

INFO 6210 Data Management and Database Design

PROJECT SUBMISSION - 2

DATABASE DESIGN FOR REUSABLE ITEMS ON E-COMMERCE PLATFORM

TEAM DETAILS:

Team Name: Thrift Green

Team Members:

- Amogha Shettar (NUID : 001585117)
- Amoolya Shettar (NUID : 001582650)
- Giang Vu (NUID :001537937)

Introduction:

Items such as furniture, clothes, footwear, containers and cans have increasingly been thrown away even when they can be reused. There is a need for an online system where all these records can be stored, that can be used again by the customers who are willing to buy them at a cheaper rate.

Problem statement:

- Currently, most users don't have access to an online local store for used items. The buyer usually goes to a local thrift store if they want to look for used/2nd-hand items.
- Other online platforms usually are too general (like Amazon) with too much information or lack severe security/payment options (like Craigslist).
- The site is post-oriented instead of item-oriented (ie, you cannot search based on item filter, see item quantities, etc)
- Users cannot see the buyer/seller information (including selling posts, review, and other things) so mostly it is anonymous.
- Users cannot make an online payment and get the post updated automatically. They will either need to remove/update the post themselves.

Objectives:

To design a database for reusable items which should have the following features:

- Users must be able to register by entering their details namely username, password, phone number, and email.
- System must be able to store all the details of the users and the used/2nd-hand items.
- Users must be able to login using their username and password.
- Users must be able to add and update the details of the items.
- Users must be able to view the details of the items and sellers.
- Users must be able to make a deal for the reusable items.
- Users must be able to view the details of the ordered items for payment purposes.
- Users must be able to view the payment details.

Database design document:

- Currently, most users don't have access to an online local store for used items. The buyer usually goes to a local thrift store if they want to look for used/2nd-hand items
 - + The database design with current 11 ENTITIES should support the whole online, automated process from customer/seller signing up -> search items -> transactions process -> shipping process with shippers also.
- The other sites are post-oriented instead of item-oriented (ie, you cannot search based on item filter, see item quantities, etc)
 - + We have CATEGORY Entity, that has PK as CATEGORY_ID. The ITEM entity will have CATEGORY_ID as Foreign Key so that user can search item using category filter (for furniture, electronic device, etc)
- Users cannot see the buyer/seller information so mostly it is anonymous.
 - + We have a separate seller and customer entity with related personal attributes so that users can retrieve info about buy/seller given their id (C_ID, S_ID)
 - + Customer/Seller Entity each have a UNIQUE key as C_EMAIL & S_EMAIL to prevent duplicate entry/record when the user signs up. This will help avoid anomalies in our relational database model.
 - + Item Entity will have S_ID & R_ID as FOREIGN KEY so user can retrieve and fetch data from SELLER Entity & REUSABLE Entity regarding seller info & item description
 - + Each Item has an S_ID (seller ID) as Foreign Key so users can always retrieve a list of items being sold by the same seller.
- Users cannot make an online payment & handle the shipment easily
 - + We have a payment and cart entity to allow the user to choose checkout method/view cart items, costs, Each payment and cart will have an unique payment id and cart id. This can be used as a key for connecting to customer entity data.
 - + Items will be put into CART which will be ordered by the customers by having CART_ID as a FK in ORDER entity.
 - + Each Shipment assigned for a specific shipper can be viewed with SHIPPER_ID as FK in SHIPMENT Entity

```

    erDiagram
        REUSABLE ||--o{ ITEM : "has"
        CATEGORY ||--o{ ITEM : "has"
        CART ||--o{ ITEM : "has"
        PAYMENT ||--o{ ORDER : "has"
        ORDER ||--o{ SELLER : "has"
        ORDER ||--o{ SHIPMENT : "has"
        ORDER ||--o{ SHIPPER : "has"
        CUSTOMER ||--o{ ADDRESS : "has"
        ORDER ||--o{ CUSTOMER : "has"
        ORDER ||--o{ SELLER : "has"
        ORDER ||--o{ SHIPMENT : "has"
        ORDER ||--o{ SHIPPER : "has"
  
```

The diagram shows the following tables and their attributes:

- REUSABLE**: R_ID (PK), NUMBER_OF_TIMES_USED, DESCRIPTION
- CATEGORY**: CATEGORY_ID (PK), CATEGORY_NAME
- CART**: CART_ID (PK), NUMBER_OF_ITEMS, TOTAL_COST, ITEM_ID (FK)
- PAYMENT**: PAYMENT_ID (PK), AMOUNT, PAYMENT_MODE, DATE
- ITEM**: ITEM_ID (PK), ITEM_NAME, PRICE, S_ID (FK), R_ID (FK), CATEGORY_ID (FK)
- ORDER**: ORDER_ID (PK), ORDER_DATE, DELIVERY_STATUS, C_ID (FK), CART_ID (FK), PAYMENT_ID (FK)
- SELLER**: S_ID (PK), S_NAME, S_PASSWORD, S_EMAIL, S_PHONE, ADDRESS_ID (FK)
- SHIPPER**: SHIPPER_ID (PK), CONTACT_NUMBER
- CUSTOMER**: C_ID (PK), C_NAME, USERNAME, PASSWORD, DATE_OF_BIRTH, C_PHONE, C_EMAIL, ADDRESS_ID (FK)
- ADDRESS**: ADDRESS_ID (PK), ADDRESS_LINE1, ADDRESS_LINE2, AREA_CODE, STATE, COUNTRY
- SHIPMENT**: SHIPMENT_ID (PK), SHIPMENT_DATE, COURIER, ADDRESS_ID (FK), SHIPPER_ID (FK), ORDER_ID (FK)

