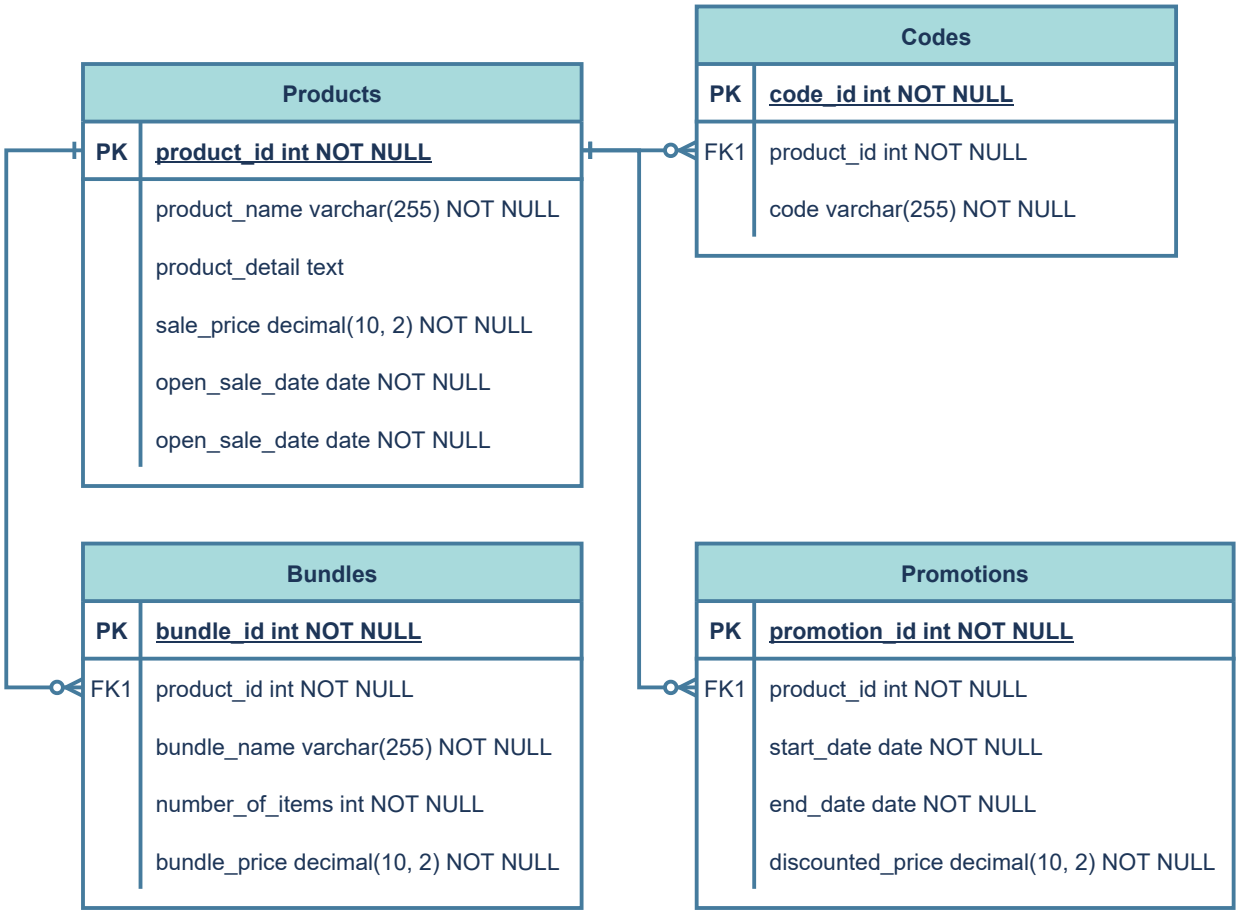

The reason in the design of database for the sale item code system, I have chosen to use four tables to represent the different entities involved in the system.

The "products" table is used to store information about the items that are for sale, such as the product name, product details, sale price, and the open and end sale date.

The "codes" table is used to store the codes that customers will receive after purchasing a product. Each code is associated with a specific product, as represented by the foreign key "product_id" in the "codes" table.

The "promotions" table is used to store information about any promotions that are running on specific products, such as the start and end date of the promotion, and the discounted price during that time. Each promotion is associated with a specific product, as represented by the foreign key "product_id" in the "promotions" table.

The "bundles" table is used to store information about any bundle sales that are running on specific products, such as the bundle name, number of items in the bundle, and bundle price. Each bundle is associated with a specific product, as represented by the foreign key "product_id" in the "bundles" table.

The reason I have chosen to design the database in this way is to ensure that the data is normalized and stored in separate tables to reduce data redundancy and improve data integrity.
Each table has a specific purpose and stores specific information, and the foreign keys in each table are used to establish relationships between the tables.

In addition, this design makes it easy to query the database for specific information and to update or delete information without affecting the data in other tables.