Introduction to Database Migrations

Database Migrations

Problems to solve:

No documentation

No record of database changes

Hard to collaborate across teams

Hard to rollback (undo) a change

Schema Migrations

Modify the **structure** of a database including table and column properties

Examples:

Create or delete a table, column, or constraint

Alter property of column e.g. name, datatype

Rename table, column, or constraint

Any changes made to an Entity-Relationship Diagram

Data Migrations

Transform existing data values

May support new business requirements

May allow for compatibility with other database systems

Data Migrations

Example: Capitalize Values

state_name
alabama
alaska
arizona
arkansas
california

state_name
Alabama
Alaska
Arizona
Arkansas
California

Data Migrations

Example: Reformat Values

phone_number
+1 (123) 456 - 7890
+1 (123) 456 - 9870
+1 (123) 654 - 7890
+1 (321) 456 - 7890
+1 (123) 456 - 7777

phone_number
11234567890
11234569870
11236547890
13214567890
11234567777

Data Migrations

Example: Map Values

timezone
America/Detroit
America/New_York
America/Anchorage
America/Chicago
America/Louisville

timezone
US/Eastern
US/Eastern
US/Alaska
US/Central
US/Eastern

Migrations

Best practices:

Incremental

Reversible

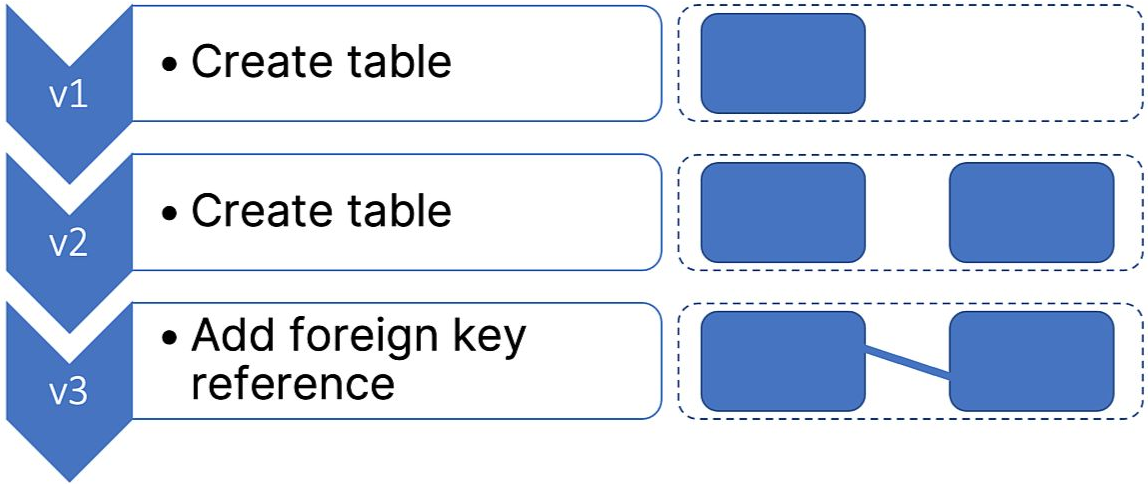
Programmatic (in forward and reverse)

Incremental

New *version* of database at each increment

Each increment is valid, working configuration

Similar in concept to Git version control system file change tracking



Reversible

Changes can be rolled back to previous version

To undo schema migration, invert the SQL query

Database backups may be useful