Introduction to Database and SQL Workshop
May 2019
Instructor: Fatemeh Salehian Kia

```
-- HANDS-ON: Select all customers in the U.S. and also in the New York city
SELECT customerName, city
FROM customers
WHERE country = 'USA' AND city = 'NYC';

-- HANDS-ON: Select all customers in the U.S whose credit limit is greater than
200.000 USD
SELECT customerName, creditlimit
FROM customers
WHERE country = 'USA' AND creditlimit > 200000;

-- HANDS-ON: Select contacts from the U.S., from the customers table and sort the
contacts by last name in ascending order
SELECT contactLastname,
contactFirstname
FROM customers
WHERE country = 'USA'
ORDER BY contactLastname DESC

-- HANDS-ON: Select 5 customers who have the highest credit limit SELECT
customernumber,
customername,
creditlimit
FROM customers
ORDER BY creditlimit ASC LIMIT 5;

-- HANDS-ON: Select products whose buy price is out of the range of $20 and $100
SELECT productCode,
productName,
buyPrice
FROM products
WHERE buyPrice NOT BETWEEN 20 AND 100

-- DATES! Get the orders whose required date is from 01/01/2003 to 01/31/2003
SELECT orderNumber,
requiredDate,
status
FROM orders
WHERE requireddate
BETWEEN CAST('2003-01-01' AS DATE) AND
 Created by: Fatemeh Salehian Kia
CAST('2003-01-31' AS DATE)
```

```
-- Find employee whose first name starts with T, ends with m and contains any
single character between e.g., Tom, Tim
SELECT employeeNumber, lastName, firstName
FROM employees
WHERE firstname LIKE 'T_m'

-- HANDS-ON: Find product whose product code contains string "201" SELECT
productCode, productName
FROM products
WHERE productCode LIKE '%201%'

-- HANDS-ON: Combine customerNumber and contactLastname from table
customers with employeeNumber and firstnam
(SELECT customerNumber id,contactLastname name
FROM customers)
UNION
(SELECT employeeNumber id,firstname name FROM employees)
ORDER BY name,id
SELECT products.productCode, products.productName,
productlines.textDescription
FROM products
INNER JOIN productlines ON products.productline = productlines.productline;

-- How many orders in each status -- SELECT status, count(*) FROM orders
GROUP BY status
-- find which order has total sales greater than $1000. We use the MySQL HAVING
clause on the aggregate as follows
SELECT ordernumber,
SUM(quantityOrdered) AS itemsCount,
SUM(priceeach) AS total FROM orderdetails GROUP BY ordernumber HAVING total
> 1000

-- INSERT
INSERT INTO table(column1,column2...)
VALUES (value1,value2,...)
INSERT INTO tasks(subject,start_date,end_date,description) VALUES ('Task
1','2014-01-18','2010-01-22','Description 1'), ('Task 2','2014-01-19','2010-01-
23','Description 2'),
('Task 3','2014-01-20','2010-01-24','Description 3');
 Created by: Fatemeh Salehian Kia

-- UPDATE
UPDATE employees
SET email = 'mary.patterso@classicmodelcars.com'
WHERE employeeNumber = 1056
```

-- DELETE
DELETE FROM table
[WHERE conditions] [ORDER BY ...] [LIMIT rows] DELETE FROM employees
WHERE officeCode = 4
Created by: Fatemeh Salehian Kia