Database Migrations - Zach Middleton

Resource: https://docs.djangoproject.com/en/4.2/topics/migrations/

Migrations are the method that Django uses to propagate model changes into a database schema. Migrations can typically be created automatically by Django, but more complex changes will require a manual migration. Similar to Git, migrations act as a version control system for database schema that stores model changes in a file, making it easy to apply those changes to the real database. Django supports having dependencies for migrations, but these dependencies can lead to compatibility issues between various apps. Migrations are written as python objects, with a dedicated class layout that denotes it to be a migration. All reversible migration operations can be undone using the migrate command and the migration number. Database data can also be updated using migrations, but Django cannot automatically make migration files for data, so the migration files have to be written manually. The number of migrations for a given app can be reduced using a process called squashing, which looks for places where migrations can be combined into a single migration. In order to create migrations Django has to serialize models; most things can be serialized, but more complex things like nested classes cannot be serialized. Custom serializers can be written to handle what Django couldn’t on its own, and a deconstruct() method can be used to serialize custom class instances.