Online Shopping application

Database Design and Applications (CSIZG518/SSZG518) – Assignment 1

Debojyoti Roy

2019ht13192@wilp.bits-pilani.ac.in

Table of Contents

[Description of the application 2](#_Toc36407272)

[Introduction 2](#_Toc36407273)

[Who will be the users? 2](#_Toc36407274)

[What are the benefits of this application? 2](#_Toc36407275)

[List of functions & features of the application 2](#_Toc36407276)

[How many users will use it simultaneously? 2](#_Toc36407277)

[ER diagram 2](#_Toc36407278)

[Description of each entities 3](#_Toc36407279)

[Customer 3](#_Toc36407280)

[Products 3](#_Toc36407281)

[Reviews 3](#_Toc36407282)

[Orders 3](#_Toc36407283)

[Payments 3](#_Toc36407284)

[Categories 3](#_Toc36407285)

[Suppliers 3](#_Toc36407286)

[Relational Model 4](#_Toc36407287)

[Normalization to 3NF 4](#_Toc36407288)

[SQL queries to fulfil the end users’ need 6](#_Toc36407289)

[Design indices 6](#_Toc36407290)

# Description of the application

## Introduction

## Who will be the users?

## What are the benefits of this application?

## List of functions & features of the application

This application will provide following features and functions –

1. Add or remove individual account for each user
2. Search product of same categories
3. Search list of products ordered in a specific date for a specific user
4. Search payment status of an order
5. Check whether payment for an order is successful or not
6. Fetch reviews of a product
7. Select a category of product from a specific Supplier
8. Look for orders between two dates
9. Find out the currier company, who will deliver the order
10. Find out the delivery date of a specific order
11. Search for the best product for a particular category

## How many users will use it simultaneously?

# ER diagram

ER diagram of Online shopping application is given below –

A picture containing text, map

Description automatically generated

## Description of each entities

### Customer

This table will have information regarding each customer.

### Products

This table will contain information regarding each product. Online portal gives facility to purchases only these products.

### Reviews

Customers can provide review for any product.

### Orders

This table tracks orders, placed by customers.

### Payments

Each order will be associated with one transaction and payment information.

### Categories

In Online Shopping sells product of multiple categories. This table contains the information of each categories.

### Suppliers

This table keeps information of all suppliers, who like to sell their products.

# Relational Model

Relational model of each entity is given below.

Customers

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User Id | Password | Email | Fname | Lname | Address1 | Address2 | Address3 | City | State | Country |

Primary key: User Id

Suppliers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Supplier Id | Company Name | Email | Company Website | Company phone No. |

Primary key: Supplier Id

Products

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Product Id | Supplier Id | Category Id | Name | Price | Number of purchases | Specification |

Primary key: Product Id

Foreign Keys: Supplier Id, Category Id

Reviews

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer Id | Product Id | Date | Title | Rating | Advantages | Disadvantages | Other Comments |

Foreign keys: Customer Id, Product Id

Orders

|  |  |  |
| --- | --- | --- |
| Order Id | User Id | Total Amount |

Primary key: Order Id

Foreign Key: User Id

Order List (This is represented using “Contain” relation)

|  |  |  |
| --- | --- | --- |
| Order Id | Product Id | Quantity |

Foreign keys: Order Id, Product Id

Payments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Payment Id | Order Id | Time | Date | Payment Type | Status |

Primary key: Payment Id

Foreign Keys: Order Id

Curriers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Currier Id | Company Name | City | Email | Company Website | Company Phone No. |

Primary key: Product Id

Category

|  |  |
| --- | --- |
| Category Id | Category Name |

Primary key: Product Id

Delivery

|  |  |  |  |
| --- | --- | --- | --- |
| Order Id | Currier Id | Delivery Time | Delivery Date |

