SQL assignment 9

```
CREATE DATABASE ASSIGNMENT;
USE ASSIGNMENT;

-- CREATE TABLE

CREATE TABLE Burger_names(
burger_id int primary key,
burger_name varchar(100)
);

CREATE TABLE Burger_runner(
runner_id int primary key,
registration_date date
);



CREATE TABLE Runner_orders(
order_id int primary key,
runner_id int,
pickup_time timestamp,
distance varchar(100),
duration varchar(100),
cancellation varchar(100),
FOREIGN KEY (runner_id) REFERENCES Burger_runner(runner_id)
);

CREATE TABLE Customer_orders(
order_id int,
customer_id int,
burger_id int,
exclusions varchar(100),
extras varchar(100),
order_time timestamp,
FOREIGN KEY (order_id) REFERENCES Runner_orders(order_id),
FOREIGN KEY (burger_id) REFERENCES Burger_names(burger_id)
);

-- Inserting data into Burger_names table
INSERT INTO Burger_names (burger_id,burger_name)
values
(1,'MEATLOVERS'),
(2,'VEGETARIAN');

-- Inserting data into Burger_runner table
INSERT INTO Burger_runner (runner_id,registration_date)
VALUES
(1,'2021-01-01'),
(2,'2021-01-03'),
(3,'2021-01-08'),
(4,'2021-01-15');

-- Inserting data into Burger_runner table
INSERT INTO Runner_orders(order_id,runner_id,pickup_time,dist
VALUES
(1,1,'2021-01-01 18:15','20 km','32 minutes',Null),
(2,1,'2021-01-01 19:10','20 km','27 minutes',Null),
(3,1,'2021-01-03 00:12','13.4 km','20 minutes',Null),
(4,2,'2021-01-04 13:53','23.4 km','40 minutes',Null),
(5,3,'2021-01-08 21:10','10 km','15 minutes',Null),
(6,3,Null,Null,NULL,'Resturant cancellation'),
(7,2,'2021-01-08 21:30','25 km','25 minutes',Null),
(8,2,'2021-01-10 00:15','23.4 km','15 minutes',null),
(9,2,Null,Null,'10 minutes','Customer Cancellation'),
(10,1,'2021-01-11 18:50','10 km','10 minutes',Null);

select * from Runner_orders;

-- Inserting data into Customer_orders table
INSERT INTO Customer_orders (order_id,customer_id,burger_id,e
VALUES
(1,101,1,Null,Null,'2021-01-01 18:05:02'),
(2,101,1,Null,Null,'2021-01-01 19:00:52'),
(3,102,1,Null,Null,'2021-01-02 23:51:23'),
(3,102,2,Null,Null,'2021-01-02 23:51:23'),
(4,103,1,'4',Null,'2021-01-04 13:23:46'),
(4,103,1,'4',Null,'2021-01-04 13:23:46'),
(4,103,2,'4',Null,'2021-01-04 13:23:46'),
(5,104,1,Null,'1','2021-01-08 21:00:29'),
(6,101,2,Null,Null,'2021-01-08 21:03:13'),
(7,105,2,Null,'1','2021-01-08 21:20:29'),
(8,102,1,Null,Null,'2021-01-09 23:54:33'),
(9,103,1,'4','1,5','2021-01-10 11:22:59'),
(10,104,1,Null,Null,'2021-01-11 18:34:49'),
(10,104,1,'2,6','1,4','2021-01-11 18:34:49');

-- Q1. How many burgers were ordered?
SELECT count(order_id) as Total_burger_ordered
FROM Customer_orders;                    -- 14 orders

-- Q2. How many unique customer orders were made?
SELECT distinct order_id as unique_customer_orders
FROM Customer_orders;                   -- 10 distinct custome

-- Q3. How many successful orders were delivered by each runne
SELECT runner_id, count(*) as successful_orders
FROM Runner_orders
where cancellation is Null              -- runner_id 1 = 4
group by runner_id                      -- runner_id 2 = 3
order by runner_id;                     -- runner_id 3 = 1

-- Q4. How many of each type of burger was delivered?
SELECT CO.burger_id,BN.burger_name, RO.cancellation,count(*) 
FROM Customer_orders as CO
JOIN Runner_orders as RO ON CO.order_id = RO.order_id
JOIN Burger_names as BN ON BN.burger_id = CO.burger_id
where cancellation is Null
group by CO.burger_id,RO.cancellation;                      -

-- Q5. How many Vegetarian and Meatlovers were ordered by eac
```

```
SELECT CO.burger_id, BN.burger_name, count(*) as No_of_Orders
FROM Customer_orders as CO
JOIN Burger_names as BN
ON BN.burger_id = CO.burger_id
group by CO.burger_id, BN.burger_name;                      -


-- Q6. What was the maximum number of burgers delivered in a
SELECT order_id, count(burger_id) as burger_count
FROM Customer_orders
group by order_id
ORDER BY burger_count DESC
LIMIT 1;

-- Q7. For each customer, how many delivered burgers had at l
SELECT customer_id,
sum(case when(exclusions is not Null or extras is not null) t
sum(case when(exclusions is Null or extras is null) then 1 el
FROM Customer_orders
group by customer_id;

-- Q8. What was the total volume of burgers ordered for each
SELECT HOUR(order_time) AS order_hour,  COUNT(*) AS total_bur
FROM Customer_orders
GROUP BY HOUR(order_time)
ORDER BY order_hour;

-- Q9. How many runners signed up for each 1 week period?
SELECT YEAR(registration_date) as year, WEEK(registration_dat
FROM  Burger_runner
group by Year, week;

-- Q10.What was the average distance travelled for each custo
SELECT co.customer_id, AVG(CAST(SUBSTRING(ro.distance, 1, LEN
FROM Customer_orders co
JOIN Runner_orders ro
ON co.order_id = ro.order_id
```

```
WHERE ro.distance IS NOT NULL
GROUP BY co.customer_id;
```