[[click here if you would like to open this page in its own window]](https://docs.google.com/document/d/1ZutRSOwmzNS3bFHc3K57zs3TQyCsiY--TQhRAC3lILg/edit?usp=sharing)

# **SQL HW2: Single Table Queries: More Advanced Filtering**

For SQL HW2, you will use the tables that you created in HW1, which should be found in your 'BIS235' database. Devise the appropriate SQL queries for each of the requests below. To receive full credit your response must be a single SQL query. In devising your query, use ONLY the information provided explicitly in that quer

y (i.e. do not “look up” any additional information manually so that you may incorporate that info into your query). As an example, if you are told to return customers who do Not live in Ohio, you may not go into the database to look up all of the other states and use that information to formulate your query.

Include in your response both the query that you issued AND the output that was returned. Be sure to include ONLY your final query and the resultant output, deleting unsuccessful attempts and associated output. Label your work.

NOTE that to receive full credit for the assignment you must NUMBER your responses accurately and eliminate any output except your final answer in terms of what you input and what the system output to you in response.

Where not otherwise specified, you may return all columns in a given query (i.e. you may use “SELECT \* …”)

1. **Retrieve the names and price of all products that cost less than $10.00**

**INPUT**

SELECT prod\_name, prod\_price

FROM products

WHERE prod\_price < 10.00;

**OUTPUT**

+---------------+------------+

| prod\_name | prod\_price |

+---------------+------------+

| .5 ton anvil | 5.99 |

| 1 ton anvil | 9.99 |

| Carrots | 2.50 |

| Fuses | 3.42 |

| Oil can | 8.99 |

| Sling | 4.49 |

| TNT (1 stick) | 2.50 |

+---------------+------------+

1. **Return the price for fuses after an 8% sales tax.**

**INPUT**

SELECT prod\_name, prod\_price, 0.08 as SalesTax,

(SELECT SalesTax) \* prod\_price + 3.42 AS AfterSalesTax

FROM products WHERE prod\_id = "FU1";

**OUTPUT**

+-----------+------------+----------+---------------+

| prod\_name | prod\_price | SalesTax | AfterSalesTax |

+-----------+------------+----------+---------------+

| Fuses | 3.42 | 0.08 | 3.6936 |

+-----------+------------+----------+---------------+

1. **Retrieve the id numbers and names of any customers for whom there is no email on file.**

**INPUT**

SELECT cust\_name, cust\_email

FROM customers

WHERE cust\_email IS NULL;

**OUTPUT**

+-------------+------------+

| cust\_name | cust\_email |

+-------------+------------+

| Mouse House | NULL |

| E Fudd | NULL |

+-------------+------------+

1. **Retrieve the IDs and names and addresses of all customers who do NOT live in Ohio.**

**INPUT**

SELECT cust\_id, cust\_name, cust\_city, cust\_state, cust\_zip,

cust\_country

FROM customers

WHERE NOT cust\_state = 'OH';

**OUTPUT**

+---------+----------------+-----------+------------+----------+--------------+

| cust\_id | cust\_name | cust\_city | cust\_state | cust\_zip | cust\_country |

+---------+----------------+-----------+------------+----------+--------------+

| 10001 | Coyote Inc. | Detroit | MI | 44444 | USA |

| 10003 | Wascals | Muncie | IN | 42222 | USA |

| 10004 | Yosemite Place | Phoenix | AZ | 88888 | USA |

| 10005 | E Fudd | Chicago | IL | 54545 | USA |

1. **Retrieve all of the products sold by the vendor whose vend\_id is 1003 that cost less than or equal to $10.**

**INPUT**

SELECT prod\_name, prod\_price

FROM products

WHERE vend\_id=1003 AND prod\_price <= 10;

**OUTPUT**

+----------------+------------+

| prod\_name | prod\_price |

+----------------+------------+

| Bird seed | 10.00 |

| Carrots | 2.50 |

| Sling | 4.49 |

| TNT (1 stick) | 2.50 |

| TNT (5 sticks) | 10.00 |

+----------------+------------+

1. **Retrieve the order numbers and customer IDs for all orders that were placed over 65 months ago.**

**INPUT**

SELECT order\_num, cust\_id

FROM orders

WHERE order\_date> now() - INTERVAL 65 month;

**OUTPUT**

Empty set (0.00 sec)

1. **Retrieve all of the order numbers and their corresponding age in years. Round to two decimal places when reporting the age.**

**INPUT**

SELECT order\_num, year(curdate())-year(order\_date)

AS AGE

FROM orders;

**OUTPUT**

+-----------+------+

| order\_num | AGE |

+-----------+------+

| 20005 | 7 |

| 20006 | 17 |

| 20007 | 8 |

| 20008 | 11 |

| 20009 | 9 |

+-----------+------+

1. **Retrieve all of the customers who live in a state that borders Ohio. (Those states include IN, KY, MI, PA, and WV, and you may use this info in your query).**

**INPUT**

SELECT cust\_name, cust\_state

FROM customers

WHERE cust\_state != 'OH';

**OUTPUT**

+----------------+------------+

| cust\_name | cust\_state |

+----------------+------------+

| Coyote Inc. | MI |

| Wascals | IN |

| Yosemite Place | AZ |

| E Fudd | IL |

+----------------+------------+

1. **Retrieve the list of the products that cost $10 or more from either vendor 1001 or 1003.**

**INPUT**

SELECT prod\_name, prod\_price

FROM products

WHERE prod\_price >= 10 AND (vend\_id=1001 OR vend\_id=1003);

**OUTPUT**

+----------------+------------+

| prod\_name | prod\_price |

+----------------+------------+

| 2 ton anvil | 14.99 |

| Detonator | 13.00 |

| Bird seed | 10.00 |

| Safe | 50.00 |

| TNT (5 sticks) | 10.00 |

+----------------+------------+

1. **Retrieve the order item, product ID, quantity ordered, item price, and the total item cost (price times quantity) for each item on order number 20005.**

**INPUT**

SELECT prod\_id, quantity, item\_price,

item\_price \* quantity

AS TotalItemCost

FROM orderitems

WHERE order\_num = 20005;

**OUTPUT**

+---------+----------+------------+---------------+

| prod\_id | quantity | item\_price | TotalItemCost |

+---------+----------+------------+---------------+

| ANV01 | 10 | 5.99 | 59.90 |

| ANV02 | 3 | 9.99 | 29.97 |

| TNT2 | 5 | 10.00 | 50.00 |

| FB | 1 | 10.00 | 10.00 |

+---------+----------+------------+---------------+

1. **Your boss remembers that an important customer is located on “Riverside”, although can’t remember the street number or whether it is Street, Road, Lane, etc., or for that matter what state they are in. Retrieve all customers that have “Riverside” anywhere in their addresses.**

**INPUT**

SELECT DISTINCT cust\_name, cust\_address

FROM customers

WHERE cust\_address LIKE '%Riverside%';

**OUTPUT**

+----------------+---------------------+

| cust\_name | cust\_address |

+----------------+---------------------+

| Yosemite Place | 829 Riverside Drive |

+----------------+---------------------+

1. **Retrieve all customers whose address begins with a “1”.**

**INPUT**

SELECT DISTINCT cust\_name, cust\_address

FROM customers

WHERE cust\_address LIKE '1%';

**OUTPUT**

+-----------+---------------+

| cust\_name | cust\_address |

+-----------+---------------+

| Wascals | 1 Sunny Place |

+-----------+---------------+