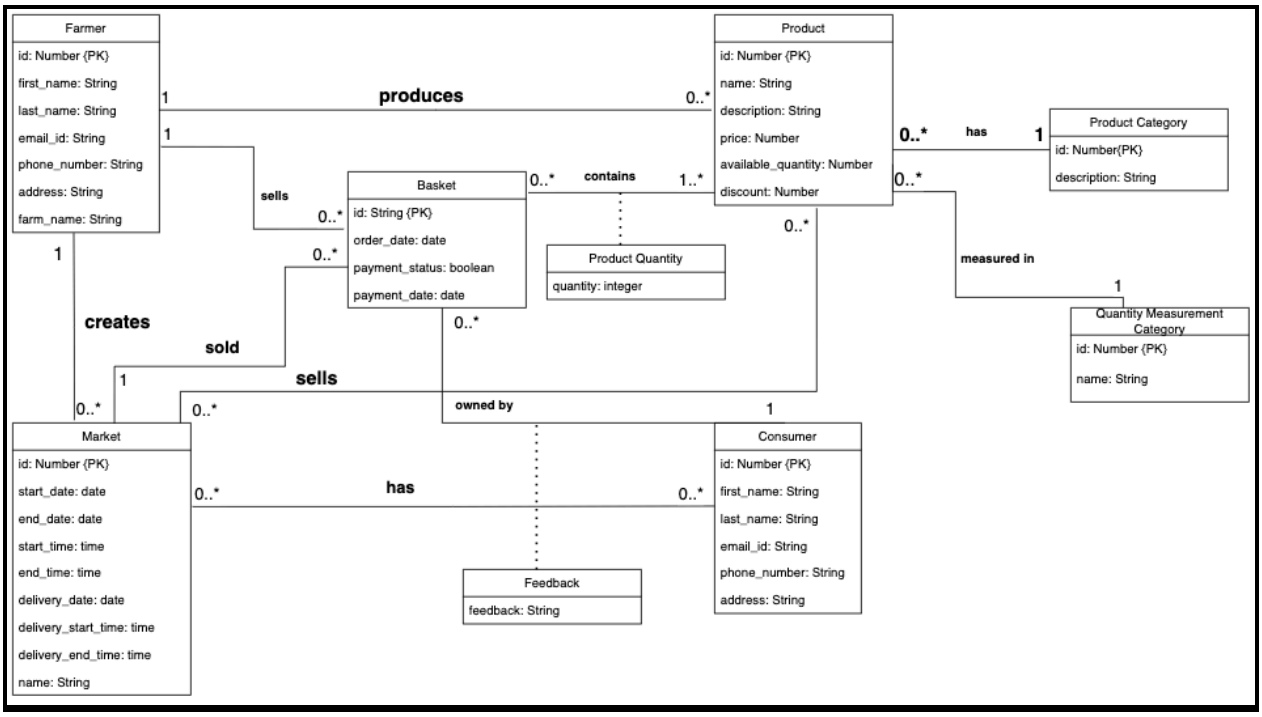

COMMENTS

- The market association with producer class describes the relationship of the producer and market from the question. The market association with basket provides data to compute the number of orders and the average amount of orders (as the basket contains order_date field to aid the calculation of average). The market association with customers aids in the computation of the number of customers per market, and the market association (sells) with product aids in the computation of products sold per market.
- The association between consumer and basket provides data for computation of a consumer's purchases and amount spent, while the market and consumer association provides data for computation of a list of markets and producers patronized by a consumer.
- Some assumptions made are:
    - A product has a measurement category, as some products could be measured in Kg while some in litres or cartons.
    - A customer can give feedback for each basket containing the products they ordered.
    - Due to the fact that a basket contains various products and each product can be bought in various quantities but not below the number of available quantities, an association between Basket and Product will provide the quantity of a specific product ordered by a customer in a specific basket.
- Every other association represents the constraint from the question or implied by the question.

3rd normal form normalized relational schema

Farmer(#id: Number, first_name: String, last_name: String, email_id: String, phone_number: String, address: String, farm_name: String)
Constraint:
- email should be unique
- phone number, address, farm name and first name must not be null

Market(#id: Number, start_date: Date, end_date: Date, start_time: Time, end_time: Time, delivery_date: Date, delivery_start_time: Time, delivery_end_time: Time, name: String, farmer_id => farmer)
Constraint:
- start_date, end_date, start_time, farmer_id and end_time can not be null

Basket(#id: Number, order_date: Date, payment_status: Boolean, payment_date: Date, farmer_id => farmer, market_id => market, consumer_id => consumer,)
Constraint:
- order_date, payment_status, farmer_id and market_id can not be null

ProductQuality(#basket_id => basket, #product_id => product, quantity: Number)
Constraint:
- quantity cannot be null and should be equal or greater than 0.

Product(#id: Number, name: String, description: String, price: Number, available_quantity: Number, discount: Number, category => ProductCategory, measurement => QuantityMeasurementCategory, farmer_id => farmer)
Constraint:
- price can not be null and must be > 0
- available_quantity, and discount cannot be null and should be equal or greater than 0
- measurement and farmer_id can not be null

Consumer(#id: Number, first_name: String, last_name: String, email_id: String, phone_number: String, address: String)

Constraint:
- email_id should be unique
- phone number, first name and address should not be null

ProductCategory(#id: Number, description: String)
Constraint:
- description can not be null

QuantityMeasurementCategory(#id: Number, name: String)
Constraint:
- name can not be null

Feedback(#basket_id => basket, consumer_id => consumer)

ProductsSoldInMarket(market_id => market, product_id => product)

MarketAttendance(market_id => market, consumer_id => consumer)