**SQL Training**

**Chapter 1**

SQL Basics

/\*\* COMMENTS \*\*/

The list we want to create:

Grocery List:

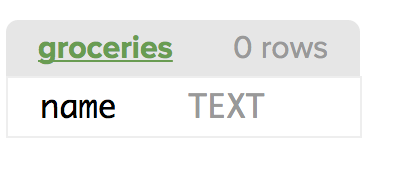
Bananas (4)

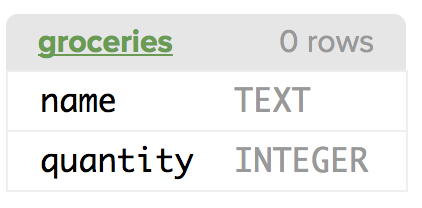
Peanut Butter (1)

Dark Chocolate Bars (2)

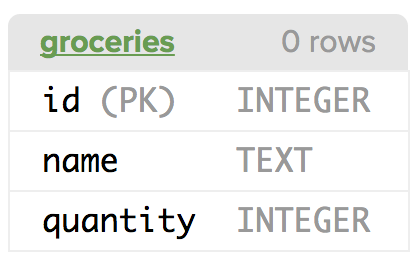
**I. CREATING A TABLE**

CREATE TABLE groceries (name TEXT);



CREATE TABLE groceries (name TEXT, quantity INTEGER); 

CREATE TABLE groceries (id INTEGER PRIMARY KEY, name TEXT, quantity INTEGER);



[the primary signals to the database that it should treat this item (id) as a row identifier and that each row must have a unique value for this column ]

**II. INSERT INFORMATION INTO TABLE**

[after creating your table]

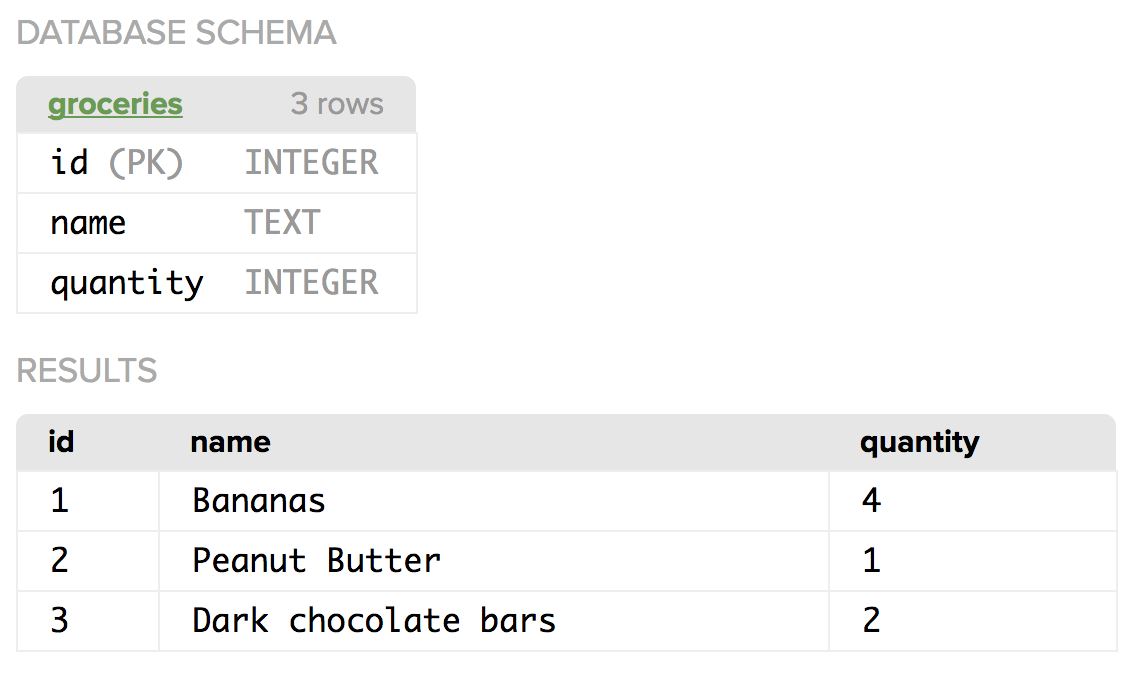
INSERT INTO groceries VALUES (1, "Bananas", 4);

