**CS550 Fall 2024 Project 1 Part 2**

**Name:- Ruchita Darur**

**GNo:- G01514822**

**Project Description:-**

**A diagram of a product

Description automatically generated**

**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);