SQL CODING ASSESSMENT

DATABASE USED : Burger Bash db

CREATE TABLES AND INSERTED VALUES:

```
CREATE TABLE runner_orders(
  order_id     INTEGER  NOT NULL PRIMARY KEY
 ,runner_id    INTEGER  NOT NULL
 ,pickup_time  datetime
 ,distance     VARCHAR(7)
 ,duration     VARCHAR(10)
 ,cancellation VARCHAR(23)
);
INSERT INTO runner_orders VALUES (1,1,'2021-01-01 18:15:34','20km','32 minutes',NULL);
INSERT INTO runner_orders VALUES (2,1,'2021-01-01 19:10:54','20km','27 minutes',NULL);
INSERT INTO runner_orders VALUES (3,1,'2021-01-03 00:12:37','13.4km','20 mins',NULL);
INSERT INTO runner_orders VALUES (4,2,'2021-01-04 13:53:03','23.4','40',NULL);
INSERT INTO runner_orders VALUES (5,3,'2021-01-08 21:10:57','10','15',NULL);
INSERT INTO runner_orders VALUES (6,3,NULL,NULL,NULL,'Restaurant Cancellation');
INSERT INTO runner_orders VALUES (7,2,'2021-01-08 21:30:45','25km','25mins',NULL);
INSERT INTO runner_orders VALUES (8,2,'2021-01-10 00:15:02','23.4 km','15 minute',NULL);
INSERT INTO runner_orders VALUES (9,2,NULL,NULL,NULL,'Customer Cancellation');
INSERT INTO runner_orders VALUES (10,1,'2021-01-11 18:50:20','10km','10minutes',NULL);

CREATE TABLE customer_orders(
  order_id    INTEGER  NOT NULL
 ,customer_id INTEGER  NOT NULL
 ,burger_id   INTEGER  NOT NULL
 ,exclusions  VARCHAR(4)
 ,extras      VARCHAR(4)
 ,order_time  datetime NOT NULL
);
INSERT INTO customer_orders VALUES (1,101,1,NULL,NULL,'2021-01-01 18:05:02');
INSERT INTO customer_orders VALUES (2,101,1,NULL,NULL,'2021-01-01 19:00:52');
INSERT INTO customer_orders VALUES (3,102,1,NULL,NULL,'2021-01-02 23:51:23');
INSERT INTO customer_orders VALUES (3,102,2,NULL,NULL,'2021-01-02 23:51:23');
INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (4,103,2,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (5,104,1,NULL,'1','2021-01-08 21:00:29');
INSERT INTO customer_orders VALUES (6,101,2,NULL,NULL,'2021-01-08 21:03:13');
INSERT INTO customer_orders VALUES (7,105,2,NULL,'1','2021-01-08 21:20:29');
INSERT INTO customer_orders VALUES (8,102,1,NULL,NULL,'2021-01-09 23:54:33');
INSERT INTO customer_orders VALUES (9,103,1,'4','1, 5','2021-01-10 11:22:59');
INSERT INTO customer_orders VALUES (10,104,1,NULL,NULL,'2021-01-11 18:34:49');
INSERT INTO customer_orders VALUES (10,104,1,'2, 6','1, 4','2021-01-11 18:34:49');

CREATE TABLE burger_names(
  burger_id   INTEGER  NOT NULL PRIMARY KEY
 ,burger_name VARCHAR(10) NOT NULL
);
INSERT INTO burger_names(burger_id,burger_name) VALUES (1,'Meatlovers');
INSERT INTO burger_names(burger_id,burger_name) VALUES (2,'Vegetarian');

CREATE TABLE burger_runner(
  runner_id   INTEGER  NOT NULL PRIMARY KEY
```

```
  ,registration_date date NOT NULL
);
INSERT INTO burger_runner VALUES (1,'2021-01-01');
INSERT INTO burger_runner VALUES (2,'2021-01-03');
INSERT INTO burger_runner VALUES (3,'2021-01-08');
INSERT INTO burger_runner VALUES (4,'2021-01-15');
```

QUERIES:

```
      -- 1. Maximum and minimum distance covered by each runner (only completed orders)
SELECT r.runner_id,
      MAX(r.distance) AS max_distance,
      MIN(r.distance) AS min_distance
FROM runner_orders r
JOIN customer_orders c ON r.order_id = c.order_id
WHERE r.cancellation IS NULL
GROUP BY r.runner_id;
```

	runner_id	max_distance	min_distance
1	1	20km	10km
2	2	25km	23.4
3	3	10	10

```
--  2. Number of orders for each burger(only delivered ones)
SELECT c.burger_id,
      COUNT(*) AS total_orders
FROM customer_orders c
JOIN runner_orders r ON c.order_id = r.order_id
WHERE r.cancellation IS NULL
GROUP BY c.burger_id;
```

	burger_id	total_orders
1	1	9
2	2	3

```
-- 3. Burger-wise minimum order date
SELECT c.burger_id,
      MIN(c.order_time) AS first_order_time
FROM customer_orders c
JOIN runner_orders r ON c.order_id = r.order_id
WHERE r.cancellation IS NULL
GROUP BY c.burger_id;
```

	burger_id	first_order_time
1	1	2021-01-01 18:05:02.000
2	2	2021-01-02 23:51:23.000

```
-- 4. Number of burgers per order
SELECT order_id, COUNT(*) AS burger_count
FROM customer_orders
GROUP BY order_id;
```

	order_id	burger_count
1	1	1
2	2	1
3	3	2
4	4	3
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	2

```sql
-- 5. Earliest pickup time by each runner (completed orders only)
SELECT r.runner_id,
    MIN(r.pickup_time) AS first_pickup
FROM runner_orders r
JOIN customer_orders c ON r.order_id = c.order_id
WHERE r.cancellation IS NULL
GROUP BY r.runner_id;
```

	runner_id	first_pickup
1	1	2021-01-01 18:15:34.000
2	2	2021-01-04 13:53:03.000
3	3	2021-01-08 21:10:57.000

```sql
-- 6. Average Distance Covered by Each Runner (Only Completed Orders)
SELECT r.runner_id,
    AVG(TRY_CAST(r.distance AS FLOAT)) AS avg_distance
FROM runner_orders r
JOIN customer_orders c ON r.order_id = c.order_id
WHERE r.cancellation IS NULL
GROUP BY r.runner_id;
```

	runner_id	avg_distance
1	1	NULL
2	2	23.4
3	3	10