INSERT INTO groceries VALUES (2, "Peanut Butter", 1);

INSERT INTO groceries VALUES (3, "Dark chocolate bars", 2);

Note:

* The strings are in double quotes “” and the integers are not
* After every statement in the insert there is a semicolon to let SQL know that this request is completed
* You are inputting the PK for every value. Can this be done automatically?



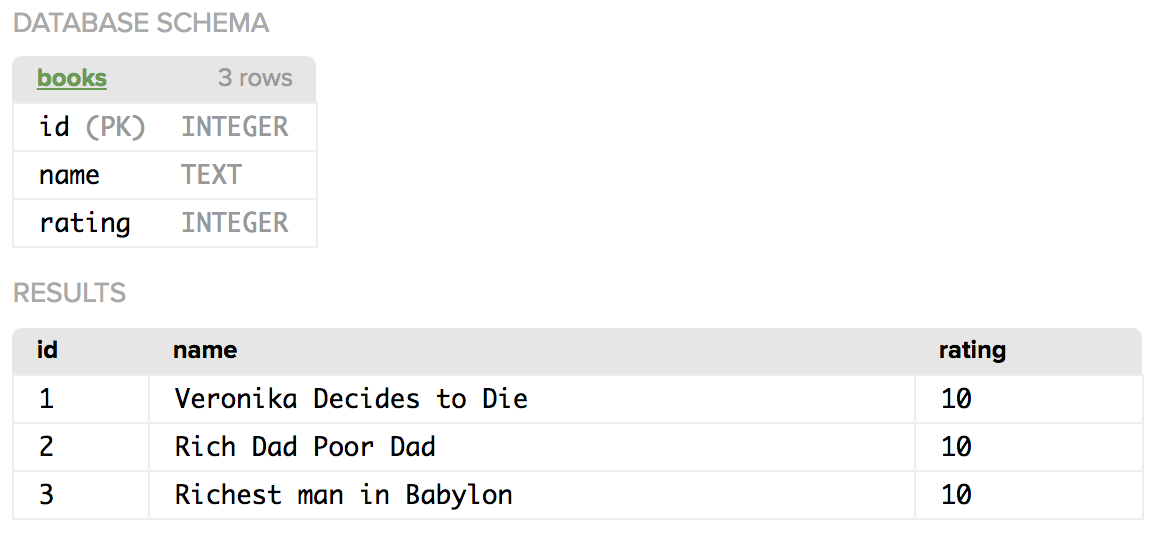
**III. CHALLENGE 1: Create a book database it must include a column for id, name, and rating**

CREATE TABLE books (id INTEGER PRIMARY KEY, name TEXT, rating INTEGER);

INSERT INTO books VALUES (1, "Veronika Decides to Die", 10);

INSERT INTO books VALUES (2, "Rich Dad Poor Dad", 10);

INSERT INTO books VALUES (3, "Richest man in Babylon", 10);



**IV. QUERYING THE TABLE**

CREATE TABLE groceries (id INTEGER PRIMARY KEY, name TEXT, quantity INTEGER, aisle INTEGER);

INSERT INTO groceries VALUES (1, "Bananas", 4, 7);

