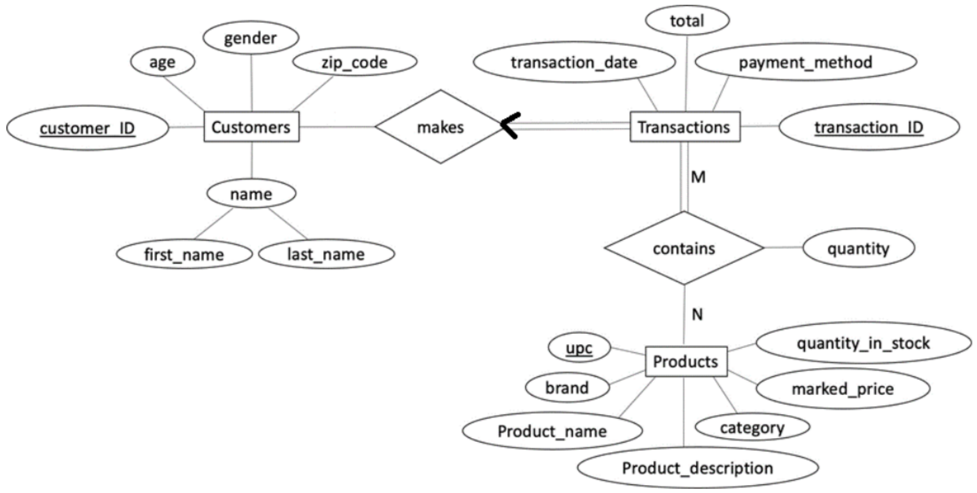
CS550 Fall 2024 Project 1 Part 2

Name:- Ruchita Darur
GNo:- G01514822

Project Description:-



Relational Schemas For The TABLE Customers

Primary Key:- **CUSTOMER_ID**

CUSTOMER_ID	FIRST_NAME	LAST_NAME:	AGE	GENDER	ZIP_CODE
INT	VARCHAR(50)	VARCHAR	INTEGER	CHAR(1) ('M', 'F')	VARCHAR(10)
Primary Key					

Relational Schemas For The TABLE Transactions

Primary Key:- **TRANSACTION_ID**

TRANSACTION_ID	TRANSACTION_DATE	TOTAL	PAYMENT_METHOD	CUSTOMER_ID
INTEGER	DATE	DECIMAL (10,2)	INTEGER	INTERGER
				FOREIGN KEY

- customer_ID INTEGER (Foreign Key referencing Customers(customer_ID))

Relational Schemas For The TABLE Transactions

Primary Key:- **upc**

UPC	PRODUCT_NAME	BRAND	PRODUCT_DESCRIPTION	CATEGORY	MARKED_PRICE	QUANTITY_IN_STOCK
VARCHAR (12)	VARCHAR (50)	VARCHAR (50)	VARCHAR (225)	VARCHAR (50)	VARCHAR (50)	INTEGER
PRIMARY KEY						

Relational Schemas For The TABLE contains

TRANSACTION_ID	UPC	QUANTITY
INTEGER	INTEGER	INTERGER
PRIMARY KEY/FOREIGN KEY	FOREIGN KEY	

a) Convert the above ER diagram into relational schemas. In a PDF file, specify the followings for each relation:

(i) The Name Of The Relation

Answer: - The name of the relation from the ER Diagram:

- **"makes"** is the initial relation that relates the "Transactions" and "Customers" entities.

- **"contains"** is the following relation that relates the "Transactions" and "Products" entities.

(ii) The Names of Its Attributes

Answer: - Here's a list of all the attributes present in the given ER diagram:

♦ Customer:

- customer_ID

- name
  - First_name
  - Last_name
- age
- gender
- zip_code

♦ Transactions:
- transaction_ID
- transaction_date
- total
- payment_method

♦ Products:
- upc
- Brand
- product_name
- product_description
- category
- marked_price
- Quantity_in_stock
  - The attribute **quantity** is relationship ("Contains") where Transaction contains Products.

♦ Contain:
- transaction_ID
- upc
- quantity

(iii) The Domain Of each attribute

Answer:

Customers Table

♦ customer_ID:
  - Domain: Interger.
  - Most likely an integer that acts as the main key to identify every consumer in a unique way.

