

Furniture Shop Management System

Final Project for SQL Model

By Tanaya Kamble

1. Description:

The following database schema is designed to function as a backend storage database for a web application built to manage a Furniture Shop.

By storing information in a relational database, all the tasks related to the daily functioning of the furniture shop can be performed easily and much more efficiently.

Some of the benefits of using this system to store data over traditional paper registers are as follows:

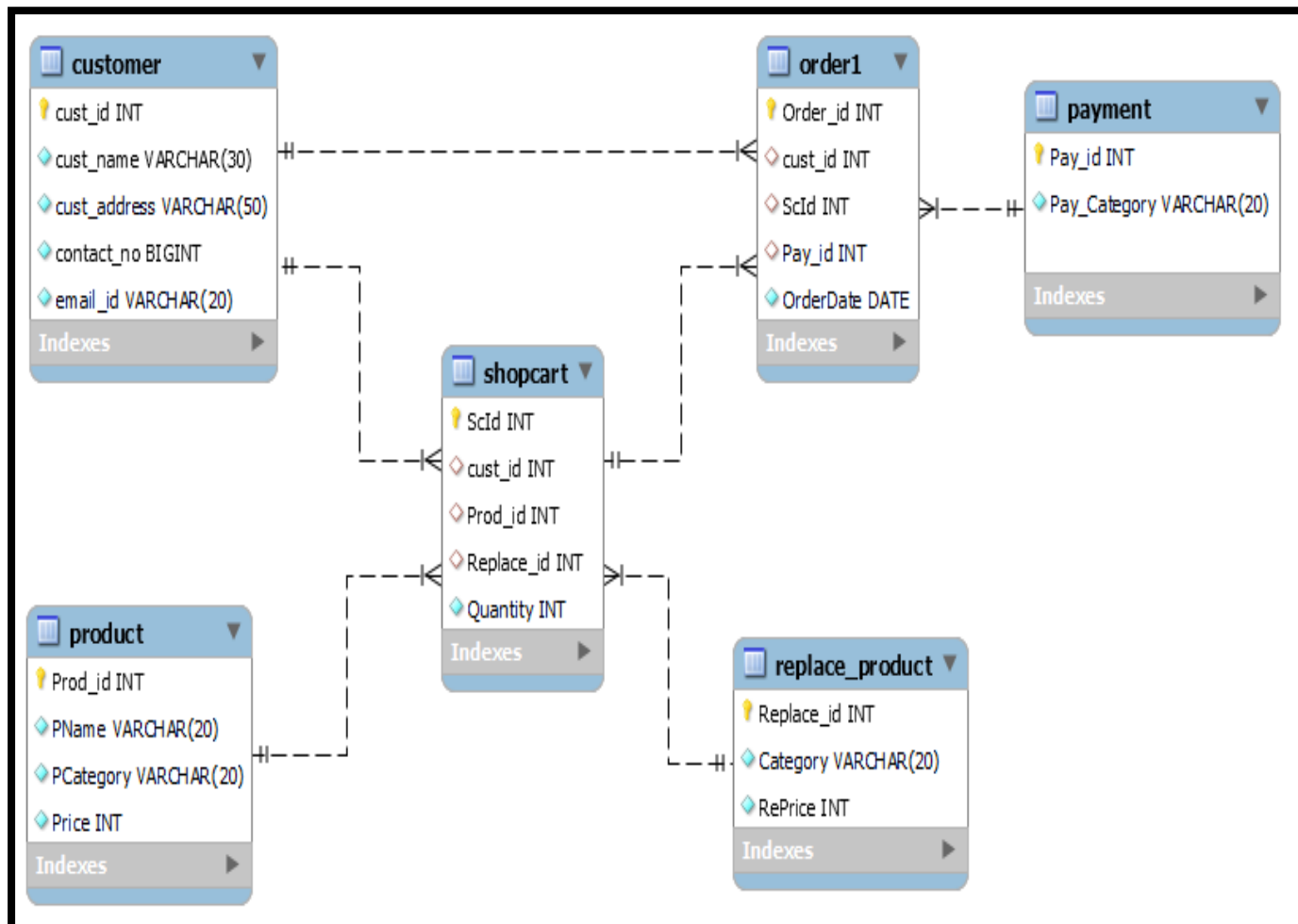
- ☐ Maintaining customer details and their account is easy and efficient.
- ☐ RDBMS provides many ways to analyze available data, thus helping in making more informed decisions about inventory management and other aspects of Furniture shop management.

