Stu\_Data\_Normalization

Concept: **Foreign Keys** are the way in which one row in a table can reference another row in another table. This feature is what makes relational databases relational. If we had a table of people and each person had one address stored in an address table, we could create a foreign key for our people table using the following code: Before you declare a Foreign Key in your table the table you're referencing with that Foreign Key should be created.

*CREATE TABLE people (*

*id SERIAL PRIMARY KEY,*

*name VARCHAR(30) NOT NULL,*

*age INTEGER NOT NULL,*

*likes\_ice\_cream BOOLEAN NOT NULL,*

*address\_id INTEGER NOT NULL,*

*FOREIGN KEY (address\_id) REFERENCES addresses(id)*

*);*

* Activity: 03-Ins\_Foreign\_Keys, 04-Stu\_Foreign\_Keys

Concept: The optimal way to model your data is usually tied to the frequency of relationships between data. A **one-to-one** relationship exists where one data parameter will only correspond to a single other data parameter. For example, one person has one age. A **one-to-many** relationship exists where one data parameter corresponds to many of a different kind of data parameter. For example, one person may have multiple email addresses. A **many-to-many** relationship exists when many of the same kind of data parameter correspond to many of a different kind of data parameter. For example, one person will have multiple parents and their siblings will have the same multiple parents. Many children have many parents.

* Activity: 05-Ins\_Data\_Relationships, 06-Stu\_Data\_Relationships

Concept: Entity Relationship Diagrams (ERD) are a way to visually illustrate the relationship between tables and their fields in a database.

* Activity: 07-Ins\_ERD, 08-Par\_Designing\_ERD, 09-Par\_ERD
* Suppl link: <https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>

Concept: In SQL a **UNION** is used to combine the results of multiple queries. Where a JOIN will combine elements from different tables into a single row a UNION will simply stack the results on top of each other. An example of a union query is:

*SELECT COUNT(\*)*

*FROM city*

*UNION*

*SELECT COUNT(\*)*

*FROM country;*

* Activity: 10-Ins\_Unions, 11-Stu\_Unions
* Suppl link: https://www.w3schools.com/sql/sql\_union.asp