♦ name: A composite attribute that is written down further split into:
- First_name:
  - Domain: VARCHAR(50)
  - Representing the first name of the customer.
- Last_name
  - Domain: VARCHAR(50)
  - Indicates the last name of the customer.

♦ age:

  - Domain: INTEGER
  - This represents the customer's age.
- gender:
  - Domain: CHAR(1).
  - ('M', 'F') (Character) (e.g., 'M' or 'F')
  - A single character representing gender, e.g., 'M' or 'F'
- zip_code:
  - Domain:VARCHAR(10).
  - It is formatted to match the region’s postal code.

Transactions Table

- transaction_ID :
  - Domain: INTEGER.
  - Each transaction is assigned its unique identification.
- transaction_date:
  - Domain: INTEGER.
  - Represents a reference to the Customers table.
  - to record the exact day of the transaction.
- total:
  - Domain: Decimal (10,2).
  - Indicating the total money of the transaction.
- payment_method:
  - Domain: Integer.
  - Represents payment_method can only have ‘1’ or ‘2’ or ‘3’ as values

Products Table

- Upc :
  - Domain: Varchar(12)
  - Upc or barcode - Commonly used to uniquely identify products.
- Brand:
  - Domain: Varchar(50)
  - Represents the brand of the product.
- product_name:
  - Domain: Varchar(100)
  - Identifies the name of the product.
- product_description:
  - Domain: Varchar(225)
  - Representing the product’s description.
- category:
  - Domain: Varchar(50)
  - Indicates the category of the product.
- marked_price:
  - Domain:. Decimal (10,2)

  - Specifies the Product’s price.
- Quantity_in_stock:
  - Domain: **Integer.**
  - Displays the number of items in inventory.

Contains Table

- transaction_ID
  - Domain: INTEGER
- upc
  - Domain: INTEGER
- quantity
  - Domain: INTEGER

(iv) The Primary Key

- Customer:
  - Primary Key: **customer_ID**
- Transactions:
  - Primary Key: **transaction_ID**
- Products:
  - Primary Key: **upc**

(v) The Foreign Key(S)

- **Customers → Transactions**:

  The foreign key Customer_ID in the **Transactions** table references the **Customers** table.

- **Transactions → Products**:

  The junction table **Transaction Product** contains two foreign keys, Transaction_ID (references **Transactions**) and UPC (references **Products**), representing the many-to-many relationship between transactions and products.

Table Structure:- SQL QUERY

```
DROP TABLE Customers CASCADE CONSTRAINTS;
DROP TABLE Transactions CASCADE CONSTRAINTS;
DROP TABLE Products CASCADE CONSTRAINTS;
DROP TABLE Contains CASCADE CONSTRAINTS;


CREATE TABLE Customers (
   customer_ID NUMBER PRIMARY KEY,
   first_name VARCHAR2(50),
   last_name VARCHAR2(50),
   age NUMBER CHECK (age > 0 AND age < 120),
   gender CHAR(1) CHECK (gender IN ('M', 'F')),
   zip_code VARCHAR2(10));


CREATE TABLE Transactions (
        transaction_ID NUMBER PRIMARY KEY,
        customer_ID NUMBER,
        transaction_date DATE,
        total NUMBER(10, 2) CHECK (total >= 0),
        payment_method NUMBER CHECK (payment_method IN (1, 2, 3)),
        FOREIGN KEY (customer_ID) REFERENCES Customers(customer_ID) ON DELETE
     CASCADE);


CREATE TABLE Products (
        upc VARCHAR2(12) PRIMARY KEY,
        Product_name VARCHAR2(100),
        brand VARCHAR2(50),
        category VARCHAR2(50),
        Product_description VARCHAR2(255),
        quantity_in_stock NUMBER CHECK (quantity_in_stock >= 0),
        marked_price NUMBER(10, 2) CHECK (marked_price >= 0));


CREATE TABLE Contains (
        transaction_ID NUMBER,
        upc VARCHAR2(12),
        quantity NUMBER CHECK (quantity > 0),
        PRIMARY KEY (transaction_ID, upc),
        FOREIGN KEY (transaction_ID) REFERENCES Transactions(transaction_ID) ON
     DELETE CASCADE,
        FOREIGN KEY (upc) REFERENCES Products(upc) ON DELETE CASCADE);
```