This database contains 6 tables:

1. Customer
2. Product
3. Replace Product
4. Shop cart
5. Payment
6. Order1

How these tables/entities are related to each other is shown pictorially on the next page through ER diagram, i.e., Entity Relationship Diagram.

2. E-R Diagram:



3. Table description:

1. Customer:

Field	Type	Null	Key	Default	Extra
cust_id	int	NO	PRI	NULL	auto_increment
cust_name	varchar(30)	NO		NULL	
cust_address	varchar(50)	NO		NULL	
contact_no	bigint	NO		NULL	
email_id	varchar(20)	NO	UNI	NULL	

2. Product:

Field	Type	Null	Key	Default	Extra
Prod_id	int	NO	PRI	NULL	
PName	varchar(20)	NO		NULL	
PCategory	varchar(20)	NO		NULL	
Price	int	NO		NULL	

3. Replace Product:

Field	Type	Null	Key	Default	Extra
Replace_id	int	NO	PRI	NULL	
Category	varchar(20)	NO		NULL	
RePrice	int	NO		NULL	

4. Shop Cart:

Field	Type	Null	Key	Default	Extra
Scid	int	NO	PRI	NULL	
Cust_id	int	YES	MUL	NULL	
Prod_id	int	YES	MUL	NULL	
Replace_id	int	YES	MUL	NULL	
Quantity	int	NO		NULL	

5. Payment:

Field	Type	Null	Key	Default	Extra
Pay_id	int	NO	PRI	NULL	
Pay_Category	varchar(20)	NO		NULL	

6. Oorder1:

Field	Type	Null	Key	Default	Extra
Order_id	int	NO	PRI	NULL	
cust_id	int	NO	MUL	NULL	
Scid	int	NO	MUL	NULL	
Pay_id	int	NO	MUL	NULL	
OrderDate	date	NO			

4. Command:

a) Create Database:

create database FSMS;

b) Select Database:

use FSMS;

c) Create table Customer:

create table Customer(cust_id int primary key auto_increment, cust_name varchar(30) not null, cust_address varchar(50) not null, contact_no bigint not null, email_id varchar(20) not null unique);

d) Create table Product:

create table Product(Prod_id int primary key, PName varchar(20) not null, PCategory varchar(20) not null, Price int not null);

e) Create table Replace Product:

create table Replace_Product(Replace_id int primary key, Category varchar(20) not null, RePrice int not null);

f) Create table Shop cart:

create table Shopcart(ScId int primary key, cust_id int, Prod_id int, Replace_id int, Quantity int not null, foreign key(Prod_id) references Product(Prod_id), foreign key(cust_id) references Customer(cust_id), foreign key(Replace_id) references Replace_Product(Replace_id));

g) Create table Payment:

create table Payment(Pay_id int primary key, Pay_Category varchar(20) not null);

h) Create table Order1:

create table Order1(Order_id int primary key, cust_id int, ScId int, Pay_id int, OrderDate date not null, foreign key(cust_id) references Customer(cust_id), foreign key(ScId) references Shopcart(ScId), foreign key(Pay_id) references Payment(Pay_id));

i) Populate Customer table:

insert into Customer(cust_name, cust_address, contact_no, email_id) values
(('Vihan Kurte', 'Ratnagiri', 9087656743, 'vihank06@gmail.com'),
(('Natasha Patil', 'Mirjole', 9908765432, 'natashap22@gmail.com'),
(('Pravara Kamble', 'Kalyan', 9580764323, 'pravara03@gmail.com'),
(('Harsh Joshi', 'Pune', 9780654323, 'harshj09@gmail.com'),
(('Astha Kamble', 'Thane', 9678023451, 'astha25@gmail.com'),
(('Sarvesh Hatiskar', 'Mumbai', 9678543210, 'sarvesh67@gmail.com'),
(('Isha Joshi', 'Kalyan', 9456789032, 'isha1234@gmail.com'),
(('Shivansh Thakur', 'Mulund', 9554341229, 'shivansh77@gmail.com'),
(('Anika Sawant', 'Kurla', 9675843567, 'anika7890@gmail.com'),
(('Gauri Palkar', 'Thane', 8907655456, 'gauri45@gmail.com');

j) Populate Product table:

