DBMS-2006 Final Project Milestone One
Project Proposal

NOTE: Students are not permitted to have a project based on an academic environment *(students, courses, books, instructors....)*. Project must be unique and cannot be anything that was used in this or previous courses. **Students found using AI to generate the Project Proposal will receive an automatic zero grade for the project**.

Developer Name: Dongok Yang

Project Name: K-pop merchandise stock management system

Date: 2024 -11-13

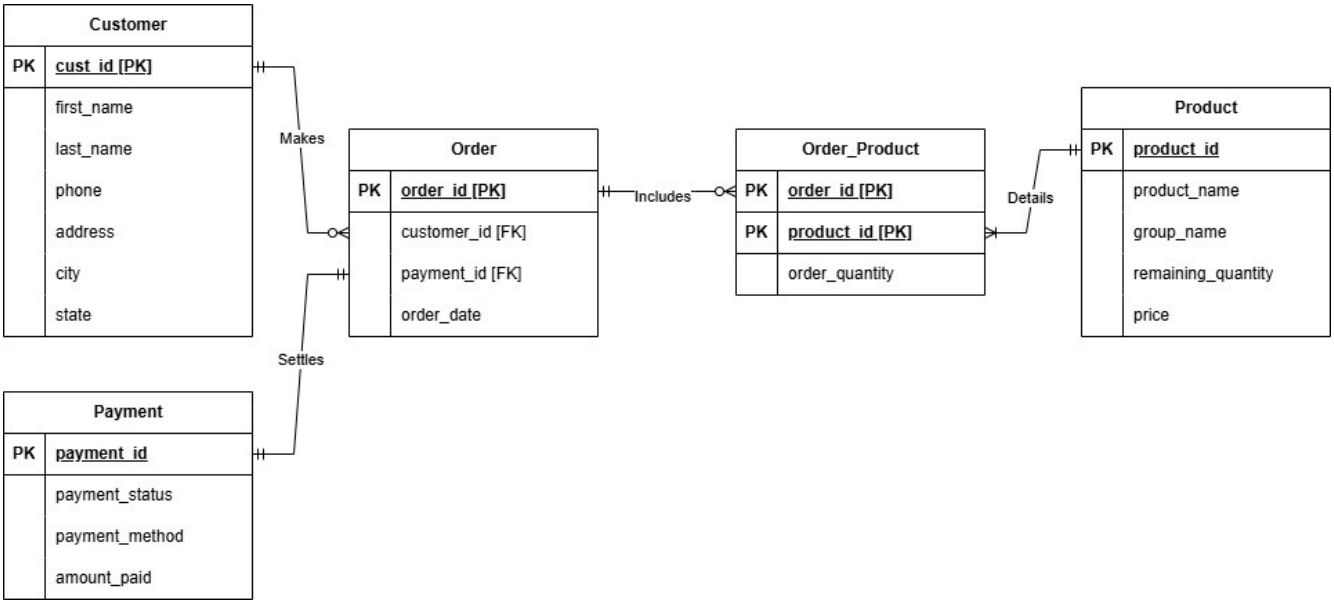
---

1. Project description
   This project is designed to handle the operation of a K-pop merchandise business. The system will mainly handle inventory management, customer orders, and payments. It will allow shop owners to manage stock efficiently.
   I chose this project because I have recently launched a K-pop merchandise business and realized the need for a better system than using excel to manage everything. This project not only allows me to apply the lessons from school to a real-word application, but also gives me to see the impact of the database through the eyes of a business owner.

2. Business Case
   Currently, I'm using Google Sheets to manage all my data such as stock levels and store customer information. It has some advantages, like being easy to read and accessible from anywhere since it's stored in the cloud. However, doing everything manually is time consuming and can lead to errors. By creating a system to handle the database, I can reduce the amount of workload and lower the chance of making mistakes.

3. Business Rules/Assumptions
   A customer can make an account without making an order and make an order later.
   A customer can place multiple orders, but each order is linked to exactly one customer.
   Each order can include multiple products, and each product can belong to multiple orders. And this creates a many-to-many relationship.
   An Order can exist without any Product. Which means an order can be created first, and products can be added to it later.
   Each product included in an order must be associated with its corresponding product details.
   The order_quantity must be greater than 0 and can’t exceed the remaining_quantity of the product.
   Payment settles exactly one Order and each order is settled by exactly one payment.

4. Entity Relationship Diagram


Customer
PK
cust_id [PK]
first_name
last_name
phone
address
city
state
Makes
Order
PK
order_id [PK]
customer_id [FK]
payment_id [FK]
order_date
Includes
Order_Product
PK
order_id [PK]
PK
product_id [PK]
order_quantity
Details
Product
PK
product_id
product_name
group_name
remaining_quantity
price
Settles
Payment
PK
payment_id
payment_status
payment_method
amount_paid