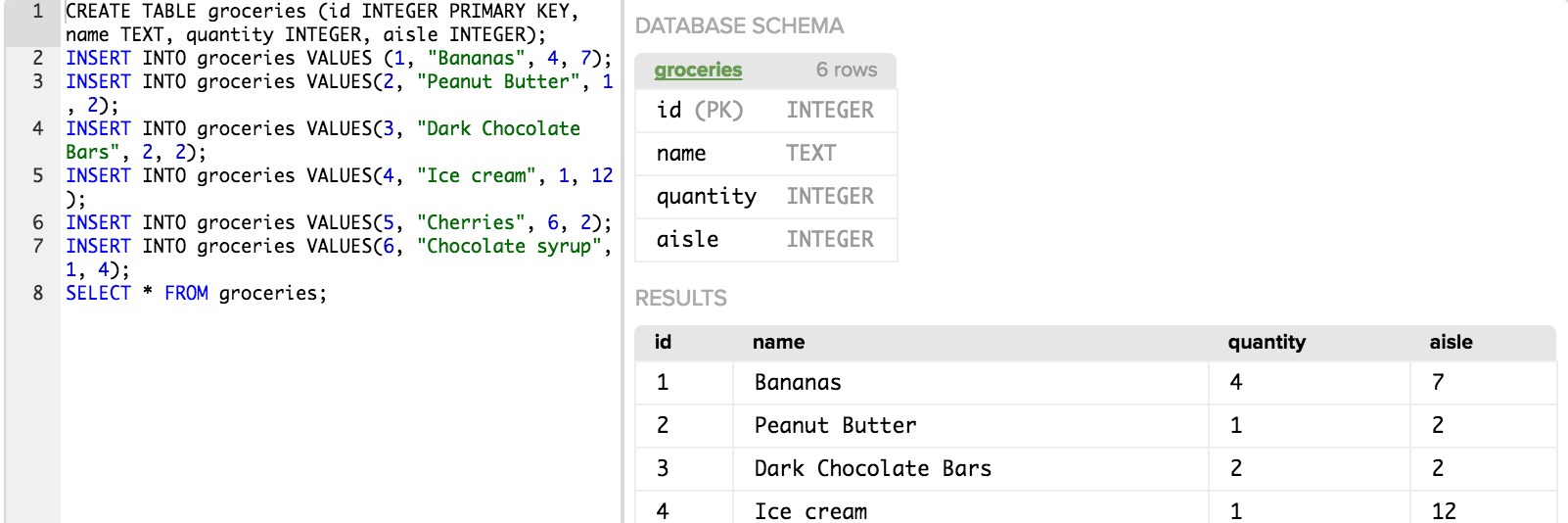
INSERT INTO groceries VALUES(2, "Peanut Butter", 1, 2);

INSERT INTO groceries VALUES(3, "Dark Chocolate Bars", 2, 2);

INSERT INTO groceries VALUES(4, "Ice cream", 1, 12);

INSERT INTO groceries VALUES(5, "Cherries", 6, 2);

INSERT INTO groceries VALUES(6, "Chocolate syrup", 1, 4);



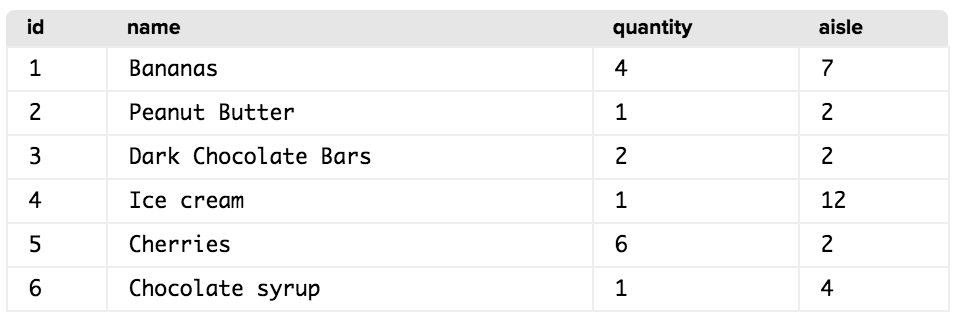
To pull a specific things from the table:

SELECT name FROM groceries;



