SQL-HandsOn4

# Alberta “Albi” Kovatcheva

## Part 1

1. Create a view named "initialCustomers" that shows the first name, last name, and email address of customers that have an id of less than 100. Once that is complete, select and view your newly created view.

Query 1: Examine customer data.

SELECT \* FROM customer LIMIT 5;

Query 2: Create view.

CREATE VIEW initialCustomers AS

SELECT first\_name, last\_name, email

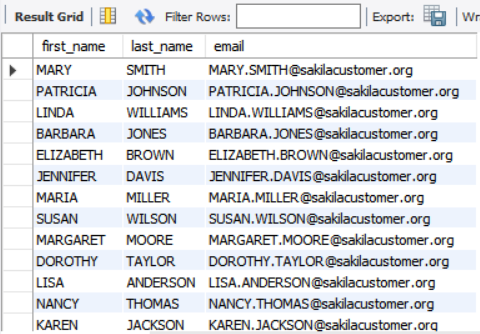
FROM customer

WHERE customer\_id < 100;

Query 3: See data in newly created view.

SELECT \* FROM initialCustomers;

Results (99 rows returned):



## Part 2

1. Complete the following:

Create a table named "ProductList". Include the following columns:

* ProductId
* ProductName
* Price
* DateAdded
* EmployeeSupportId

Include the following requirements:

* Every product should have an automatically generated id number that should be unique for each product.
* Give each column a data type that would apply
* Give the DateAdded column a default value for the current local time.
* All columns CANNOT be null.

Query 1: Delete already-existing table called ‘ProductList’.

DROP TABLE ProductList;

Query 2: Create table.

CREATE TABLE ProductList (

ProductId INTEGER PRIMARY KEY AUTO\_INCREMENT,

ProductName VARCHAR(50) NOT NULL,

Price DECIMAL(13,2) NOT NULL,

DateAdded DATETIME DEFAULT CURRENT\_TIMESTAMP,

EmployeeSupportId TINYINT UNSIGNED NOT NULL);

Next, insert one product into the table following the given guidelines when the table was created. When inserting the data, don't include the ProductId or the DateAdded. Finally, run a query to see the single product in your table.

Query 3: See data in ‘staff’ table.

SELECT \* FROM staff LIMIT 5;

Query 4: Insert data into newly-created ‘ProductList’ table.

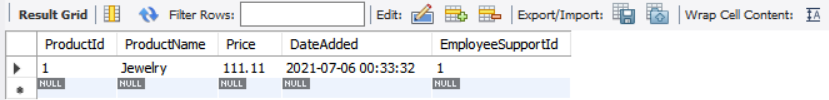
INSERT INTO ProductList (ProductName, Price, EmployeeSupportId)

VALUES("Jewelry", 111.11, 1);

Query 5: View data added to ‘ProductList’ table.

SELECT \* FROM ProductList;

Results (1 row returned):



## Part 3

Experimentation with using a foreign key.

Query 1: Creating a table. Both the EmployeeSupportId and staff\_id need to have the same variable type

for this to work.

CREATE TABLE ProductList1 (

ProductId INTEGER PRIMARY KEY AUTO\_INCREMENT,

ProductName VARCHAR(50) NOT NULL,

Price DECIMAL(13,2) NOT NULL,

DateAdded DATETIME DEFAULT CURRENT\_TIMESTAMP,

EmployeeSupportId TINYINT UNSIGNED NOT NULL,

FOREIGN KEY(EmployeeSupportId) REFERENCES staff(staff\_id));

Query 2: Finding the variable type of staff\_id to determine appropriate values for EmployeeSupportId.

DESCRIBE staff;

Query 3: Inserting data into newly-created table.

INSERT INTO ProductList1 (ProductName, Price, EmployeeSupportId)

VALUES("Jewelry", 111.11, 1);

Query 4: Viewing newly-added data.

SELECT \* FROM ProductList1;

Query 5: Dropping the redundant table used for this experiment.

DROP TABLE ProductList1;

Results (1 row returned):

