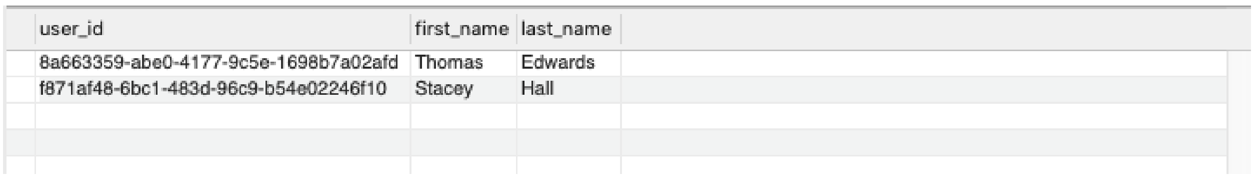
Last Name: Rachmat First Name: Anthony Student ID: 26339003

1. [10pts] For all customers who have placed at least 3 orders with a per-order total price greater than $100, list their user_id, first_name and last_name. Expected result row(s): 2

a) [7pts] SQL Query:

```
SELECT DISTINCT U.user_id, U.first_name, U.last_name
FROM user U, customers C, orders O
WHERE U.user_id = C.user_id AND
        C.user_id = O.customer_id
GROUP BY O.customer_id
HAVING 3  <= (SELECT DISTINCT COUNT(*)
                FROM orders O2
                WHERE O.customer_id = O2.customer_id AND
                O2.total_price > 100)
```

b) [3pts] Result:



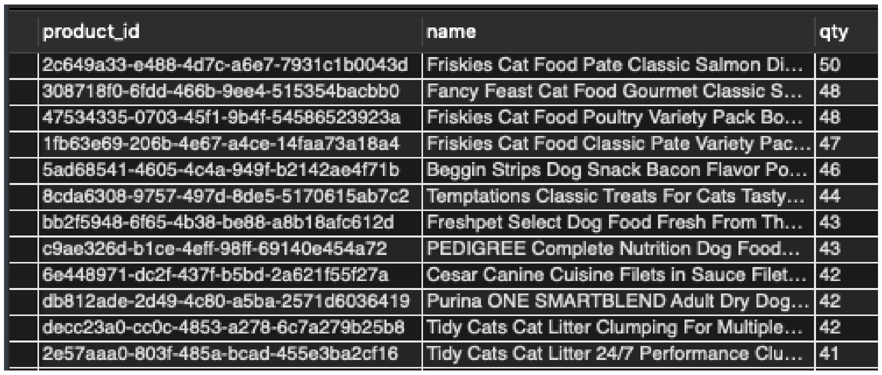
user_id	first_name	last_name
8a663359-abe0-4177-9c5e-1698b7a02afd	Thomas	Edwards
f871af48-6bc1-483d-96c9-b54e02246f10	Stacey	Hall

2. [10pts] For each product in the ‘Pet Care’ category, find the product id, product name, and the highest quantity of the product stocked by any of the stores. Rank those products by this quantity from high to low and show only the top 12 products. Expected result row(s): 12

a) [7pts] SQL Query:

```
SELECT DISTINCT P.product_id, P.name, SB.qty

FROM products P, stores S, stockedby SB

WHERE P.category = 'Pet Care'

        AND S.store_id = SB.store_id

        AND P.product_id = SB.product_id

GROUP BY P.product_id

HAVING MAX(SB.qty)

ORDER BY SB.qty DESC

LIMIT 12
```

b) [3pts] Result:



