Lesson 7

* Relational Databases
  + Stores the data used in an application. Imagine a social media web app, such as Facebook. There is a lot of data to make that application work. All the user info, posts and comments, ads, and everything else you can think of is data and needs a place to live. That place is a database.
  + Every time you create a new post, comment, add a friend, or take any action on the webapp, that data has to be remembered somehow, and that is accomplished by saving it in a database. The backend of an application can then receive requests from the front end, pull the needed data from the database to fulfill the request, and send it back to the front end.
  + You can visualize a relational database like a bunch of spreadsheets that are all connected to each other because of related data. For example, you could have one table with contact information for a user and another table with all the posts users make, and those two tables could be related because they have different pieces of data that describe a single user. We will talk more about the relationship side of database later, but for now just know that a relational database has a bunch of tables that contain related data.
  + The relational databases we will be using is MySQL.
    - Install at: <https://dev.mysql.com/downloads/mysql/>
      * Remember the username and password you set up
    - Install test database: <https://github.com/datacharmer/test_db>
      * Navigate to the directory and run mysql < employees.sql
      * If you set up a password and user run mysql -u [username] -p < employees.sql
  + Testing your database installation
    - Mysql -u [username] -p
    - Show databases; look for employees database
    - Use employees;
    - Show tables;
    - Desc employees; and test other tables.
* There are four operations you can perform on data in any application, you can Create, Read, Update, or Delete data. The acronym for this is CRUD. You will hear the term CRUD operations, and that means actions that Create, Read, Update, or Delete data.
  + To perform these operations, and interact with our database in other ways as well, we will use SQL (sequel) or some call it S.Q.L. This stands for Structured Query Language.
  + Let’s start with reading data from a database. To do this, we use the SELECT command. Capitalization doesn’t matter with commands, but you usually see the commands capitalized.
  + To select all data in a table, run SELECT \* FROM [tablename]; employees in this case
  + To see just the first fifty rows we could add a LIMIT clause.
    - Now we can look at the table headers and see what everything is
  + Another very helpful clause is WHERE. This allows us to add a Boolean expression that will only return data where that expression is true.
    - Where emp\_no < 10050
    - Where gender=’f’
    - Where last\_name LIKE ‘a%’;
    - Where first\_name in (‘Naftali, ‘Matt’)
    - You can also add ANDs and ORs
  + You can order your result set with ORDER BY [column name] then choose ASC or DESC.
    - ASC is default if nothing is specified
  + You can also specify specific columns to select rather than all \*
* Let’s take a look at how to create, or insert, data in MySQL.
  + To do this, we will use the INSERT INTO command.
    - insert into employees values (10000, '1992-12-12', 'Tom', 'Sawyer', 'M', '2006-12-12');
    - if some columns are nullable, and you don’t want to provide a value for each column, you can specify which columns you will insert data into as follows:
      * insert into employees(first\_name, last\_name) values (‘Tim’, ‘Smith’);
      * however, with this table it won’t work because all columns need a value as we can see be running desc employees;
    - you can enter multiple rows at once too by separating the rows in parentheses by commas.
* If you want to change data, or update it, we use the Update command.
  + Update [tablename] set column = [value] where [condition]
  + Make sure you include the where clause and condition, otherwise every single row will be updated
    - If you forget the where clause and mess up all of your data, you will need to recreate the database as shown in the previous video that talks about installing MySQL
    - The update statement will update all rows that the condition is true for, so if you included the clause WHERE gender= ‘m’, it would update all rows that have a male gender.
  + Update employees set first\_name = ‘Thomas’ where emp\_no = 10001;
  + You can also update a column with another column name or an expression, it doesn’t have to be a literal value.
    - update employees set last\_name = first\_name where emp\_no = 10001;
* Deleting data works very similarly to updating.
  + DELETE FROM [tablename] WHERE [condition]
  + Delete from employees where emp\_no < 10008;
  + Just like with the UPDATE command, be very careful to always include your WHERE clause or else you will delete every role. Also, every row that the Boolean expression is true for will be delete, so check your command closely before you run it.
    - One way you can see every row your delete command will delete before running it is by running your where clause with a select statement. This will return all rows that would be delete if ran with a delete statement.
    - Say we wanted to delete Saniya on the row with emp\_no 10008. We could run the command delete from employees where first\_name =’Saniya’, and that would definitely work and delete the row we want to get rid of. However, it would do more than just that. If we run the same where clause in a select statement we will see every row that would be deleted if we ran it with the delete statement.
      * Select \* from employees where first\_name = ‘Saniya’;
      * Select \* from employees where emp\_no = 10008;
      * Since that only returns one row, we know that’s the condition we want to include in the delete statement.
      * Delete from employees where emp\_no = 10008;
      * To verify it worked run select \* from employees limit 10;
      * You can also know that emp\_no is safe to only delete one row max because if you run: desc employees, you will see that emp\_no is a primary key, meaning that it will be unique.