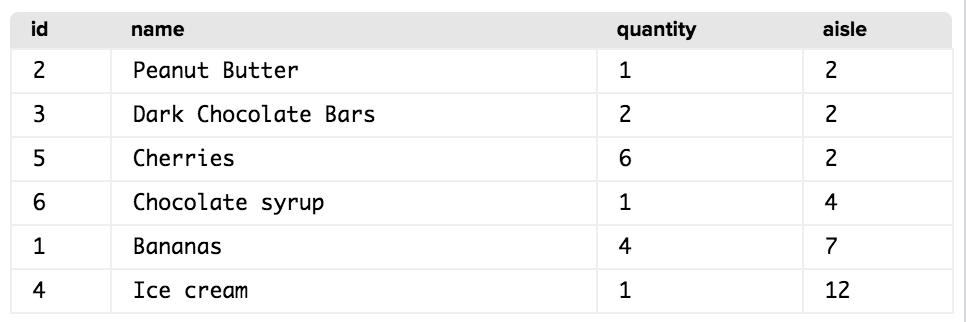
To pull all the column names and their information:

SELECT \* FROM groceries;



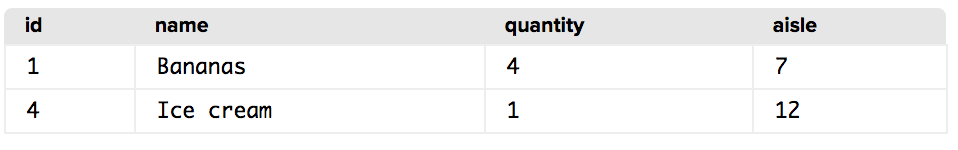
To organize the data by aisle from least to largest

SELECT \* FROM groceries ORDER BY aisle;



To learn only items that are in aisles 6-12:

SELECT \* FROM groceries WHERE aisle > 5 ORDER BY aisle;



**V. CHALLENGE: Select within the database**

You have this dataset

CREATE TABLE movies (id INTEGER PRIMARY KEY, name TEXT, release\_year INTEGER);

INSERT INTO movies VALUES (1, "Avatar", 2009);

INSERT INTO movies VALUES (2, "Titanic", 1997);

INSERT INTO movies VALUES (3, "Star Wars: Episode IV - A New Hope", 1977);

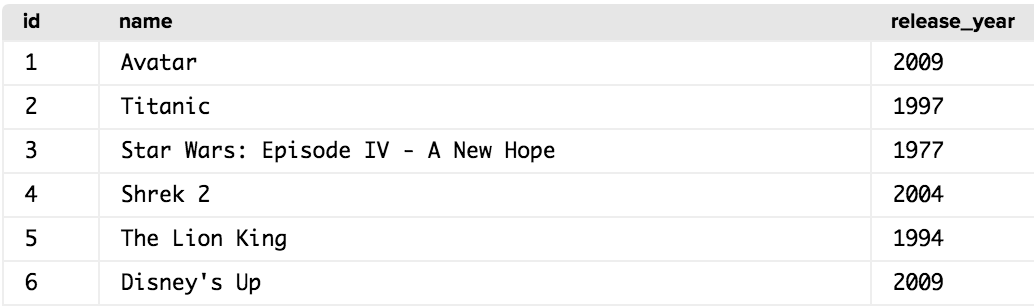
INSERT INTO movies VALUES (4, "Shrek 2", 2004);

INSERT INTO movies VALUES (5, "The Lion King", 1994);

INSERT INTO movies VALUES (6, "Disney's Up", 2009);

You want to select all the movies and all their information:

SELECT \* FROM movies;



Add query that only shows the movies that were released after the year 2000 then order it so the ealier movies are listed first.

