

# CIND 119: Introduction to Big Data Analytics Assignment 2 (15% of the final grade)

## Querying an RDBMS database using SQLite Studio

By: Stephanie Boissonneault

Complete this assignment using SQLite Studio.

**Question 1: Create an SQLite database called "sample".**

**Question 2: Within the "sample" database, create a table called "test\_data" and load the following data into the table: (5 points)**

SQLiteStudio (3.4.4) - [test\_data (sample)]

Database Structure View Tools Help

Databases

Filter by name

- OnlineMusicStore (SQLite 3)
- sample (SQLite 3)
  - Tables (1)
    - test\_data
  - Views

Structure Data Constraints Indexes Triggers DDL

Grid view Form view

	Order_ID	Product_Name	Category	Quantity	Price
1	1	Laptop	Electronics	5	500
2	2	Headphones	Electronics	3	100
3	3	Chair	Furniture	2	200
4	4	Desk	Furniture	1	400
5	5	iPhone	Electronics	2	800
6	6	Book	Books	10	20
7	7	Shoes	Apparel	4	150
8	8	T-shirt	Apparel	7	50
9	9	Watch	Apparel	1	250
10	10	Blender	Home Appliance	1	300
11	11	Fridge	Home Appliance	1	1200
12	12	Cookware Set	Home Appliance	3	100
13	13	Vacuum Cleaner	Home Appliance	1	350
14	14	Keyboard	Electronics	2	75
15	15	Monitor	Electronics	1	200

Status

[21:02:25] Error while executing SQL query on database 'sample': UNIQUE constraint failed: test\_data (1)

[21:03:20] Query finished in 0.010 second(s).

[21:03:25] Query finished in 0.190 second(s). Rows affected: 15

SQL editor 1 test\_data (sample)

SQLiteStudio (3.4.4) - [test\_data (sample)]

Database Structure View Tools Help

Databases

Filter by name

- OnlineMusicStore (SQLite 3)
  - sample (SQLite 3)
    - Tables (1)
      - test\_data
    - Views

Table name: test\_data

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	Order_ID	INTEGER	PRIMARY KEY							NULL
2	Product_Name	VARCHAR ...								NULL
3	Category	VARCHAR ...								NULL
4	Quantity	INTEGER								NULL
5	Price	INTEGER								NULL

Type Name Details

1 PRIMARY KEY (Order\_ID)

Status

[21:03:20] Query finished in 0.010 second(s).

[21:03:25] Query finished in 0.190 second(s). Rows affected: 15

SQLiteStudio (3.4.4) - [SQL editor 1]

Database Structure View Tools Help

Databases

Filter by name

- OnlineMusicStore (SQLite 3)
  - sample (SQLite 3)
    - Tables (1)
      - test\_data
    - Views

Query History

```

1 CREATE TABLE test_data (
2 Order_ID INTEGER,
3 Product_Name VARCHAR(120),
4 Category VARCHAR(120),
5 Quantity INTEGER,
6 Price INTEGER,
7 PRIMARY KEY (Order_ID))
8
9 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(1, 'Laptop', 'Electronics', 5, 500);
10 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(2, 'Headphones', 'Electronics', 3, 100);
11 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(3, 'Chair', 'Furniture', 2, 200);
12 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(4, 'Desk', 'Furniture', 1, 400);
13 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(5, 'iPhone', 'Electronics', 2, 800);
14 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(6, 'Book', 'Books', 10, 20);
15 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(7, 'Shoes', 'Apparel', 4, 150);
16 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(8, 'T-shirt', 'Apparel', 7, 50);
17 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(9, 'Watch', 'Apparel', 1, 250);
18 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(10, 'Blender', 'Home Appliance', 1, 300);
19 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(11, 'Fridge', 'Home Appliance', 1, 1200);
20 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(12, 'Cookware Set', 'Home Appliance', 3, 100);
21 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(13, 'Vacuum Cleaner', 'Home Appliance', 1, 350);
22 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(14, 'Keyboard', 'Electronics', 2, 75);
23 INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price) VALUES(15, 'Monitor', 'Electronics', 1, 200);

