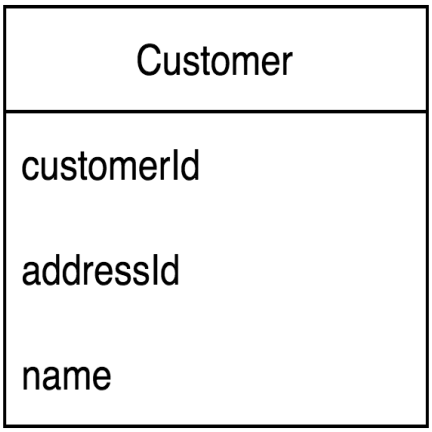
RESTAURANT FOOD ORDER DATA MODELING

- I will use any RDBMS : **MySQL** preferred.
- For given scenarios total 7 tables required. I haven't included the payments table here.
- All tables and information are explained below.

TABLES & INFO:

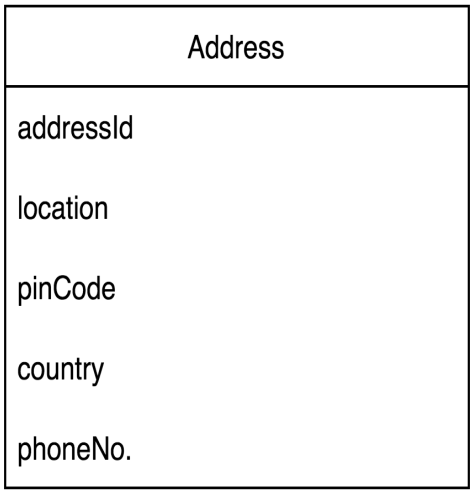
1. **Customer Table:** When a new user comes to the portal we can take and store the information about the user in the customer table. (I haven't Included DOB etc. kept it short).

   customerId : primary key , **addressId**: foreign key, **name**: varchar



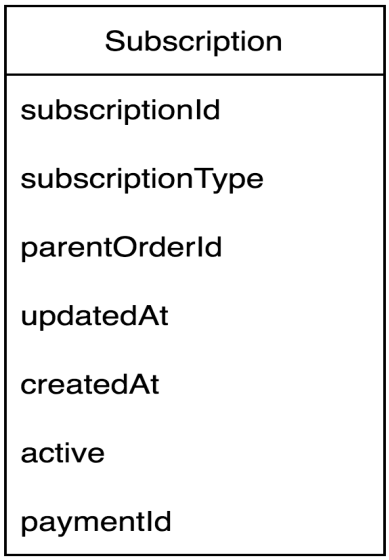
2. **Address Table:** single table to store information about the address/location of customer, restaurant, orderlocation.

   addressId: primary key, **location**: varchar, **pinCode**: varchar, **country**:varchar, **phoneNo**.: varchar



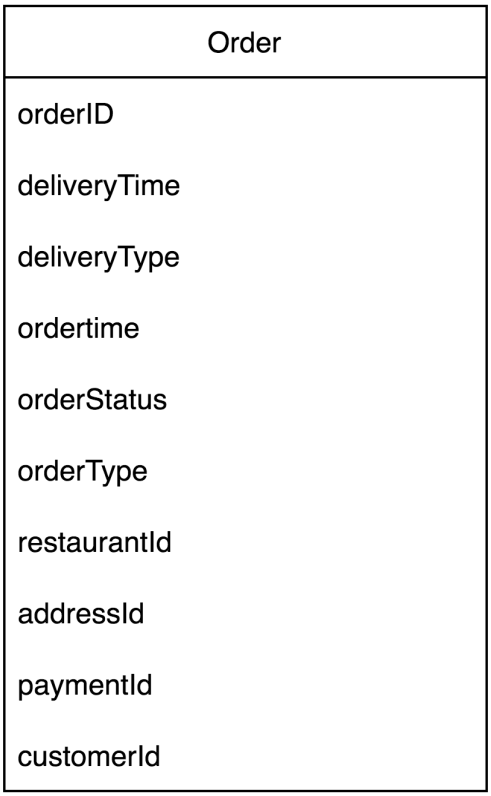
3. **Subscription Table:** In problem it's given to develop a feature so that user/customer can subscribe to meals. So, we have to store the information about the subscription. Field subscriptionType will store the information about what kind of subscription it is.(eg. Tiffin , breakfast etc ..). Before adding the data to the subscription table we need to create an order for it and store the orderId in parentOrderId. createdAt stores the datetime when subscription is created. updatedAt stores the datetime until subscription is valid. A cron job will check whether the subscription is expired or not if it is expired then change the active state of the subscription.

