

```
DROP TABLE IF EXISTS runners
CREATE TABLE runners (
  runner_id INTEGER,
  registration_date DATE
)
INSERT INTO runners
(runner_id, registration_date)
VALUES
(1, '2021-01-01'),
(2, '2021-01-03'),
(3, '2021-01-08'),
(4, '2021-01-15');
SELECT * FROM runners

DROP TABLE IF EXISTS customer_orders
CREATE TABLE customer_orders (
```

97 %

Results Messages

	runner_id	registration_date
1	1	2021-01-01
2	2	2021-01-03
3	3	2021-01-08
4	4	2021-01-15

Query executed successfully.

Customer pizza orders are captured in the below `customer_orders` table. One single order may contain more than 1 pizza.

The `pizza_id` relates to the type of pizza which was ordered whilst the `exclusions` are the `ingredient_id` values which should be removed from the pizza and the `extras` are the `ingredient_id` values which need to be added to the pizza.

Customers can order multiple pizzas in a single order with varying `exclusions` and `extras` values even if the pizza is the same type!

The `exclusions` and `extras` columns will need to be cleaned up before using them in the queries.

```

DROP TABLE IF EXISTS customer_orders
CREATE TABLE customer_orders (
  order_id INTEGER,
  customer_id INTEGER,
  pizza_id INTEGER,
  exclusions VARCHAR(4),
  extras VARCHAR(4),
  order_time DATETIME
)
INSERT INTO customer_orders
(order_id, customer_id, pizza_id, exclusions, extras, order_time)
VALUES
('1', '101', '1', '', '', '2020-01-01 18:05:02'),
('2', '101', '1', '', '', '2020-01-01 19:00:52'),
('3', '102', '1', '', '', '2020-01-02 23:51:23'),
('3', '102', '2', '', NULL, '2020-01-02 23:51:23'),
('4', '103', '1', '4', '', '2020-01-04 13:23:46'),
('4', '103', '1', '4', '', '2020-01-04 13:23:46'),
('4', '103', '2', '4', '', '2020-01-04 13:23:46'),
('5', '104', '1', 'null', '1', '2020-01-08 21:00:29'),
('6', '101', '2', 'null', 'null', '2020-01-08 21:03:13'),
('7', '105', '2', 'null', '1', '2020-01-08 21:20:29'),
('8', '102', '1', 'null', 'null', '2020-01-09 23:54:33'),
('9', '103', '1', '4', '1, 5', '2020-01-10 11:22:59'),
('10', '104', '1', 'null', 'null', '2020-01-11 18:34:49'),
('10', '104', '1', '2, 6', '1, 4', '2020-01-11 18:34:49');
SELECT * FROM customer_orders

```

97 %

Results Messages

	order_id	customer_id	pizza_id	exclusions	extras	order_time
1	1	101	1			2020-01-01 18:05:02.000
2	2	101	1			2020-01-01 19:00:52.000

✓ Query executed successfully.

After each orders are received through the system - they are assigned to a runner - however not all orders are fully completed and can be cancelled by the restaurant or the customer.

The `pickup_time` is the datetime at which the runner arrives at the Pizza Runner headquarters to pick up the freshly cooked pizzas.

The `distance` and `duration` fields are related to how far and long the runner had to travel to deliver the order to the respective customer.

There are some known data issues with this table so I need to be careful and make sure to check the data types for each column in the schema SQL!

```

DROP TABLE IF EXISTS runner_orders
CREATE TABLE runner_orders (
  order_id INTEGER,
  runner_id INTEGER,
  pickup_time VARCHAR(19),
  distance VARCHAR(7),
  duration VARCHAR(10),
  cancellation VARCHAR(23)
)
INSERT INTO runner_orders
(order_id, runner_id, pickup_time, distance, duration, cancellation)
VALUES
('1', '1', '2020-01-01 18:15:34', '20km', '32 minutes', ''),
('2', '1', '2020-01-01 19:10:54', '20km', '27 minutes', ''),
('3', '1', '2020-01-03 00:12:37', '13.4km', '20 mins', NULL),
('4', '2', '2020-01-04 13:53:03', '23.4', '40', NULL),
('5', '3', '2020-01-08 21:10:57', '10', '15', NULL),
('6', '3', 'null', 'null', 'null', 'Restaurant Cancellation'),
('7', '2', '2020-01-08 21:30:45', '25km', '25mins', 'null'),
('8', '2', '2020-01-10 00:15:02', '23.4 km', '15 minute', 'null'),
('9', '2', 'null', 'null', 'null', 'Customer Cancellation'),
('10', '1', '2020-01-11 18:50:20', '10km', '10minutes', 'null');
SELECT * FROM runner_orders

```

97 %

Results Messages

	order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	1	2020-01-01 18:15:34	20km	32 minutes	
2	2	1	2020-01-01 19:10:54	20km	27 minutes	
3	3	1	2020-01-03 00:12:37	13.4km	20 mins	NULL
4	4	2	2020-01-04 13:53:03	23.4	40	NULL
5	5	3	2020-01-08 21:10:57	10	15	NULL

✓ Query executed successfully.

At the moment - Pizza Runner only has 2 pizzas available the Meat Lovers or Vegetarian!

Each `pizza_id` has a standard set of `toppings` which are used as part of the pizza recipe.

```

DROP TABLE IF EXISTS pizza_names
CREATE TABLE pizza_names (
  pizza_id INTEGER,
  pizza_name TEXT
)
INSERT INTO pizza_names
(pizza_id, pizza_name)
VALUES
(1, 'Meatlovers'),
(2, 'Vegetarian');
SELECT * FROM pizza_names

```

97 %

	pizza_id	pizza_name
1	1	Meatlovers
2	2	Vegetarian

```

DROP TABLE IF EXISTS pizza_recipes
CREATE TABLE pizza_recipes (
  pizza_id INTEGER,
  toppings TEXT
)
INSERT INTO pizza_recipes
(pizza_id, toppings)
VALUES
(1, '1, 2, 3, 4, 5, 6, 8, 10'),
(2, '4, 6, 7, 9, 11, 12');
SELECT * FROM pizza_recipes

```

97 %

	pizza_id	toppings
1	1	1, 2, 3, 4, 5, 6, 8, 10
2	2	4, 6, 7, 9, 11, 12

The below table contains all of the `topping_name` values with their corresponding `topping_id` value

```
DROP TABLE IF EXISTS pizza_toppings
CREATE TABLE pizza_toppings (
  topping_id INTEGER,
  topping_name TEXT
)
INSERT INTO pizza_toppings
(topping_id, topping_name)
VALUES
(1, 'Bacon'),
(2, 'BBQ Sauce'),
(3, 'Beef'),
(4, 'Cheese'),
(5, 'Chicken'),
(6, 'Mushrooms'),
(7, 'Onions'),
(8, 'Pepperoni'),
(9, 'Peppers'),
(10, 'Salami'),
(11, 'Tomatoes'),
(12, 'Tomato Sauce');
SELECT * FROM pizza_toppings
```

97 %

Results Messages

	topping_id	topping_name
1	1	Bacon
2	2	BBQ Sauce
3	3	Beef
4	4	Cheese
5	5	Chicken

✓ Query executed successfully.