Insert into Product values

(1, 'King Bed', 'Bed', 30000),
(2, 'Office Chair', 'Chair', 19990),
(3, 'Dining Table Set', 'Dining Furniture', 25000),
(4, 'Sliding door', 'Wardrobes', 35000),
(5, 'Sofa Cum Bed', 'Sofas', 39990),
(6, 'Bookcase with drawer', 'Bookcases units', 6490),
(7, 'Coffee table', 'Tables', 2690),
(8, 'Mirrored wardrobes', 'Wardrobes', 20000),
(9, 'Wall shelf', 'Shelving units', 8660),
(10, '3-seat sofa', 'Sofas', 20990);

k) Populate Replace Product table:

insert into Replace_Product values

(1, 'Bed', 5000),
(2, 'Chair', 2000),
(3, 'Sofas', 4000),
(4, 'Bookcases unit', 1000),
(5, 'Tables', 500),
(6, 'Shelving unit', 1000),
(7, 'Dining furniture', 3000),
(8, 'Wardrobes', 5000),
(9, 'Swing Chairs', 500),
(10, 'TV Unit', 800);

l) Populate Shop cart table:

insert into Shopcart(Scid, cust_id, Prod_id, Replace_id, Quantity) values

(1,5,6,4,1),
(2,7,9, null,3),
(3,2,4,8,1),
(4,10,1,3,1),
(5,4,7, null,2),
(6,1,5,3,1),
(7,2,6,4,2),
(8,4,10, null,1),
(9,1,3,7,1),
(10,6,2, null,4);

m) Populate Payment table:

insert into Payment values

(101, 'Cash On Delivery'),

(102, 'Debit Card'),

(103, 'Credit Card'),

(104, 'Cheque'),

(105, 'Net Banking');

n) Populate Order1 table:

insert into Order1 values

(1,2,3,101, '2023-08-18'),

(2,4,8,103, '2023-09-01'),

(3,9, null,104, '2023-09-20'),

(4,10,4,102, '2023-10-18'),

(5,1,6,101, '2023-11-28'),

(6,7,2,105, '2023-12-31'),

(7,2,7,104, '2024-01-01'),

(8,1,9,102, '2024-01-26'),

(9,5,1,103, '2024-02-01'),

(10,6,10,105, '2024-03-08');

5. Query:

1. Group by:

select count(Prod_id), PCategory from Product group by PCategory;

Prod_id	PCategory
1	Bed
1	Chair
1	Dining Furniture
2	Wardrobes
2	Sofas
1	Bookcases units
1	Tables
1	Shelving units

2.Find the records of Replace Product whose price is greater than 2000:

select * from Replace_Product where RePrice>2000;

Replace_id	Category	RePrice
1	Bed	5000
3	Sofas	4000
7	Dining furniture	3000
8	Wardrobes	5000

3.Select Customer who have not place the order:

select cust_id, cust_name as Customer_Name from Customer where cust_id NOT IN (select cust_id from Order1);

Cust_id	Customer_Name
3	Pravara Kamble
8	Shivansh Thakur

6. Join:

1) Display list of Customers who have purchased Product with Debit Card:

select Order_id, cust_name as Customer_Name, Pay_Category from Customer natural join Order1 natural join Payment where Pay_Category='Debit Card';

Order_id	Customer_Name	Pay_Category
4	Gauri Palkar	Debit Card
8	Vihan Kurte	Debit Card

2) Display list of Customers who have not replace the Furniture:

select ScId, cust_name as Customer_Name from Customer natural join Shopcart where Replace_id is null;

ScId	Customer_Name
2	Isha Joshi
5	Harsh Joshi
8	Harsh Joshi
10	Sarvesh Hatiskar