SELECT \* FROM movies WHERE release\_year > 1999 ORDER BY release\_year;



**VI. AGGREGATING DATA**

This is our data

CREATE TABLE groceries (id INTEGER PRIMARY KEY, name TEXT, quantity INTEGER, aisle INTEGER);

INSERT INTO groceries VALUES (1, "Bananas", 4 7);

INSERT INTO groceries VALUES(2, "Peanut Butter", 1, 2);

INSERT INTO groceries VALUES(3, "Dark Chocolate Bars", 2, 2);

INSERT INTO groceries VALUES(4, "Ice cream", 1, 12);

INSERT INTO groceries VALUES(5, "Cherries", 6, 2);

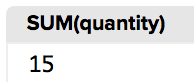
INSERT INTO groceries VALUES(6, "Chocolate syrup", 1, 4);

We want to add the total of the number of tems we have we have

CREATE TABLE groceries (id INTEGER PRIMARY KEY, name TEXT, quantity INTEGER, aisle INTEGER);

The way to do this is to SUM so we will write the formula

SLECT SUM(quantity) FROM groceries;



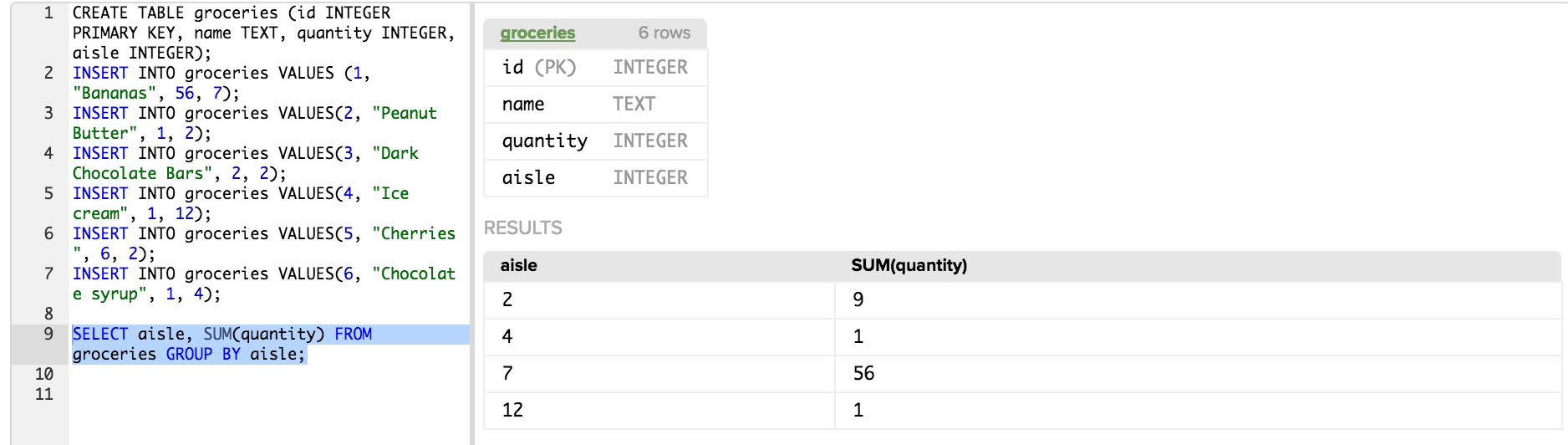
We can use other function to make calculations

If we want to know what is the maximum of any one item we will be buying:

SELECT MAX(quantity) FROM groceries;

If we want to know how many items we will be getting from every item we can use the follow formula:

SELECT aisle, SUM(quantity) FROM groceries GROUP BY aisle;



**Note**:

Total number of bananas changed

SQL reads right to left so it will runa query from the right to the left

**VII. CHALLENGE**

Here's a table containing a TODO list with the number of minutes it will take to complete each item. Then select the SUM of minutes it will take to do all of the items on your TODO list. You should only have one SELECT statement.

KEY, item TEXT, minutes INTEGER);

INSERT INTO todo\_list VALUES (1, "Wash the dishes", 15);

INSERT INTO todo\_list VALUES (2, "vacuuming", 20);

INSERT INTO todo\_list VALUES (3, "Learn some stuff on KA", 30);

SELECT SUM(minutes) FROM todo\_list