   subscriptionId: primary key, **subscriptionType**:, **parentOrderId**: foreign key, **updatedAt**: date time, **createdAt**: date time, **active**: boolean, **paymentId**: foreign key



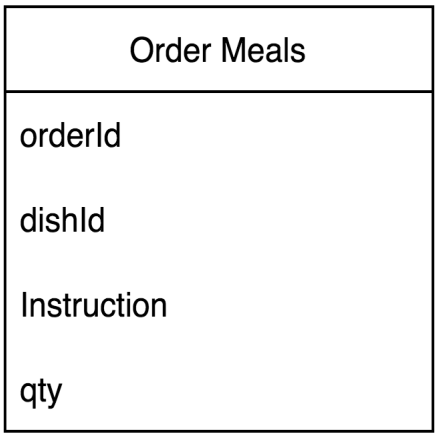
4. **Order Table:**deliveryTime is updated once the food is delivered. orderStatus stores the state whether the order is in progress, cancelled or delivered. deliveryType stores the mode of delivery (served at restaurant, via delivery boy).

orderId: primary key, **deliveryTime**: datetime,**deliveryType**: enum, **orderTime**: datetime, **orderStatus**: enum, **orderType**: enum, **restaurantId**: foreign key, **addressId**: foreign key, **paymentId**: foreign key, **customerId**: foreign key



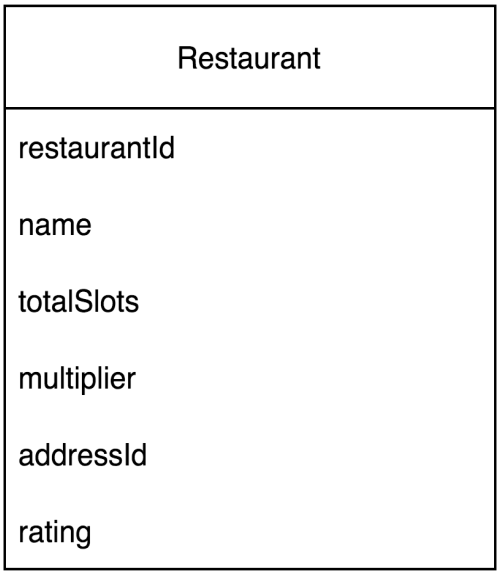
5. **Order Meals Table:** order meals table stores and maintains the meal/dish quantity & meal/dish type. The instruction field stores what customers mentioned and the exact recipe which they want.

orderId: keys, **dishId:** keys, **instruction:** varchar , **qty:** integer



6. **Restaurant Table:** restaurant table maintains info about all the available restaurants. Name is the field that is the name of the restaurant, multiplier is the number used for conversion. (eg: 1km can be covered in 8 minutes. Now multiplier will be 8). Ratings will have the user ratings. TotalSlots is an integer value indicating how many slots are available for cooking.

   restaurantId: primary key, **name**: varchar, **totalSlots**: integer, **multiplier**: double, **addressId**: foreign key, **rating**: double



7. **Dish Table:** Stores information about dishes and meals such as what type of dish it is either appetizer, dessert, main meal etc. . DishName is a varchar that stores the dish name.(eg curd, kadai paneer etc.) Description is a detailed information about the dish. Ingredients is a field that stores all the ingredients used in the dish.

dishId: primary key, **dishName:** varchar, **dishType:** enum, **description:** text, **ingredients:** varchar, **restaurantId:** foreign key