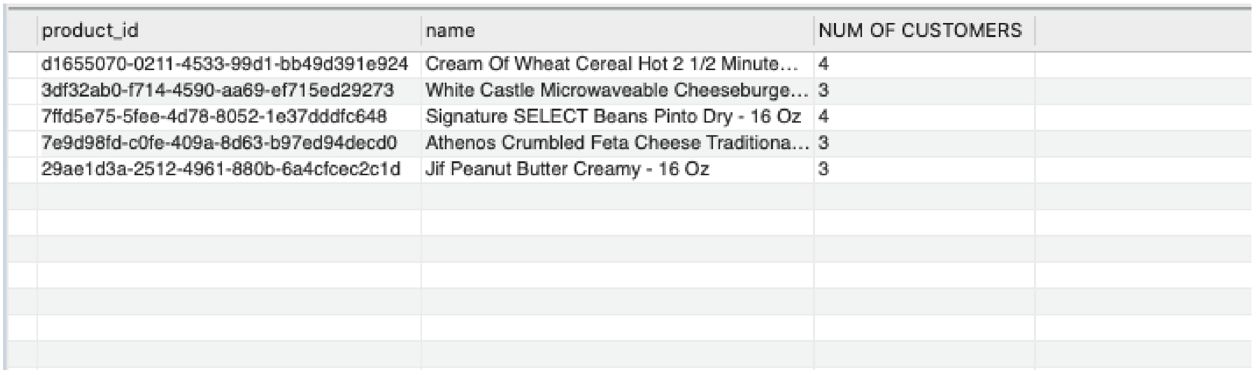
product_id	name	qty
2c649a33-e488-4d7c-a6e7-7931c1b0043d	Friskies Cat Food Pate Classic Salmon Di…	50
308718f0-6fdd-466b-9ee4-515354bacbb0	Fancy Feast Cat Food Gourmet Classic S…	48
47534335-0703-45f1-9b4f-54586523923a	Friskies Cat Food Poultry Variety Pack Bo…	48
1fb63e69-206b-4e67-a4ce-14faa73a18a4	Friskies Cat Food Classic Pate Variety Pac…	47
5ad68541-4605-4c4a-949f-b2142ae4f71b	Beggin Strips Dog Snack Bacon Flavor Po…	46
8cda6308-9757-497d-8de5-5170615ab7c2	Temptations Classic Treats For Cats Tasty…	44
bb2f5948-6f65-4b38-be88-a8b18afc612d	Freshpet Select Dog Food Fresh From Th…	43
c9ae326d-b1ce-4eff-98ff-69140e454a72	PEDIGREE Complete Nutrition Dog Food…	43
6e448971-dc2f-437f-b5bd-2a621f55f27a	Cesar Canine Cuisine Filets in Sauce Filet…	42
db812ade-2d49-4c80-a5ba-2571d6036419	Purina ONE SMARTBLEND Adult Dry Dog…	42
decc23a0-cc0c-4853-a278-6c7a279b25b8	Tidy Cats Cat Litter Clumping For Multiple…	42
2e57aaa0-803f-485a-bcad-455e3ba2cf16	Tidy Cats Cat Litter 24/7 Performance Clu…	41

3. [10pts] For all products that have been ordered by at least 5% of all customers, list their product id, product name, and the number of customers who have ordered them. Expected result row(s): 5

a) [7pts] SQL Query:

```
SELECT DISTINCT P.product_id, P.name, COUNT(O.customer_id) AS 'NUM OF CUSTOMERS'

FROM products P, orders O, orderitems OI

WHERE P.product_id = OI.product_id AND O.order_id = OI.order_id

GROUP BY P.product_id

HAVING 0.05 <= ((SELECT DISTINCT COUNT(*)

                                        FROM products P2, orders O2, orderitems OI2

           WHERE P2.product_id = OI2.product_id AND O2.order_id = OI2.order_id AND

                                                  P2.product_id = P.product_id) / (SELECT DISTINCT COUNT(*)

                                                            FROM customers C))
```

b) [3pts] Result:



product_id	name	NUM OF CUSTOMERS
d1655070-0211-4533-99d1-bb49d391e924	Cream Of Wheat Cereal Hot 2 1/2 Minute…	4
3df32ab0-f714-4590-aa69-ef715ed29273	White Castle Microwaveable Cheeseburge…	3
7ffd5e75-5fee-4d78-8052-1e37dddfc648	Signature SELECT Beans Pinto Dry - 16 Oz	4
7e9d98fd-c0fe-409a-8d63-b97ed94decd0	Athenos Crumbled Feta Cheese Traditiona…	3
29ae1d3a-2512-4961-880b-6a4cfcec2c1d	Jif Peanut Butter Creamy - 16 Oz	3

4. **Views** [20 pts]

Congratulations! The CTO has formed a data science team to analyze the shopping activities of each customer who has shopped at **Jackson Food Store**, and she has made you the head of that team. As the team leader, you have been asked to create a SQL view so that the rest of the team can simply look at the data and draw meaningful conclusions without having to deal with all of its underlying complexity.

