

# **REPORT ON ONLINE RETAIL APPLICATION**

Creating a database for an online retail application involves designing a schema that efficiently handles various aspects of retail operations, such as product management, customer information, orders, and inventory. Below, I'll outline a basic database schema and key considerations for an online retail application.

## **1. Database Schema**

### **Tables and Their Descriptions**

#### **1. Customers**

- CustomerID (Primary Key, Auto Increment)
- FirstName
- LastName
- Email (Unique)
- PasswordHash
- PhoneNumber
- Address
- City
- State
- ZipCode
- Country
- DateCreated

#### **2. Products**

- ProductID (Primary Key, Auto Increment)
- ProductName
- Description
- Price

- CategoryID (Foreign Key)
- StockQuantity
- DateAdded

### 3. Categories

- CategoryID (Primary Key, Auto Increment)
- CategoryName
- Description

### 4. Orders

- OrderID (Primary Key, Auto Increment)
- CustomerID (Foreign Key)
- OrderDate
- TotalAmount
- ShippingAddress
- OrderStatus (e.g., Pending, Shipped, Delivered, Canceled)

### 5. OrderItems

- OrderItemID (Primary Key, Auto Increment)
- OrderID (Foreign Key)
- ProductID (Foreign Key)
- Quantity
- UnitPrice

### 6. Reviews

- ReviewID (Primary Key, Auto Increment)
- ProductID (Foreign Key)
- CustomerID (Foreign Key)
- Rating (1 to 5)
- Comment
- ReviewDate

### 7. PaymentDetails

- PaymentID (Primary Key, Auto Increment)

- OrderID (Foreign Key)
- PaymentMethod (e.g., Credit Card, PayPal)
- PaymentStatus (e.g., Completed, Pending, Failed)
- PaymentDate

## 8. Inventory

- InventoryID (Primary Key, Auto Increment)
- ProductID (Foreign Key)
- StockLevel
- LastUpdated

## 9. Discounts

- DiscountID (Primary Key, Auto Increment)
- DiscountCode (Unique)
- Description
- DiscountAmount
- DiscountType (e.g., Percentage, Fixed Amount)
- StartDate
- EndDate

## Relationships

- Customers to Orders: One-to-Many
- Orders to OrderItems: One-to-Many
- Products to OrderItems: One-to-Many
- Categories to Products: One-to-Many
- Customers to Reviews: One-to-Many
- Products to Reviews: One-to-Many
- Orders to PaymentDetails: One-to-One
- Products to Inventory: One-to-One

## 2. Key Considerations

1. **Normalization:** Ensure the database design is normalized to reduce redundancy and improve data integrity. Typically, a schema is normalized to the third normal form (3NF).
2. **Indexes:** Create indexes on columns that are frequently used in queries, such as `Email` in the `Customers` table, `ProductName` in the `Products` table, and `OrderDate` in the `Orders` table.
3. **Security:** Store sensitive information, such as passwords, securely using hashing algorithms. Ensure secure access to the database.
4. **Scalability:** Design the database with scalability in mind, considering potential growth in data volume and user load. Techniques such as sharding, partitioning, or using a cloud-based database service can help.
5. **Backups:** Implement a robust backup and recovery plan to protect data against loss or corruption.
6. **Performance Optimization:** Regularly monitor database performance and optimize queries and schema design to ensure fast response times.
7. **Compliance:** Ensure the database design complies with relevant regulations and standards, such as GDPR for customer data privacy.

This schema provides a solid foundation for an online retail application, but it may need to be adjusted based on specific business requirements or additional features.

## **CODING FOR ONLINE RETAIL APPLICATION**

create database ManageShopOnline

create table product(

ProId varchar(10) primary key,

ProName varchar(50) not null,

ProDescription varchar(500),

ProPrice money check(ProPrice>0),

Stock int check (Stock>0 or Stock=0)