```

Status

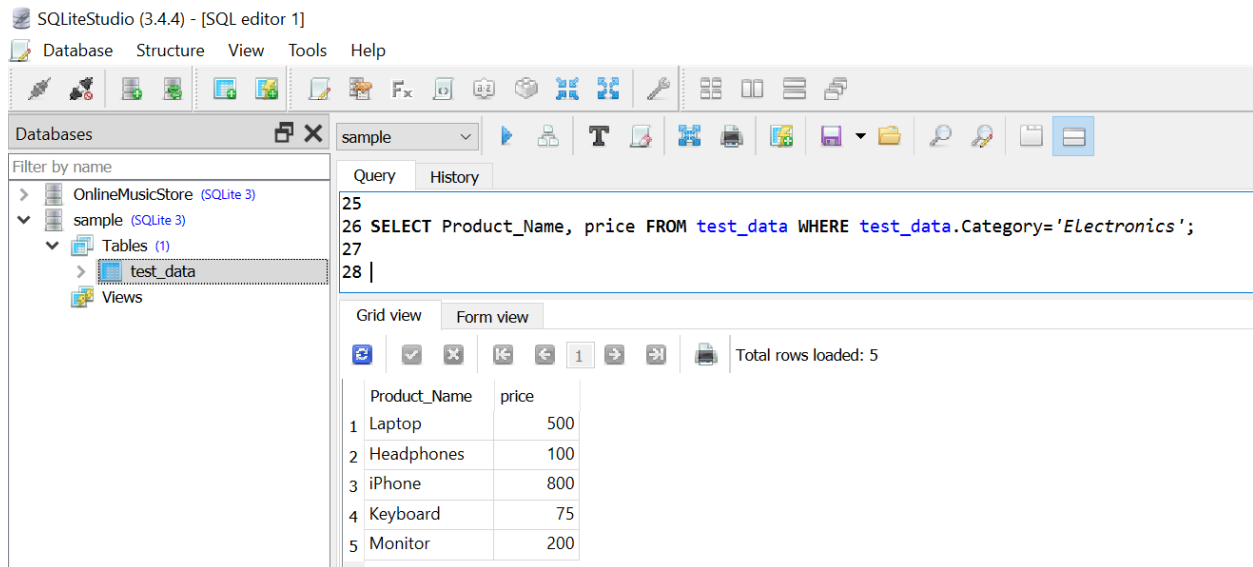
[21:03:20] Query finished in 0.010 second(s).

[21:03:25] Query finished in 0.190 second(s). Rows affected: 15

```
CREATE TABLE test_data (  
Order_ID INTEGER,  
Product_Name VARCHAR(120),  
Category VARCHAR(120),  
Quantity INTEGER,  
Price INTEGER,  
  
PRIMARY KEY (Order_ID))  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(1,'Laptop', 'Electronics', 5, 500);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(2,'Headphones', 'Electronics', 3, 100);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(3,'Chair', 'Furniture', 2, 200);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(4,'Desk', 'Furniture', 1, 400);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(5,'iPhone', 'Electronics', 2, 800);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(6,'Book', 'Books', 10, 20);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(7,'Shoes', 'Apparel', 4, 150);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(8,'T-shirt', 'Apparel', 7, 50);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(9,'Watch', 'Apparel', 1, 250);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(10,'Blender', 'Home Appliance', 1, 300);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(11,'Fridge', 'Home Appliance', 1, 1200);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(12,'Cookware Set', 'Home Appliance', 3, 100);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(13,'Vacuum Cleaner', 'Home Appliance', 1, 350);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(14,'Keyboard', 'Electronics', 2, 75);  
INSERT INTO test_data (Order_ID, Product_Name, Category, Quantity, Price)  
VALUES(15,'Monitor', 'Electronics', 1, 200);
```

**Question 3: Write SQL queries to select/compute data from the "test\_data" table.  
(2 points each)**

**a. Select the Product\_Name and Price of products where the Category is 'Electronics'.**



The screenshot shows the SQLiteStudio interface. The left pane displays the database structure with 'sample' selected, containing a table 'test\_data'. The main editor shows the following SQL query:

