

Introduction to SQL for Data Analysis (Student Version)

BISA x Deloitte Workshop

Business Information Systems Association

Wednesday 1st May, 2019

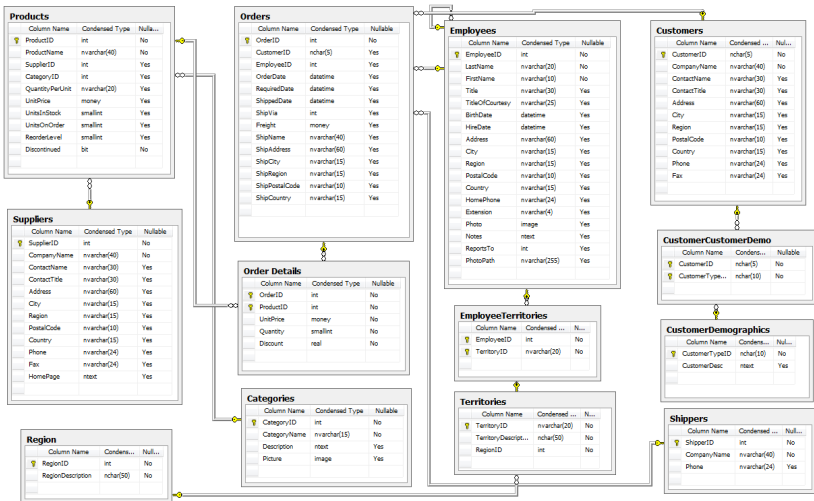
Materials developed by Jeffrey Lo

Activities Outline

1. Activity 1 - SQL Fundamentals
2. Activity 2 - SQL Joins
3. Activity 3 - Exercises

Activity 1 - SQL Fundamentals

Northwind ERD



SQLite Interface

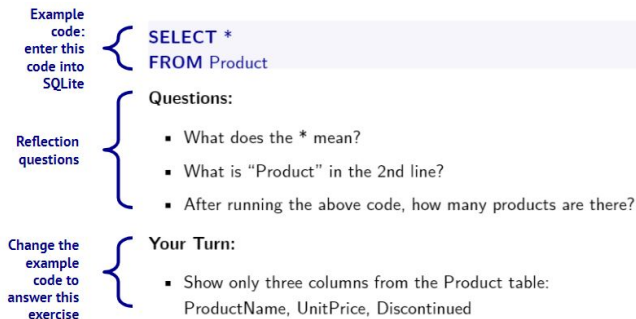
Import database Execute command Create extra tabs to run different codes

Tables

The screenshot shows the SQLite Interface application. On the left, a sidebar lists various database objects: Table, Category, Customer, CustomerCustomerDemo, CustomerDemographic, Employee, EmployeeTerritory, Order, OrderDetail, Product, Region, Shipper, Supplier, and Territory. Below these are View, Index, Trigger, and Syntax. A blue bracket on the left groups the 'Table' category and its sub-items under the label 'Tables'. The main area displays a table with 15 rows and 3 columns: id, name, and hint. The first row is a header, and the following 14 rows contain data. Above the table, there is a text input field with the SQL command '1. SELECT * FROM demo;'. To the right of the table, there is a green button with a plus sign. The top of the application has a green header bar with buttons for File, Link, Run, Export, and Import, and a 'Sign In' button on the right.

id	name	hint
1	SQLite 3.27.2	OnLine on JavaScript
2	MultiVersion	3.15.0 to Last (load on settings)
3	Dark style	Sign in - Premium (free test)
4	Size table	Fast scroll million rows
5	SQL Editor	autocomplete: [Ctrl+Space] or [Alt+Space], run: [Shift+Enter]
6	Left-Panel, Table	[RightClick] mouse "PopMenu" or [DbClick]
7	Link	Create public link DB
8	ai Url	https://old.sqliteonline.com/
9	ai Color	#9999ad
10	ai Image	Blob - png, jpg, gif or String(base64) [DbClick] row
11	SQL	Syntax example library
12	CREATE	CREATE TABLE table_name (col1, col2)
13	SELECT	SELECT * FROM table_name
14	INSERT	INSERT INTO table_name (col1, col2) VALUES ('example', 'test')
15	UPDATE	UPDATE table_name SET col1='work' WHERE col2='test'

Structure of activities will be three-fold:



We will then present a sample solution for each exercise.

