Introduction to SQL for Data Analysis (Student Version)

BISA x Deloitte Workshop

Business Information Systems Association

Wednesday 1st May, 2019

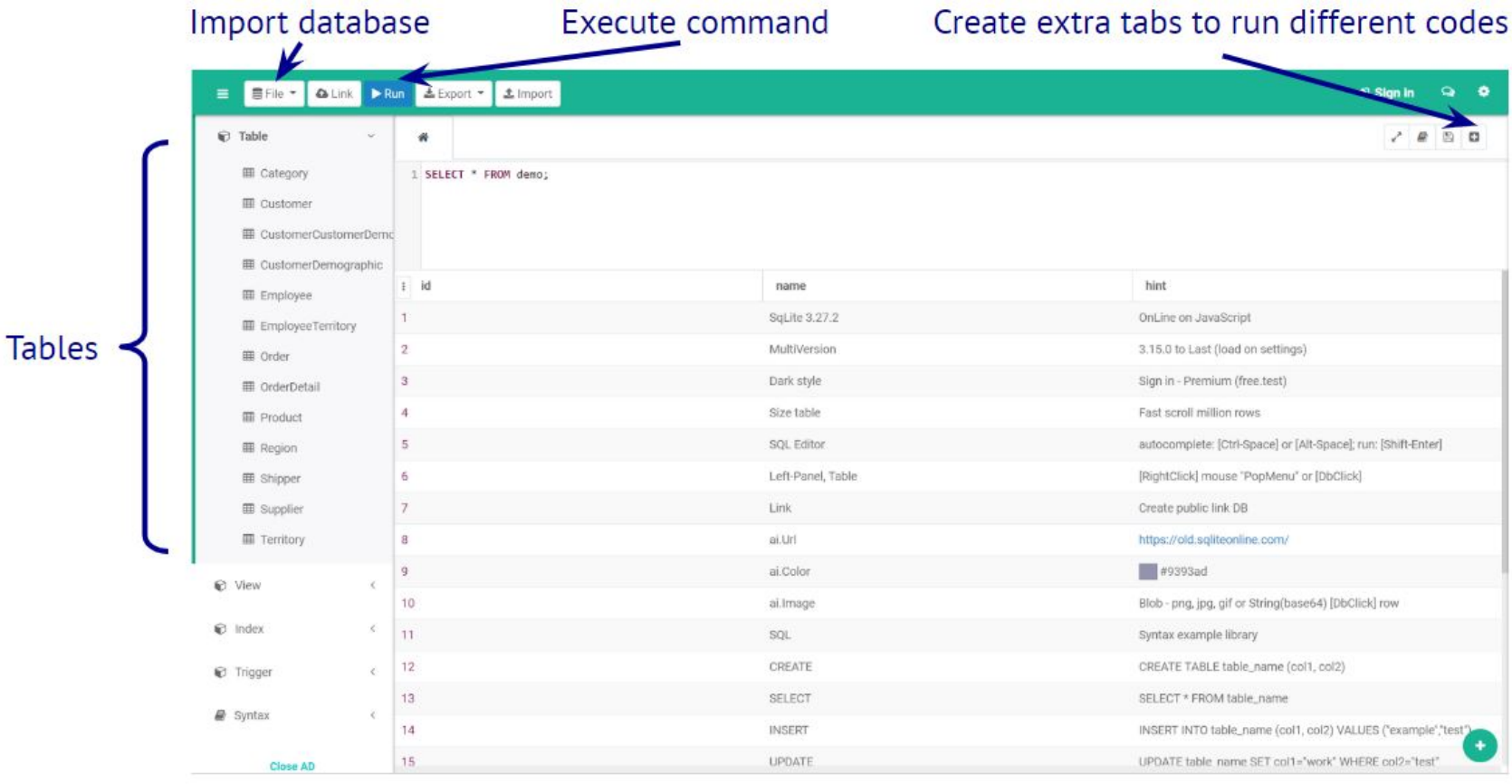
Materials developed by Jeffrey Lo

Activities Outline

1. Activity 1 - SQL Fundamentals
2. Activity 2 - SQL Joins & Union
3. Activity 3 - Exercises
4. Feedback

Activity 1 - SQL Fundamentals

SQLite Interface



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 & Union

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 Union Clause

```
SELECT City, Region, Country
FROM Customers
UNION
SELECT City, Region, Country
FROM Suppliers
```

Question:

- What is the point of using a union in the above example?

Your Turn:

- Find which tables in the database have phone numbers, and collect the entire list of phone numbers in one result.

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.

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 (UnitPrice - Discount) * Quantity

Feedback

Feedback

Please fill a short survey about the workshop, we would really appreciate it!

tinyurl.com/bisaSQLfeedback

The complete set of slides will be uploaded after the workshop.