**Database Design for Online Food Ordering System**

**Ronak Patel**

**The data model consists of three subject areas:**

**Restaurants & Customers**

**Menus**

**Orders**

A screenshot of a cell phone

Description automatically generated

Restaurants and Customers:-

It contains only two attributes, city\_name and zip\_code. If we operate in more than one country, we would also need a country dictionary that would be related to this table, but we won’t go into that here.

Next, we need a list of all the restaurants we operate. We’ll use the **restaurant** table for that. To keep things simple, we’ll only store each restaurant’s address and a reference to the **city** where it is located.

* customer\_name – The customer’s full name.
* city\_id – References the city where the customer lives.
* address – The customer’s address.
* contact\_phone – The customer’s phone number.
* email – The email address the customer used during the registration process.
* confirmation\_code – A confirmation code used during the registration process.
* password – The password selected by the customer for this app.

Menus:-

The first table is the **category** dictionary. It contains only one UNIQUE attribute, category\_name. This field will probably hold the usual menu categories, such as “drinks”, “starters”, “salads”, “sandwiches”, “pizza”, etc.

Next, we have the menu\_item table. It lists all items we have (or had) on any of our menus. For each item, we’ll store:

* item\_name – A name for that item, e.g. “chicken sandwich”.
* category\_id – References the category that the item belongs to, e.g. “sandwiches”.
* description – A description of that item. This should be the same as on the printed menu.
* ingredients – The ingredients used to produce that item and their quantities. This field could actually store a recipe.
* price – The current price for one item (e.g. one chicken sandwich).
* active – If the item is offered on the current menu.

Most restaurants have special, limited-time offers. They may also have some offers for an unlimited amount of time. We’ll use the offer table for these. For each one, we’ll have:

* date\_active\_from and date\_active\_to – Together, these define when this offer is active. If an offer has an unlimited duration or if it’s based on hours rather than days, these two attributes will contain NULL values. An example of this type of offer is “During the month of March, buy one curry and get one 50% off”.
* time\_active\_from and time\_active\_to – Defines the time of day an offer is valid – e.g. “Get a free coffee from 6-9 a.m. every day”.
* offer\_price – The price for that offer.

All menu items included in offers are stored in the **in\_offer** table. This table contains the UNIQUE pair of offer\_id – menu\_item\_id

**Orders :-**

The central table here is the **placed\_order** table.

* restaurant\_id – The ID of the restaurant related to that order.
* order\_time – A timestamp of when the order was placed.
* estimated\_delivery\_time – A timestamp of the planned delivery of this order.
* actual\_delivery\_time – A timestamp of when this order was actually delivered. It will be NULL until the food is delivered to the customer.
* delivery\_address – The address where the order should be delivered.
* customer\_id – The ID of the customer who placed that order. This attribute could contain a NULL value if the order was placed by someone who is not a registered app user.
* price – The price for all items included in that order.
* discount – The amount of discount (e.g. coupon or loyalty discount) applied to the price, if any.
* final\_price – The order price minus the discount.
* comment – Additional comments inserted by the customer when the order was placed. This could be additional delivery instructions or anything else the customer finds important.
* ts – A timestamp of when this record was inserted in the table.

The in\_order table lists all items or special offers that are included in an order. For each record in this table, we’ll store:

* placed\_order\_id – The ID of the related order.
* offer\_id – References the offer table, but only when one or more offers are included in this order. In that case, the menu\_item\_id attribute will be NULL.
* menu\_item\_id – References the menu\_item table, but only if this record is related to a menu item and not an offer.
* quantity – How many offers or menu items are included in this order.
* item\_price – The price of a single offer or menu item.
* price – The total price for this line, expressed as item\_price \* quantity.
* comment – Any comments inserted by the customer that relate specifically to that order item, e.g. “Please cut pizza into 8 slices”.

The **comment** table lets us support the insertion of multiple comments related to orders. For each comment, we’ll store the ID of the related order and the ID of the customer. We’ll also store a timestamp of when this comment was entered. We’ll also mark whether this comment was a complaint or a compliment; only one of these two can be set at one time. If none are set, then we’ll treat this comment as neutral.

The last two tables in our model are related to statuses we’ll assign to orders. The **status\_catalog** table contains a list of all possible UNIQUE status\_name attributes that we could assign to orders. The **order\_status** table contains all statuses that are assigned to orders. For each record in this table, we’ll store foreign keys related to order and status and the timestamp when this status was assigned.