Primary key: Product Id

Foreign Keys: Order Id, Currier Id

# Normalization to 3NF

In this online shopping system following functional dependencies exists –

1. Customers
   1. {User\_Id} 🡪 {Password, Email, Fname, Lname, Address1, Address2, Address3, City, State, Country}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist transitive dependencies because State and Country depends on City only. Hence, it is not in 3NF. Therefore, if we introduce separate table as below, it will satisfy 3NF as well –

* 1. {City} 🡪 {State, Country}

1. Suppliers
   1. {Supplier Id} 🡪 {Company Name, Email, Company Website, Company phone No}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Products
   1. {Product Id, Supplier Id, Category Id} 🡪 {Name, Price, Number of purchases, Specification}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Reviews
   1. {Customer Id, Product Id} 🡪 {Date, Title, Rating, Advantages, Disadvantages, Other Comments}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Orders
   1. {Order Id, User Id} 🡪 {Total Amount}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Order List
   1. {Order Id, Product Id} 🡪 {Quantity}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Payments
   1. {Payment Id, Order Id} 🡪 Time, Date, Payment Type, Status}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Curriers
   1. {Currier Id} 🡪 {Company Name, City, Email, Company Website, Company Phone No.}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Category
   1. {Category Id} 🡪 {Category Name}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

1. Delivery
   1. {Order Id, Currier Id} 🡪 {Delivery Time, Delivery Date}

This table consists of atomic attributes. So, it is in 1NF.

There exist no partial dependencies. Hence, it is in 2NF.

There exist no transitive dependencies, Hence, it is in 3NF.

All the tables are satisfying 3NF properties. So, whole system is in 3NF.

# SQL queries to fulfil the end users’ need

## Add or remove individual account for each user

### Add user:

Command - insert into Customers (User\_Id, User\_Password, Email, Fname, Lname, Address1, Address2, Address3, City, State, Country) values ('user\_id\_N', 'password\_abcd', 'emailid\_xyz@website.com', 'Firstnameabcd', 'Lastnameabcd', 'AddressX1a1', 'AddressX2a2', 'AddressX3a3', 'CityC', 'StateS', 'CountryCy');

### Delete user:

Command - delete from Customers where user\_id='user\_id\_N';

## Search product of same categories

Input: Name of Category

Command - select \* from Products where Category\_Id = 'category\_N';

## Search list of products ordered in a specific date for a specific user

Input: User id

Command - select o.User\_Id, o.Order\_Id, ol.Product\_Id from Orders o, Order\_List ol where o.Order\_Id = ol.Order\_Id and o.User\_Id = 'user\_id\_N';

## Search payment status of an order

Input: Order Id

Command - select o.User\_Id, o.Order\_Id, py.status from Orders o, Payments py where o.Order\_Id = py.Order\_Id and o.Order\_Id = 'Order\_Id\_N';

## Check whether payment for an order is successful or not

Input: Order Id

Command - select status from (select User\_Id, Order\_Id, status, row\_number, max(row\_number) over (partition by Order\_Id) as max\_row from (select o.User\_Id, o.Order\_Id, py.status, ROW\_NUMBER() over (order by (o.Order\_Id)) as ROW\_NUMBER from Orders o, Payments py where o.Order\_Id = py.Order\_Id and o.Order\_Id = 'Order\_Id\_N')) where row\_number = max\_row;

## Fetch reviews of a product

Inputs: Product Name

Command - select Rw.Rating, Rw.Advantages, Rw.Disadvantages from Reviews Rw, Products Pr where Rw.product\_id = Pr.product\_id and Pr.Product\_Name like 'Product\_Name';

## Select a category of product from a specific Supplier

Inputs: Name of the category, supplier’s company name.

Command - select pr.Product\_Id, pr.Supplier\_Id, pr.Category\_Id, pr.Product\_Name, pr.Price, pr.Number\_of\_purchases, pr.Product\_Specification from Products pr, Categories cr, Supplier sp where cr.category\_name = 'category\_name' and sp.Company\_Name = 'Company\_Name\_N' and pr.Category\_Id = cr.Category\_Id and pr.supplier\_id = sp.supplier\_id;

# Design indices