#Akshat's queries -------------------------------------------

#Insert new product

INSERT INTO products (productId, Name, cost, rating, business\_user\_userID)

VALUES (201, 'filter', 20, 5, 30532106);

#Find water utility and contaminant info by zip code lookup

SELECT `Water Utility\_Name`, ContaminantName, amount, threshold

FROM `water utility\_has\_contaminant` INNER JOIN Contaminant ON `water utility\_has\_contaminant`.Contaminant\_ContaminantName = Contaminant.ContaminantName

WHERE amount > threshold AND `Water Utility\_Name` IN

(SELECT `Water Utility\_Name`

FROM `zip\_has\_water utility`

WHERE Zip\_Zip = '00001');

#Lookup contact info for water utility by zip

SELECT Name, phone, Zip\_Zip

FROM `water utility` a, `zip\_has\_water utility` b

WHERE b.Zip\_Zip = '04231' AND a.name = b.`Water Utility\_Name`;

#Get all contaminant data by zipcode

SELECT a.Zip\_zip, b.Contaminant\_ContaminantName, b.amount

FROM `zip\_has\_water utility` a, `water utility\_has\_contaminant` b

WHERE a.`Water utility\_name` = b.`Water Utility\_Name`;

#Update contaimant thresholds according to new standards

UPDATE contaminant

SET Threshold = .072

WHERE ContaminantName = 'Lead';

#Retrieve a list of zipcodes whose water utilities have a contaminant that exceeds the safe threshold

SELECT COUNT(a.Zip\_Zip)

FROM `zip\_has\_water utility` a

INNER JOIN `water utility\_has\_contaminant` b

ON a.`Water Utility\_Name` = b.`Water Utility\_Name`

INNER JOIN Contaminant c

ON b.Contaminant\_ContaminantName = c.ContaminantName

WHERE amount > Threshold;

#Trevor's Queries -------------------------------------------

#Get a list of relevant products for a user based on their zip code and what contaminants there are

select products.name, Products\_productID from `zip\_has\_water utility` INNER JOIN `water utility\_has\_contaminant` ON `zip\_has\_water utility`.`Water Utility\_Name` = `water utility\_has\_contaminant`.`Water Utility\_Name`

INNER JOIN good\_for ON `water utility\_has\_contaminant`.Contaminant\_ContaminantName = good\_for.Contaminant\_ContaminantName

INNER JOIN products ON good\_for.Products\_productID = products.productID

where Zip\_Zip = '01519';

#Get information about product details for customer

select Name, rating, review, contaminant\_contaminantname from products, `customer reviews`, good\_for

where products.productID = `customer reviews`.Products\_productID and

good\_for.products\_productID = products.productID and

products.productID = 5;

#Create new customer service instance (i.e., when a customer files a ticket and gets their service this record is created)

insert into `customer service instance` (dateOfService, `Customer Service Employee\_EmployeeID`, User\_userID)

VALUES(CURDATE(), 'E30213000', 41221993);

#See any tickets/service instances assigned to a particular employee employee

select \* from `customer service instance` where `Customer Service Employee\_EmployeeID` = 'E21000143';

#Prosper's Queries -------------------------------------------

#Insert new reviews

#Launches trigger that grabs business ID and business name from a product ID

INSERT INTO `customer reviews` (Products\_productId, review, Customer\_User\_userID)

VALUES (146, 'Culligan WSH-C125 Wall-Mount Showerhead with Shower Filter', '12000291');

#Trigger for inserting customer reviews

DELIMITER $$

CREATE TRIGGER `triggerCustomerReview` BEFORE INSERT ON `Customer Reviews`

FOR EACH ROW

BEGIN

SET NEW.Products\_Business\_User\_userID = (SELECT Business\_User\_userID from products where productID = NEW.Products\_productID);

SET NEW.Products\_Business\_Name = (SELECT Name from business where User\_UserID = NEW.Products\_Business\_User\_userID);

END;

$$

DELIMITER ;

# Business owner accessing data on product reviews

SELECT Review, Customer\_User\_userID

FROM `Customer Reviews`

WHERE Products\_productID=153;

Dionne

# create shopping cart table for customers

CREATE TABLE Shopping\_cart (

Customer\_ID int,

Price decimal (5),

Product\_ID int,

);

#add items to shopping cart

INSERT INTO shopping\_cart (customer\_ID, price, product\_ID)

VALUES (10000357, 17.99, 9);

# As a customer I want to view items in shopping cart

SELECT product\_ID

FROM shopping\_cart

WHERE Customer\_User\_userID = ‘10000357’

# As a customer I want to view items in my order

SELECT order\_has\_products.Products\_productID, order\_has\_products.Order\_orderID, order.customer\_user\_userID

FROM order, order\_has\_products

WHERE order.orderID = order\_has\_products.Order\_orderID AND Customer\_User\_userID = ‘10000654’

# As a customer I want to know if there are related products

SELECT Products\_productID

FROM good\_for

WHERE Contaminant\_contaminantname = ‘lead’

# trigger for payment added

DELIMITER $$

CREATE

TRIGGER my\_trigger BEFORE INSERT

ON payments

FOR EACH ROW BEGIN

INSERT INTO trigger\_test VALUES (‘added payment’);

END$$

DELIMITER ;

# insert credit card payment for order

INSERT INTO payments (creditcard, Order\_orderID, Order\_Customer\_User\_userID)

VALUES (1234159878960123, 10000014, 10000014);

#As a customer I want to view price total for order

SELECT orderID, customer\_user\_userID, SUM (totalprice)

FROM order

GROUP BY order\_ID

# As a customer I want to be able to provide a shipping address for the products I purchase

INSERT INTO shipment (address, Order\_orderID, Order\_Customer\_User\_userID)

VALUES (456 lake drive, 10000014, 10000014);

|  |
| --- |
|  |

|  |
| --- |
|  |