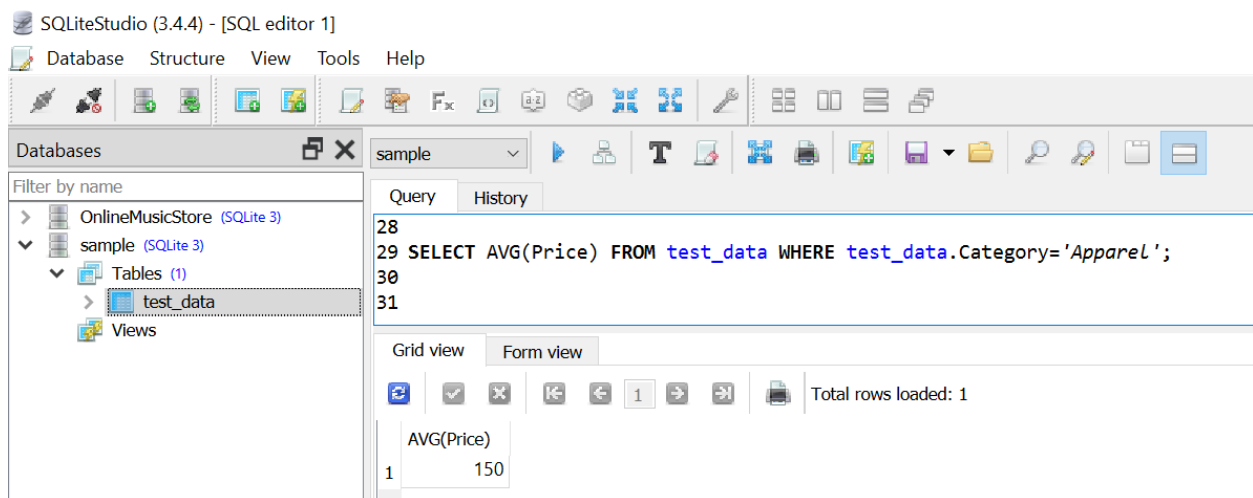
```
25  
26 SELECT Product_Name, price FROM test_data WHERE test_data.Category='Electronics';  
27  
28
```

The query results are displayed in a table with 5 rows:

	Product_Name	price
1	Laptop	500
2	Headphones	100
3	iPhone	800
4	Keyboard	75
5	Monitor	200

SELECT Product\_Name, price FROM test\_data WHERE  
test\_data.Category='Electronics';

**b. Compute the average price of products in the 'Apparel' category.**



The screenshot shows the SQLiteStudio interface. The left pane displays the database structure with 'sample' selected, containing a table 'test\_data'. The main editor shows the following SQL query:

```
28  
29 SELECT AVG(Price) FROM test_data WHERE test_data.Category='Apparel';  
30  
31
```

The query results are displayed in a table with 1 row:

	AVG(Price)
1	150

SELECT AVG(Price) FROM test\_data WHERE test\_data.Category='Apparel';

**c. Select all fields of products where the price is less than 200.**

SQLiteStudio (3.4.4) - [SQL editor 1]

Database Structure View Tools Help

Databases

Filter by name

- OnlineMusicStore (SQLite 3)
- sample (SQLite 3)
  - Tables (1)
    - test\_data
  - Views

Query History

```
31  
32 SELECT*FROM test_data WHERE test_data.Price < 200;  
33
```

Grid view Form view

Total rows loaded: 6

	Order_ID	Product_Name	Category	Quantity	Price
1	2	Headphones	Electronics	3	100
2	6	Book	Books	10	20
3	7	Shoes	Apparel	4	150
4	8	T-shirt	Apparel	7	50
5	12	Cookware Set	Home Appliance	3	100
6	14	Keyboard	Electronics	2	75

SELECT\*FROM test\_data WHERE test\_data.Price < 200;

**d. Select the Order\_ID and Product\_Name of products where the Quantity is equal to 1.**

SQLiteStudio (3.4.4) - [SQL editor 1]

Database Structure View Tools Help

Databases

Filter by name

- OnlineMusicStore (SQLite 3)
- sample (SQLite 3)
  - Tables (1)
    - test\_data
  - Views

Query History

```
34  
35 SELECT Order_ID, Product_Name FROM test_data WHERE test_data.Quantity = 1;  
36  
37
```

Grid view Form view

Total rows loaded: 6

	Order_ID	Product_Name
1	4	Desk
2	9	Watch
3	10	Blender
4	11	Fridge
5	13	Vacuum Cleaner
6	15	Monitor

SELECT Order\_ID, Product\_Name FROM test\_data WHERE test\_data.Quantity = 1;

**e. Compute the total revenue (Price \* Quantity) for each Category.**

SQLiteStudio (3.4.4) - [SQL editor 1]

Database Structure View Tools Help

Databases sample

Filter by name

- OnlineMusicStore (SQLite 3)
- sample (SQLite 3)
  - Tables (1)
    - test\_data
  - Views

Query History

```
37
38 SELECT Category, (Price * Quantity) FROM test_data GROUP BY Category;
39
```

Grid view Form view

Total rows loaded: 5

	Category	(Price * Quantity)
1	Apparel	600
2	Books	200
3	Electronics	2500
4	Furniture	400
5	Home Appliance	300

SELECT Category, (Price \* Quantity) FROM test\_data GROUP BY Category;