The view should provide access to a combination of the following information:

- ❏ Customers info (user_id, first_name, last_name)
- ❏ Orders info (order_id, total_price, time_placed)
- ❏ Vehicles info (state, license_plate, make, color)

a) [15 pts] Create the desired view (JFSCustomers) by writing an appropriate CREATE VIEW statement.

CREATE VIEW JFSCustomers…;

```
 CREATE VIEW JFSCustomers (user_id, first_name, last_name, order_id, total_price, time_placed, state,
license_plate, make, color)

AS

SELECT C.user_id, U.first_name, U.last_name, O.order_id, O.total_price, O.time_placed, V.state,
V.license_plate, V.make, V.color

FROM customers C, user U, orders O, vehicles V, stores S

WHERE C.user_id = U.user_id AND

        O.customer_id = C.user_id AND

        V.license_plate = O.license_plate AND

        V.state = O.state AND

        O.store_id = S.store_id AND

        S.name = 'Jackson Food Store'
```

b) [5 pts] Show the usefulness of your view by writing a SELECT query against the view that prints the user_id, first_name, and last_name of every customer who placed an order with the highest total price. Expected result row(s): 1

```
SELECT DISTINCT C.user_id, C.first_name, C.last_name

FROM JFSCustomers C

WHERE C.total_price IN (SELECT MAX(C2.total_price)

                                        FROM JFSCustomers C2)
```

5. **Stored Procedures** [20 pts]

a) [15 pts] Create and exercise a SQL stored procedure called RegisterShopper(…) that the application can use to add a new shopper with a mobile phone to the database.

```
DELIMITER //
CREATE PROCEDURE RegisterShopper(
        user_id char(36),
   email varchar(50),
   first_name varchar(30),
   last_name varchar(30),
   mobile_number varchar(20),
   capacity integer)
BEGIN
        -- insert into User table
        INSERT INTO user(user_id, email, first_name, last_name)
        VALUES(user_id, email, first_name, last_name);
         -- insert into UserPhone table
        INSERT INTO userphone(user_id, type, number)
        VALUES(user_id, 'mobile', mobile_number);
        -- insert into Shoppers table
        INSERT INTO Shoppers(user_id, capacity)
        VALUES(user_id, capacity);
END; //
DELIMITER ;
```

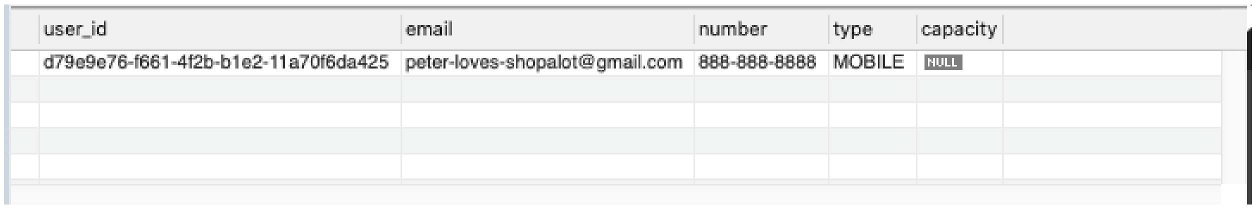
b) [5pts] Verify that your new stored procedure works properly by calling it as follows to add a new shopper and then running a SELECT query to show the stored procedure's after-effects.

```
CALL RegisterShopper (
    "d79e9e76-f661-4f2b-b1e2-11a70f6da425",
    "peter-loves-shopalot@gmail.com",
    "Peter", "Anteater", "888-888-8888", NULL);
```

```
SELECT U.user_id, U.email, P.number, P.type, S.capacity
FROM User U, Shoppers S, UserPhone P
WHERE U.user_id = S.user_id AND
      P.user_id = S.user_id AND
        S.user_id = "d79e9e76-f661-4f2b-b1e2-11a70f6da425";
```