)

create table Categories(

CateID varchar(10) primary key,

CateName varchar(50) not null,

ProID varchar(10)foreign key references

Product(ProID),

CateDescription varchar(500) not null

)

```
create table Customer(  
  CusID varchar(10) primary key,  
  CusName varchar(50) not null,  
  CusEmail varchar(50) unique,  
  CusAddress varchar(100)not null  
)
```

```
create table Invoice(  
  InvoiceID varchar(10) primary key,  
  CusID varchar(10) foreign key references  
  Customer(CusID),  
  [date] datetime  
)
```

```
create table Invoice_Product(  
  InvoiceID varchar(10) foreign key references  
  Invoice(InvoiceID),  
  ProID varchar(10) foreign key references  
  Product(ProID),  
  Quantity int check(Quantity>0)  
)
```

```
create table Feedback(  

```

```
CusID varchar(10) foreign key references
Customer(CusID),
ProID varchar(10) foreign key references
Product(ProID),
comment varchar(300)unique,
[date] datetime
)
```

```
create table ManageAccount(
AccName varchar(30) not null,
AccPass varchar(20) not null,
AccPhone varchar(11)not null,
AccAddress varchar(50) not null,
AccEmail varchar(50) unique,
[status] varchar(20) check ([status]='activate' or
[status]= 'deactivate')
)
```

```
--Insert data into Product
select * from Product
```

insert into Product values ('P001','Book','New',500,100)

insert into Product values ('P111','Laptop  
Dell','New',1500,50)

insert into Product values ('P801','Pen','New',3,1000)

insert into Product values ('P246','SmartPhone  
Iphone','New',1000,25)

insert into Product values ('P456','Fan','New',35,100)

insert into Product values  
('P852','Notebook','New',10,500)

--Insert data into Categories

insert into Categories  
values('Cat03','Laptop','P111','sales up30%')

insert into Categories  
values('Cat08','Fan','P456','sales','upto 20%')

insert into Categories  
values('Cat13','Laptop','P111','sales up30%')

insert into Categories  
values('Cat30','Laptop','P111','sales up30%')

insert into Categories  
values('Cat32','Laptop','P111','sales up30%')



--Insert data into Customer

insert into Customer

values('Cus111','Naman','2416546465','naman@gmail.com','Bihar')

insert into Customer

values('Cus123','Nipun','2416421465','nipun@gmail.com','Uttarakhand')

insert into Customer

values('Cus114','Nitin','1116546465','nitin@gmail.com','Haridwar')

insert into Customer

values('Cus162','Visha','9416546465','vishal@gmail.com','Bhagalpur')

insert into Customer

values('Cus154','Saran','2452546465','saran@gmail.com','Bihar')

insert into Customer

values('Cus168','kishl','4826546465','kishl@gmail.com','Muzaffarpur')

```
insert into Customer  
values('Cus111','Naman','2416546465','naman@gmail.  
com','Bihar')
```

```
insert into Customer  
values('Cus111','Naman','2416546465','naman@gmail.  
com','Bihar')
```

--Insert data into Invoice

```
insert into Invoice values('I162','Cus111','4/12/2022')
```

```
insert into Invoice values('I168','Cus168','4/10/2022')
```

```
insert into Invoice values('I154','Cus154','4/15/2022')
```

```
insert into Invoice values('I123','Cus123','4/09/2022')
```

```
insert into Invoice values('I114','Cus114','4/16/2022')
```

--Insert data into Invoice\_Product

```
insert into Invoice_Product('I162','P111',2)
```

```
insert into Invoice_Product('I168','P246',1)
```

```
insert into Invoice_Product('I154','P852',5)
```

```
insert into Invoice_Product('I123','P977',5)
```

```
insert into Invoice_Product('I114','P456',15)
```

--Insert data into Feedback

```
insert into Feedback values('Cus162','P852','it is  
good','4/1/2022')
```

```
insert into Feedback values('Cus168','P111','it is  
good','4/5/2022')
```

```
insert into Feedback values('Cus154','P246','it is  
good','4/6/2022')
```

```
insert into Feedback values('Cus123','P977','it is  
good','4/2/2022')
```

```
insert into Feedback values('Cus114','P456','it is  
good','4/4/2022')
```

--Insert data into ManageAccount

```
insert into ManageAccount  
values('naman','CH0503','0985463254','naman@gmail.  
com','activate')
```

```
insert into ManageAccount  
values('nipun','123456','4528763254','nipun@gmail.co  
m','deactivate')
```

```
insert into ManageAccount  
values('nitin','452583','3256463254','nitin@gmail.com',  
'activate')
```