1.1 SELECT Statement

```
SELECT *  
FROM Products
```

Questions:

- What does the * mean?
- What is “Products” in the 2nd line?
- After running the above code, how many products are there?

Your Turn:

- Show only three columns from the Product table:
ProductName, UnitPrice, Discontinued
- Bonus: After trying the above, can you try and add an
ORDER BY clause (similar to the FROM clause), to sort the
result by UnitPrice?

1.2 WHERE Clause (Basic Filtering)

```
SELECT ProductName, UnitPrice, Discontinued  
FROM Products  
WHERE Discontinued = 1
```

Questions:

- What does the WHERE clause mean?
- What is the unit price for the product “Alice Mutton”?

Your Turn:

- How to only show products with UnitsInStock less than 10?

1.3 BETWEEN Operator

```
SELECT OrderID, CustomerID, OrderDate, ShippedDate, Freight  
FROM Orders  
WHERE OrderDate BETWEEN '2012-12-25' AND '2012-12-31'
```

Question:

- How many orders were placed between 25th and 31st December in 2012?

Your Turn:

- What are the total costs of Freight for orders shipped between 17th and 18th September in 2013? Hint: add up the three freight manually.

1.4 LIKE Operator

```
SELECT *  
FROM Customers  
WHERE ContactName LIKE 'a%'
```

Questions:

- What does this show?
- Why do we use wildcards and the LIKE operator?

Your Turn:

- List all customers with a phone number that contains '555'
e.g. (5) 555-4729

1.5 AVG () and COUNT () Functions

```
SELECT AVG(UnitPrice)  
FROM Products
```

Question:

- What is the average unit price of products in the database?

Your Turn:

- Using the count () function, how many products are there which has a UnitPrice between \$15 and \$30? Hint: you will need to add another clause to specify the additional constraint.

1.6 GROUP BY Clause

```
SELECT CategoryID, AVG(UnitPrice)
FROM Products
GROUP BY CategoryID
```

Questions:

- What is the average unit price for category 5?
- What do you notice when you added the Group By clause, compared to just the aggregation

Your Turn:

- What is the maximum Freight Cost for each ShipRegion?
Hint: use the 'Order' table

Activity 2 - SQL Joins

2.1 INNER and LEFT Join Clauses

```
SELECT Customers.CustomerID, Customers.ContactName,  
Orders.OrderID  
FROM Customers  
INNER JOIN Orders  
ON Customers.CustomerID = Orders.CustomerID
```

Question:

- The Customer and Order tables have a 1:M relationship. The convention is FROM table 1 JOIN table 2. For an Inner Join, does the order of specifying tables matter?

Your Turn:

- Change the join type to LEFT JOIN. Do you think you get more results returned compared to an INNER JOIN?

2.2 Cross Join Clause

The syntax to implement a cross join is:

```
SELECT * FROM table_name_1 CROSS JOIN table_name_2
```

Questions:

- Before using a cross join, how many rows are there in the following tables: Region, Territory?
- Multiply the number of rows in those 2 tables - this is the expected number of rows in the result set in the following exercise.

Your Turn:

- Show all the combinations of regions and territories using a cross join.

Activity 3 - Exercises

Exercise 1

Run a query that retrieves a list of products that has less units in stock than units on order. List the product name, units on order, units in stock. Order by the product name.

Question: How many products are on the list?

Exercise 2

Run a query that shows a list of orders shipped to Belgium (country), and the first and last name of employees who placed those orders.

Hint: you will need to use JOIN and WHERE clauses.

Question: How many orders that shipped to Belgium did Margaret Peacock place?

Exercise 3

Below shows the first 5 rows of the OrderDetail table, which shows what products are inside each Order. The first 2 rows show that there are 2 products ordered in OrderId 10250. Similarly, there are 3 products in OrderId 10251.

i	Id	OrderId	ProductId	UnitPrice	Quantity	Discount
	10250/51	10250	51	42.4	35	0.15
	10250/65	10250	65	16.8	15	0.15
	10251/22	10251	22	16.8	6	0.05
	10251/57	10251	57	15.6	15	0.05
	10251/65	10251	65	16.8	20	0

Run a query that calculates the **revenue for each order**.

Hint: You only need to use the Order Details table. **Revenue for each product** is $(\text{UnitPrice} - \text{Discount}) * \text{Quantity}$