Expected result row(s): 1 (Ignore the last row of null values if it appears)

Result:



user_id	email	number	type	capacity
d79e9e76-f661-4f2b-b1e2-11a70f6da425	peter-loves-shopalot@gmail.com	888-888-8888	MOBILE	NULL

6. **Alter Table** [10 pts]

a) [5 pts] Write and execute the ALTER TABLE statement(s) needed to modify the WorkFor table so that when the shopper associated with a WorkFor record is deleted the WorkFor record is ***not*** also deleted. It should be retained instead. (*Note:* The name of the existing foreign key constraint for shopper_id is *workfor_ibfk_2*).

DELIMITER //

BEGIN;

ALTER TABLE workfor

DROP CONSTRAINT workfor_ibfk_2;

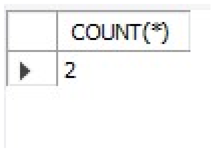
END; //

DELIMITER ;

b) [5 pts] Execute the following DELETE and SELECT statements to show the effect of your change. Report the COUNT's result (just the number) returned by the SELECT statement.

```sql
DELETE FROM Shoppers
WHERE user_id = "156d2344-227b-448a-b4e2-c460435b3a73";

SELECT COUNT(*)
FROM WorkFor W
WHERE W.shopper_id = "156d2344-227b-448a-b4e2-c460435b3a73";
```

Result:



	COUNT(*)
▶	2

7. **Triggers** [20 pts]

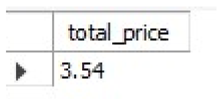
a) [15 pt] Write a CREATE TRIGGER statement (**by hand** of course!) in MySQL to define a trigger that will do the desired job.

```
DELIMITER //
CREATE TRIGGER update_total_price
AFTER INSERT ON orderitems
FOR EACH ROW
BEGIN
        UPDATE orders
        SET orders.total_price = orders.total_price + (new.qty*new.selling_price)
        WHERE new.order_id = orders.order_id;

END; //
DELIMITER ;
```

b) [5 pts] Execute the following INSERT and SELECT statements to show the effect of your trigger. Report the total_price returned by each SELECT statement.

```
SELECT O.total_price
FROM Orders O
WHERE O.order_id = "c814978d-cb1b-473c-a0e0-757c5e0fec66";
```

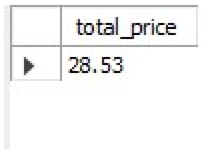
Result:



```
total_price
3.54
```

```
INSERT INTO OrderItems VALUES (
      "0f02d15f-132a-462b-8341-1c7e1e8a298b", 1, 24.99,
      "c814978d-cb1b-473c-a0e0-757c5e0fec66",
      "09b521fc-7d51-4f5e-9855-226a6ec6b631"
);

SELECT O.total_price
FROM Orders O
WHERE O.order_id = "c814978d-cb1b-473c-a0e0-757c5e0fec66";
```

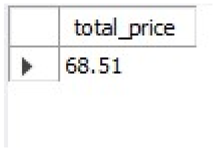
Result:



```
total_price
28.53
```

```
INSERT INTO OrderItems VALUES (
      "a1ace930-91b9-4394-bf41-bb44415e5e75", 2, 19.99,
      "c814978d-cb1b-473c-a0e0-757c5e0fec66",
      "17d368bd-5f5a-4540-bb79-a78f0399923f"
);

SELECT O.total_price
FROM Orders O
WHERE O.order_id = "c814978d-cb1b-473c-a0e0-757c5e0fec66";
```

Result:



```
total_price
68.51
```