Online Shopping for Gadgets

Domain Description

In today's tech-driven world, gadgets have become an integral part of daily life. Gadgets could encompass various items like smartphones, laptops, tablets, smartwatches, etc. The system will include functionalities for users to browse, search, purchase, and manage orders of these gadgets.

Tables Before Normalization

User Table

UserID	Username	Email	Password	Address	Phone	OrderID	ProductID
1	JohnDoe	john@example.com	hashed123	123 Main St	555-1234	101	01
2	JaneSmith	jane@example.com	qwerty456	456 Elm St	555-5678	102	02

Products Table

Product ID	ProductName	Description	Price	Quantity Available	UserID	CategoryID	CategoryName
01	Smartphone Model X	Description of Model X	599.99	100	1	001	Smartphone
02	Laptop Model Pro	Description of Laptop	899.99	50	1	002	Laptop
03	Tablet Model Mini	Description of Tablet	349.99	75	2	003	Tablet

Orders Table

OrderID	UserID	OrderDate	TotalAmount	ShippingAddress	Status
101	1	2023-12-01 08:00	1499.97	123 Main St	Shipped
102	2	2023-12-02 10:30	1049.98	456 Elm St	Delivered

OrderDetails

OrderDetailID	OrderID	ProductID	Quantity	TotalAmount

201	101	01	2	1199.98
202	101	02	1	899.99
203	102	03	3	1049.97

Normalization

First Normal Form (1NF): Ensure atomic values in each cell and unique column names.

Second Normal Form (2NF): Remove partial dependencies.

Third Normal Form (3NF): Eliminate transitive dependencies.

There are redundant fields such as ProductName, Description, Price, Quantity present in multiple tables, leading to data duplication.

There's a lack of proper separation of concerns, causing dependencies that could lead to data inconsistencies.

Composite fields like OrderID, ProductID, and UserID appearing in multiple tables without normalized relations.

Tables After Normalization

User Table

UserID	**Username**	**Email**	**Password**	**Address**	**Phone**
1	JohnDoe	john@example.com	hashed123	123 Main St	555-1234
2	JaneSmith	jane@example.com	qwerty456	456 Elm St	555-5678

Products Table

ProductID	**ProductName**	**Description**	**Price**	**QuantityAvailable**	**CategoryID**
01	Smartphone Model X	Description of Model X	599.99	100	001
02	Laptop Model Pro	Description of Model Pro	899.9	50	002
03	Tablet Model Mini	Description of Model Mini	349.9	75	003

Category Table

CategoryID	**CategoryName**	**Description**
001	Smartphone	Smartphone description

002	Laptop	Laptop description
003	Tablet	Tablet description

Orders Table

OrderID	**UserID**	**OrderDate**	**TotalAmount**	**ShippingAddress**	**Status**
101	1	2023-12-01 08:00	2099.97	123 Main St	Shipped
102	2	2023-12-02 10:30	1049.98	456 Elm St	Delivered

OrderDetails Table

OrderDetailID	**OrderID**	**ProductID**	**Quantity**	**Subtotal**
201	101	01	2	1199.98
202	101	02	1	899.99
203	102	03	3	1049.97

The tables have atomic values. So, 1 NF

There are no partial dependencies in the tables. So, 2 NF

Third normal form is maintained as there are no transitive dependencies. So, 3 NF

Table Description

Users

Attributes	**Datatype**	**Constraints**
UserID	Integer	Primary Key
Username	String	Not Null
Email	String	Not Null
Password	String	Not Null
Address	String	Not Null
Phone	String	Not Null

Products

Attributes	**Datatype**	**Constraints**

ProductID	Integer	Primary Key
ProductName	String	Not Null
Description	Text	Not Null
Price	Decimal	Not Null
QuantityAvailable	Integer	Not Null
CategoryID	Integer	Foreign key

Categories

Attributes	**Datatype**	**Constraints**
CategoryID	Integer	Primary Key
CategoryName	String	Not Null
Description	Text	Not Null

