# Querying Relational Databases

## Database Normalization

### Why We Make Databases Relational

Benefits:

* Maximizing storage
* Better application functionality
* Cleaner, richer data for business reporting

### Database Normalization

DN: separate data into tables.

### How Normalization Helps Us

#### Quiz

* Relational databases:
  + Tables are linked, or “related,” via common fields.
  + First conceptualized in 1970s
  + Eliminate data modification (update) anomalies and increase data integrity
* All CRUD operations benefit from a well-normalized db design.
* Normalization: process of designing a relational db

### Set Theory and Relational Databases

Set Theory:

* Founded in 1874, and is a relatively new mathematical discipline, as compared to Algebra or Calculus.
* Depict as Venn diagrams (circles)

Set operations:

* Intersection: where 2 circles overlap
* Union: all elements in both circles
* Except: area outside intersection; no overlap

Result set: when you query a table, you get the whole set or subset of data back.

#### Quiz

* Set: a collection of things with common properties
* Venn Diagram: represents a set

## Database Keys

### Unique Keys

Unique Key:

* Column that doesn’t allow duplicates
* Can be NULL
* Multiple unique keys per table
* Can be modified to a new value

### Primary Keys

PK:

* Unique key, but never a NULL value
* One primary key per table
* Cannot be modified to a new value
* Can be text or integer

#### Quiz

* Database keys
  + Cannot guarantee a table does not return data when queried unless a specific password is supplied.
  + Ensure a value doesn’t repeat in a given column
  + Act as a point or link back to another table
  + Guarantee an entire row is unique within the table
* When a unique constraint is violated, the db doesn’t allow the data to be written to the table and an error is returned.
* A primary key will allow more than one NULL value.

### Foreign Keys

FK:

* Reference from one table to another
* Columns that contain data that relate to the PK in another table
* Adding FKs:
  + Columns designed correctly
  + FK constraint: a rule the db must enforce

Referential Integrity: FK values must also exist as PK values in reference table.

#### Quiz

* Once a Foreign Key Constraint is created to enforce the referential integrity between two tables, if you try to insert a value that does NOT exist in the primary key table it will not insert the data and will return an error.
* FK constraint: defines the physical relationship between 2 tables in a db.

### Table Relationships

### One to Many

Three types of table relationships:

1. 1 to 1
2. 1 to many (most common)
   1. FK always goes on many side
3. Many to many

### Many to Many

Many to many:

* These don’t actually exist
* What replaces them are two 1-to-many relationships with a junction table, or associative entity

### One to One

One to one:

* These are also not common; usually, you combine the two tables on the one PK.
* Possible scenarios:
  + Boost performance with tables with lots of infrequently used columns.
  + Working with 3rd party databases

### Modeling Table Relationships

Entity Relationship Diagrams (ERDs): during the design process, tables are referred to as entities.

Crow’s Foot Notation: many side of connector.