1. SELLER ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
S_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
S_NAME	VARCHAR(40)	NOT NULL
S_PASSWORD	VARCHAR(16)	NOT NULL
S_EMAIL	VARCHAR(40)	UNIQUE KEY, NOT NULL
S_PHONE	INTEGER	NOT NULL
ADDRESS_ID	INTEGER	A Foreign Key which REFERENCES ADDRESS_ID from ADDRESS ENTITY. This is used to fetch sellers's address details. NOT NULL

2. CUSTOMER ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
C_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
C_NAME	VARCHAR(40)	NOT NULL
PASSWORD	VARCHAR(16)	NOT NULL
USERNAME	VARCHAR(40)	UNIQUE KEY, NOT NULL
DATE_OF_BIRTH	TIMESTAMP	CURRENT_TIMESTAMP
C_EMAIL	VARCHAR(40)	UNIQUE KEY, NOT NULL
C_PHONE	INTEGER	NOT NULL
ADDRESS_ID	INTEGER	A Foreign Key which REFERENCES ADDRESS_ID from ADDRESS ENTITY. This is used to fetch customer's address details. NOT NULL

3. ITEM ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
ITEM_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
ITEM_NAME	VARCHAR(40)	NOT NULL
PRICE	INTEGER	NOT NULL
S_ID	INTEGER	A Foreign Key which REFERENCES S_ID from SELLER ENTITY. This is used to fetch seller's details. NOT NULL
R_ID	INTEGER	A Foreign Key which REFERENCES R_ID from REUSABLE ENTITY. This is used to fetch reusable item details. NOT NULL
CATEGORY_ID	INTEGER	A Foreign Key which REFERENCES CATEGORY_ID from CATEGORY

		ENTITY. This is used to fetch category details. NOT NULL
--	--	---

4. REUSABLE ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
R_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
NUMBER_OF_TIMES_USED	INTEGER	NOT NULL
DESCRIPTION	VARCHAR(80)	NOT NULL

5. CATEGORY ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
CATEGORY_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
CATEGORY_NAME	VARCHAR(40)	NOT NULL

6. CART ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
CART_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
TOTAL_COST	INTEGER	NOT NULL
NUMBER_OF_ITEMS	INTEGER	NOT NULL
ITEM_ID	INTEGER	A Foreign Key which REFERENCES ITEM_ID from ITEM ENTITY. This is used to fetch item's details. NOT NULL

7. PAYMENT ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
PAYMENT_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
PAYMENT_MODE	VARCHAR(20)	NOT NULL
AMOUNT	INTEGER	NOT NULL
DATE	TIMESTAMP	CURRENT_TIMESTAMP

8. ORDER ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
ORDER_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
ORDER_DATE	TIMESTAMP	CURRENT_TIMESTAMP
DELIVERY_STATUS	VARCHAR(40)	NOT NULL
C_ID	INTEGER	A Foreign Key which REFERENCES C_ID from CUSTOMER ENTITY. This is used to fetch customer's details. NOT NULL
CART_ID	INTEGER	A Foreign Key which REFERENCES CART_ID from CART ENTITY. This is used to fetch cart details. NOT NULL
PAYMENT_ID	INTEGER	A Foreign Key which REFERENCES PAYMENT_ID from PAYMENT ENTITY. This is used to fetch payment details. NOT NULL

9. SHIPMENT ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
SHIPMENT_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
SHIPMENT_DATE	TIMESTAMP	CURRENT_TIMESTAMP
COURIER	VARCHAR(40)	NOT NULL

ADDRESS_ID	INTEGER	A Foreign Key which REFERENCES ADDRESS_ID from ADDRESS ENTITY. This is used to fetch customer's address details. NULL ALLOWED
SHIPPER_ID	INTEGER	A Foreign Key which REFERENCES SHIPPER_ID from SHIPPER ENTITY. This is used to fetch shipper's details. NOT NULL
ORDER_ID	INTEGER	A Foreign Key which REFERENCES ORDER_ID from ORDER ENTITY. This is used to fetch order details. NOT NULL

10. SHIPPER ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
SHIPPER_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
CONTACT_NUMBER	TIMESTAMP	CURRENT_TIMESTAMP

11. ADDRESS ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
ADDRESS_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
ADDRESS_LINE1	VARCHAR(40)	NOT NULL
ADDRESS_LINE2	VARCHAR(40)	NOT NULL
AREA_CODE	INTEGER	NOT NULL
STATE	VARCHAR(20)	NOT NULL
COUNTRY	VARCHAR(24)	NOT NULL