**Case Study #2 - Pizza Runner**

Danny Ma · May 4, 2021

A poster for a video game

Description automatically generated

**Introduction**

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

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

Danny 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!

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

**Available Data**

Because Danny had a few years of experience as a data scientist - he was very aware that data collection was going to be critical for his business’ growth.

He has prepared for us an entity relationship diagram of his database design but requires further assistance to clean his data and apply some basic calculations so he can better direct his runners and optimize Pizza Runner’s operations.

All datasets exist within the pizza\_runner database schema - be sure to include this reference within your SQL scripts as you start exploring the data and answering the case study questions.

**Entity Relationship Diagram**

**Table 1: runners**

The runners table shows the registration\_date for each new runner.

| **runner\_id** | **registration\_date** |
| --- | --- |
| 1 | 2021-01-01 |
| 2 | 2021-01-03 |
| 3 | 2021-01-08 |
| 4 | 2021-01-15 |

**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** |
| --- | --- | --- | --- | --- | --- |
| 1 | 101 | 1 |  |  | 2021-01-01 18:05:02 |
| 2 | 101 | 1 |  |  | 2021-01-01 19:00:52 |
| 3 | 102 | 1 |  |  | 2021-01-02 23:51:23 |
| 3 | 102 | 2 |  | NaN | 2021-01-02 23:51:23 |
| 4 | 103 | 1 | 4 |  | 2021-01-04 13:23:46 |
| 4 | 103 | 1 | 4 |  | 2021-01-04 13:23:46 |
| 4 | 103 | 2 | 4 |  | 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 |

**Table 3: runner\_orders**

After each order 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** |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2021-01-01 18:15:34 | 20km | 32 minutes |  |
| 2 | 1 | 2021-01-01 19:10:54 | 20km | 27 minutes |  |
| 3 | 1 | 2021-01-03 00:12:37 | 13.4km | 20 mins | NaN |
| 4 | 2 | 2021-01-04 13:53:03 | 23.4 | 40 | NaN |
| 5 | 3 | 2021-01-08 21:10:57 | 10 | 15 | NaN |
| 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 |

**Table 4: pizza\_names**

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

| **pizza\_id** | **pizza\_name** |
| --- | --- |
| 1 | Meat Lovers |
| 2 | Vegetarian |

**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** |
| --- | --- |
| 1 | 1, 2, 3, 4, 5, 6, 8, 10 |
| 2 | 4, 6, 7, 9, 11, 12 |

**Table 6: pizza\_toppings**

This table contains all the topping\_name values with their corresponding topping\_id value.

| **topping\_id** | **topping\_name** |
| --- | --- |
| 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 |