Code ▾

Learning SQL (Codecademy)

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook.

SQL (Structured Query Language) is a programming language used to communicate with data stored in a relational database management system.

SQL Commands Overview:

Lesson 1: Manipulation

Hide

```
-- The CREATE TABLE Command lets you create a new table, from scratch, within a database.'

CREATE TABLE table_name (column_1, column_2, column_3, etc);

-- The columns can be assigned value formats such as: INTEGER, TEXT and BOOLEAN
```

Hide

```
-- The INSERT command enters a new row into an existing table.

INSERT INTO table_name (column_1, Column_2, column_3)
VALUES ('value_1', 'value_2', 'value_3');

-- The order of values input will correspond to the order in which the columns were scripted in
 the query.
```

Hide

```
-- The ALTER TABLE command is used to add new columns to an existing table

ALTER TABLE table_name
ADD COLUMN column_name;
```

Hide

```
-- The UPDATE command will allow you to change existing rows in a table.

UPDATE table_name
SET column_name = 'value'
WHERE column_name = 'value';
```

Hide

```
-- the DELETE FROM command allows deletion of one or more rows from a table.

DELETE FROM table_name
WHERE column_name IS 'value';
```

Hide

```
-- Adding CONSTRAINTS to your queries provides information on how your columns can be used.

CREATE TABLE table_name (column_1 INTEGER PRIMARY KEY);

-- The PRIMARY KEY constraint can be used to uniquely identify the row. Attempts to insert a row
with an identical value to a row already in the table will result in a constraint violation whic
h will not allow you to insert the new row.

CREATE TABLE table_name (column_1 TEXT UNIQUE);

-- The UNIQUE constraint will ensure that all columns have different values for every row.

CREATE TABLE table_name (column_1 TEXT NOT NULL);

-- NOT NULL columns must have a value. Attempts to insert a row without a value for a NOT NULL c
olumn will result in a constraint violation and the new row will not be inserted.

CREATE TABLE table_name (column_1 TEXT DEFAULT);

-- DEFAULT columns take an additional argument that will be the assumed value for an inserted ro
w if the new row does not specify a value for that column.
```

Lesson 2: Queries

Hide

```
-- From the most basic queries to the most advanced, all SQL queries should contain a SELECT and
FROM command. WHERE commands being the next most commonly used command in queries.

SELECT columns
FROM table_name;
```

Hide

```
-- The AS keyword allows rewording of columns and tables during queries to better suit your need
s or the situation.

SELECT column_1 AS new_name
FROM table_name;
```

Hide

```
-- The DISTINCT constraint codes the query to only return unique values when generating an outpu
t.

SELECT DISTINCT column_name
FROM table_name;
```

Hide

```
-- The WHERE clause is used to add specific restrictions to what your query will show as an output.

SELECT columns
FROM table_name
WHERE [column_name] [criteria];

--WHERE constraints can include: all operators like '=', '< >' etc. LIKE and BETWEEN.
-- The AND command can be used to add multiple WHERE restrictions to the same query.
```

Hide

```
-- The ORDER BY function allows you set the way the results in the query output will be ordered to best suit your needs and observations.

SELECT columns
FROM table_name#
ORDER BY column_name;

-- DESC can be added to the end of the command to arrange it in descending order. the command uses ascending order by default.
```

Hide

```
-- The CASE statement allows for the creation of different outputs.

SELECT columns,
CASE
WHEN [column_name] [criteria] THEN 'name'
FROM table_name;
```

Lesson 3: Aggregate Functions

Hide

```
-- The COUNT command allows you to quickly count all of the non-empty values within a specified column.

SELECT COUNT(columns)
FROM table_name;
```

Hide

```
-- The SUM funtion adds all values within a specified column together.

SELECT SUM(columns)
FROM table_name;
```

Hide

```
-- The AVG command creates a mean average of the selected values within a column.

SELECT AVG(column_name)
FROM table_name;
```

Hide

```
-- The ROUND function is used to round long digit decimals for easier reading.

SELECT ROUND(column_name, decimal_places)
FROM table_name;
```

Lesson 4: Multiple Tables

Hide

```
-- The JOIN function allows you to combine multiple tables together based on a common column.

SELECT columns
FROM table_name_1
JOIN table_name_2
ON table_name_1.column_name_1 = table_name_2.column_name_1;

-- This is commonly known as INNER JOIN where only values that match the ON condition are includ
ed.
```

Hide

```
-- A LEFT JOIN will keep all rows from the first table, regardless of whether there is a matchin
g row in the second table.

SELECT columns
FROM table_1
Left JOIN table_2
on table_1.column_name_1 = table_2.column_name_1;
```

Hide

```
-- The CROSS JOIN command combines all rows in one table to all rows of another.

SELECT table_name_1.column_name_1, table_name_2.column_name_2
FROM table_name_1
CROSS JOIN table_name_2;
```

Hide

```
-- UNION allows you to stack one dataset ontop of another.

SELECT *
FROM table_1
UNION
SELECT *
FROM table_2;

-- Tables must have the same number of columns and columns must have the same data types in the
 same order as the first table.
```

Hide

```
-- WITH command allows you to combine tables while also adding previous queries as their own col
umn in the combined table.

WITH query_name AS (column_name)
SELECT (query)
SELECT *
FROM query_name
JOIN table_name
  ON _____ = _____;
```