# Modifying Data with SQL

## Adding Data to a Database

### Adding a Row to a Table

Format:

INSERT INTO <tablename> VALUES(<values>);

If you follow the exact order for the columns, then you don’t need to specify them.

INSERT INTO loans VALUES(12, “Bob”);

With auto-increment, you can enter NULL for the ID value.

INSERT INTO loans VALUES(NULL, “Bob”);

Better to use column names:

INSERT INTO *<tablename>*(*<columns>*) VALUES(*<values>*);

No need to specify NULL values.

### Adding Multiple Rows to a Table

Database seed files: seed info. into db.

Separate sets of values by commas:

INSERT INTO *<tablename>*(*<columns>*)

VALUES(*<values>*),

(*<values>*),

(*<values>*);

## Updating Data in a Database

### Updating All Rows in a Table

Format:

UPDATE *<tablename>* SET *<columnname>* = *<value>*;

Assign multiple values:

UPDATE *<tablename>*

SET *<columnname1>* = *<value1>*,

*<columnname2>* = *<value2>*;

### Updating Specific Rows

Format:

UPDATE *<tablename>* SET *<columnname>* = *<value>* WHERE *<conditions>*;

## Deleting Data from a Database

### Removing Data from All Rows in a Table

Format:

DELETE FROM *<tablename>*

### Removing Specific Rows

Format:

DELETE FROM *<tablename>* WHERE *<conditions>*

#### Quiz

* To create rows in a table: **INSERT INTO...**

## Handling Errors When Manipulating Data

### Introduction to Transactions

Terms:

* Autocommit - every statement you write gets saved to disk.
* Seeding - populating a database for the first time.
* Script file - a file containing SQL statements.

Turn off autocommit:

BEGIN TRANSACTION;

Or just:

BEGIN;

Save and turn autocommit on again:

COMMIT;

### Rolling Back from Transactions

Format:

ROLLBACK;

#### Quiz

* If your computer crashes in the middle of a transaction, no changes are made. The database will be in the same state as it was before the transaction.

### Databases with Frameworks

ORM: Object-Relational Mapping

Benefits:

* Handles transactions
* One language; no need to switch to SQL commands

Examples:

* Hibernate (Java)
* CoreData (Objective C, Swift)
* Django (Python)
* ActiveRecord (Ruby)

DML: Database Manipulation Language

#### Quiz

* Doesn’t need an asterisk:

DELETE \* FROM sports\_teams;

* Transaction: a safe way to perform multiple transactions in one go.