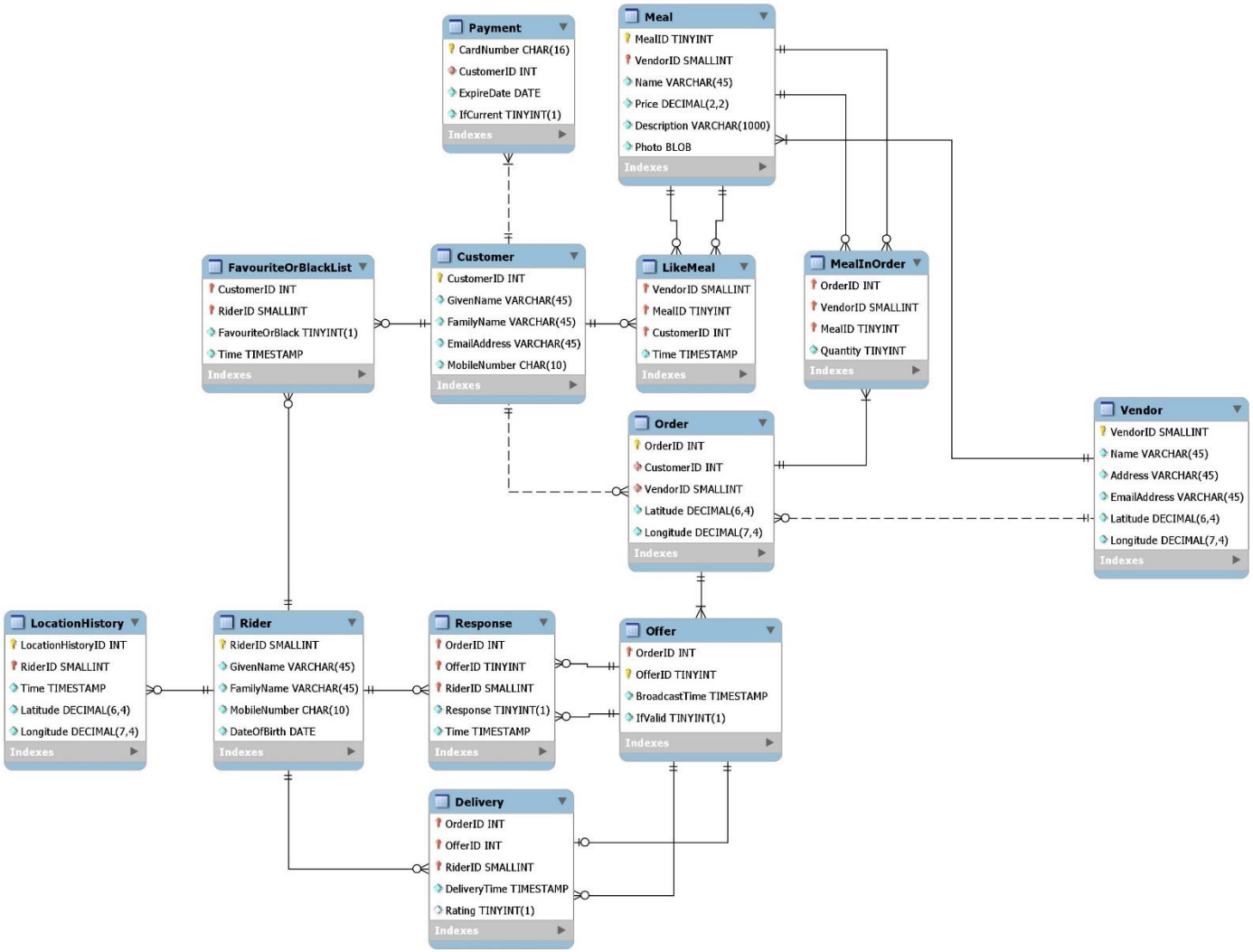
Payment
CardNumber CHAR(16)
CustomerID INT
ExpireDate DATE
IfCurrent TINYINT(1)
Indexes
Meal
MealID TINYINT
VendorID SMALLINT
Name VARCHAR(45)
Price DECIMAL(2,2)
Description VARCHAR(1000)
Photo BLOB
Indexes
FavouriteOrBlackList
CustomerID INT
RiderID SMALLINT
FavouriteOrBlack TINYINT(1)
Time TIMESTAMP
Indexes
Customer
CustomerID INT
GivenName VARCHAR(45)
FamilyName VARCHAR(45)
EmailAddress VARCHAR(45)
MobileNumber CHAR(10)
Indexes
LikeMeal
VendorID SMALLINT
MealID TINYINT
CustomerID INT
Time TIMESTAMP
Indexes
MealInOrder
OrderID INT
VendorID SMALLINT
MealID TINYINT
Quantity TINYINT
Indexes
Order
OrderID INT
CustomerID INT
VendorID SMALLINT
Latitude DECIMAL(6,4)
Longitude DECIMAL(7,4)
Indexes
Vendor
VendorID SMALLINT
Name VARCHAR(45)
Address VARCHAR(45)
EmailAddress VARCHAR(45)
Latitude DECIMAL(6,4)
Longitude DECIMAL(7,4)
Indexes
LocationHistory
LocationHistoryID INT
RiderID SMALLINT
Time TIMESTAMP
Latitude DECIMAL(6,4)
Longitude DECIMAL(7,4)
Indexes
Rider
RiderID SMALLINT
GivenName VARCHAR(45)
FamilyName VARCHAR(45)
MobileNumber CHAR(10)
DateOfBirth DATE
Indexes
Response
OrderID INT
OfferID TINYINT
RiderID SMALLINT
Response TINYINT(1)
Time TIMESTAMP
Indexes
Offer
OrderID INT
OfferID TINYINT
BroadcastTime TIMESTAMP
IfValid TINYINT(1)
Indexes
Delivery
OrderID INT
OfferID INT
RiderID SMALLINT
DeliveryTime TIMESTAMP
Rating TINYINT(1)
Indexes

Assumption:

1. It is assumed that this App is developed for Australia market. Thus the mobile number is fixed ten characters.
2. Each Meal belongs to a specific Vendor. We assume that each Vendor sells no more than a hundred Meals. Therefore, we use TINYINT for MealID and combine MealID and VendorID together as a composite primary key for the Meal table. This is also the reason that we store VendorID both in table MealInOrder and LikeMeal. The VendorID and MealID together are the composite foreign key refer to the composite primary key in the Meal table. This will also decrease the storage space of the MealID.
3. Similar to assumption 2, each Offer belongs to a specific Order and for an Order, there won't be more than a hundred Offer that will be generated. Hence, we used TINYINT for OfferID and combine it with OrderID together as the composite primary key for the Offer table. Again, in Response and Delivery table, we need to use OfferID and OrderID together as the composite foreign key.
4. It is allowed that no Rider responds the offer within the required time, which will be considered as no one accepts the Offer and a new Offer will be broadcasted. Thus the relationship between Offer and Response is one to optional many.
5. When a new Offer is broadcasted, because we are required to keep track of the Offer, the record of the old Offer will not be deleted from the database. This will lead to the fact that the invalid Offer will have no Delivery. Therefore, we use IfValid to indicate if the Offer is valid or not. And the relationship between Offer and Delivery is also one to optional many.
6. We assume that there won't be any history of location if there's no rider at all. Thus we use RiderID and LocationHistoryID together as composite primary key. This will also decrease the storage space of the LocationHistoryID.