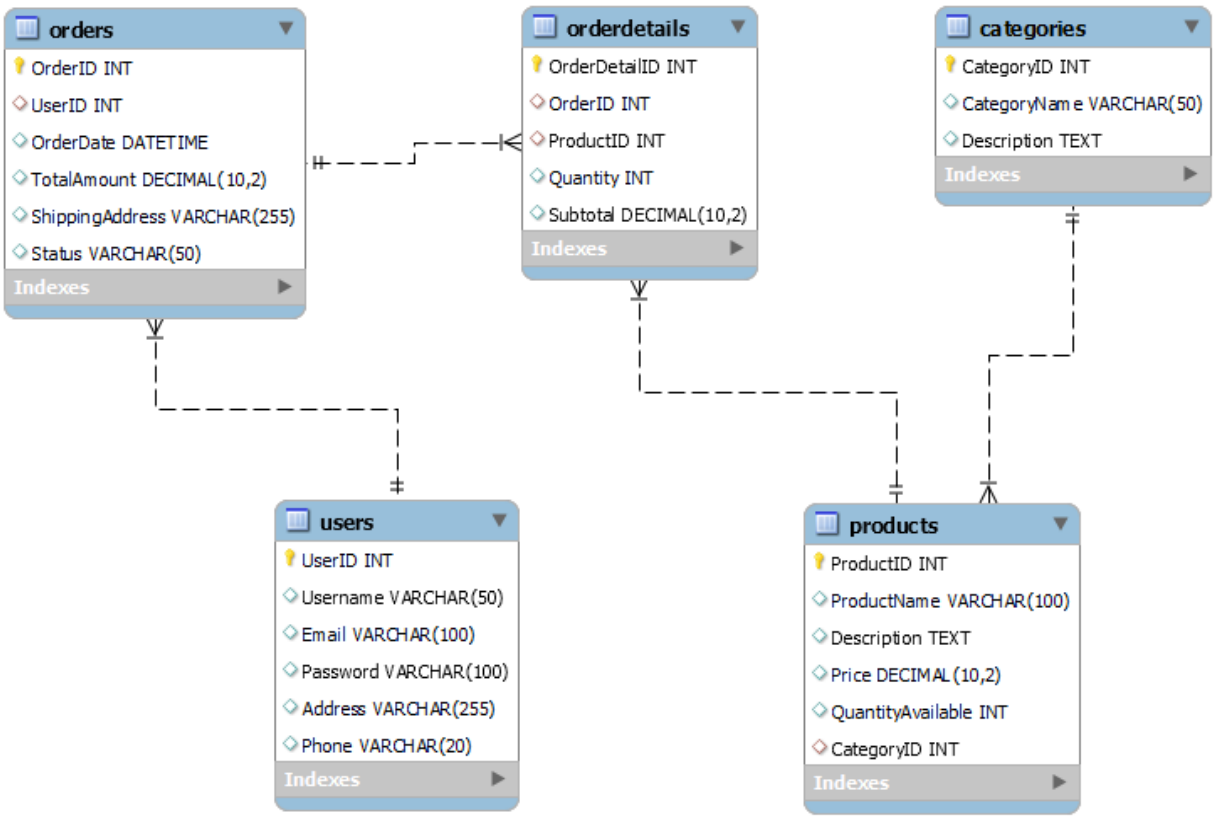
Orders

Attributes	**Datatype**	**Constraints**
OrderID	Integer	Primary Key
UserID	Integer	Foreign Key
OrderDate	Date	Not Null
TotalAmount	Decimal	Not Null
ShippingAddress	String	Not Null
Status	String	Not Null

OrderDetails

Attributes	**Datatype**	**Constraints**
OrderDetailID	Integer	Primary Key
OrderID	Integer	Foreign Key
ProductID	Integer	Foreign Key
Quantity	Integer	Not Null
Subtotal	Decimal	Not Null

Entity-Relationship Diagram (ERD)



Relationships:

- Users have orders (One-to-Many Relationship: One user can have multiple orders).
- Orders contain order details (One-to-Many Relationship: One order can have multiple order details).
- Products fall under categories (One-to-Many Relationship: One category can have multiple products).