Usecase Introduction:

Did you know that over 115 million kilograms of pizza is consumed daily worldwide??? (Well according to Wikipedia anyway…)

Tony was scrolling through his Instagram feed when something really caught his eye - “80s Retro Styling and Pizza Is the Future!”

Tony was sold on the idea, but he knew that pizza alone was not going to help him get seed funding to expand his new Pizza Empire - so he had one more genius idea to combine with it - he was going to Uberize it - and so Pizza Runner was launched!

Tony started by recruiting “runners” to deliver fresh pizza from Pizza Runner Headquarters (otherwise known as Tony’s house) and also maxed out his credit card to pay freelance developers to build a mobile app to accept orders from customers.

Table 1: runners

The `runners` table shows the `registration_date` for each new runner

runner_id	registration_date

Table 2: customer_orders

Customer pizza orders are captured in the `customer_orders` table with 1 row for each individual pizza that is part of the order.

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.

Note that 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 your queries.

order_id	customer_id	pizza_id	exclusions	extras	order_time

Table 3: runner_orders

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 timestamp 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 be careful when using this in your queries - make sure to check the data types for each column in the schema SQL!

order_id	runner_id	pickup_time	distance	duration	cancellation

Table 4: pizza_names

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

pizza_id	pizza_name

Table 5: pizza_recipes

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

pizza_id	toppings

Table 6: pizza_toppings

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

topping_id	topping_name

Pizza Metrics:

1) How many pizzas were ordered?
2) How many unique customer orders were made?
3) How many successful orders were delivered by each runner?
4) How many of each type of pizza was delivered?
5) How many Vegetarian and Meatlovers were ordered by each customer?
6) What was the maximum number of pizzas delivered in a single order?
7) For each customer, how many delivered pizzas had at least 1 change and how many had no changes?
8) How many pizzas were delivered that had both exclusions and extras?
9) What was the total volume of pizzas ordered for each hour of the day?
10) What was the volume of orders for each day of the week?