**Oracle Database Schema for E-commerce Application**

**Users**

——————

NUMBER user\_id (Primary Key)

VARCHAR2(50) first\_name

VARCHAR2(50) last\_name

VARCHAR2(100) email

VARCHAR2(255) password

VARCHAR2(20) phone

NUMBER role\_id(Foreign Key to Role.role\_id) default is CUSTOMER

TIMESTAMP create\_at

Relationships:

1 User Can Place Many Orders (1:N)

Indexes:

Create index on user\_id

Create index on phone

Create index on email

**Role**

————

NUMBER role\_id(Primary Key)

VARCHAR2(50) role

role can be 'CUSTOMER', 'ADMIN'

**Categories**

—————————

NUMBER category\_id(Primary Key)

VARCHAR2(100) cagtegory\_name

Relationships:

1 Category Can Have Many Products (1:N)

**Products**

———————————

NUMBER product\_id(Primary Key)

VARCHAR2(100) name

CLOB description

NUMBER price

NUMBER stock\_quantity

NUMBER category\_id(Foreign Key to Categories.category\_id)

TIMESTAMP created\_at

TIMESTAMP updated\_at

Relationships:

• 1 Category Can Have Many Products (1:N)

• 1 Product Can Have Many Images (1:N)

Indexes:

Create index on product\_id

Create index on category\_id

**Product\_Images**

————————————————

NUMBER product\_id(Primary Key)

NUMBER product\_id(Foreign Key to Products.product\_id)

VARCHAR2(255) image\_url

Relationships:

• 1 Product Can Have Many Images (1:N)

**Orders**

——————

NUMBER order\_id(Primary Key)

NUMBER user\_id(Foreign Key to Users.user\_id)

NUMBER total\_price

NUMBER status\_id(Foreign Key to Order\_Status.status\_id) default is PENDING

TIMESTAMP created\_at

TIMESTAMP updated\_at

Relationships:

• 1 User Can Place Many Orders (1:N)

• 1 Order Can Have Many Products (via Order\_Items) (1:N)

Indexes:

Create index on order\_id

Create index on product\_id

**Order\_Status**

———————————

NUMBER status\_id(Primary Key)

VARCHAR2(20) status

Status can be 'PENDING', 'SHIPPED', 'DELIVERED', 'CANCELLED’.

**Order\_Items**

————————

NUMBER order\_item\_id(Primary Key)

NUMBER order\_id(Foreign Key to Orders. order\_id)

NUMBER product\_id(Foreign Key to Products.product\_id)

NUMBER quantity

NUMBER unit\_price

NUMBER total\_price

Relationships:

• 1 Order Can Have Many Products (1:N)

**Payments**

———————————-

NUMBER payment\_id(Primary Key)

NUMBER order\_id(Foreign Key to Orders. order\_id)

NUMBER user\_id(Foreign Key to Users.user\_id)

NUMBER amount

NUMBER payment\_method\_id(Foreign Key to Payment\_Method\_Type.payment\_method\_id)

NUMBER payment\_status\_id(Foreign Key to Payment\_Status\_Type.payment\_status\_id) DEFAULt PENDING

VARCHAR2(100) transaction\_id

TIMESTAMP created\_at

TIMESTAMP updated\_at

Relationships:

• 1 Order Has 1 Payment (1:1)

Indexes:

Create index on order\_id

Create index on user\_id

**Payment\_Method\_Type**

———————————————————

NUMBER payment\_method\_id(Primary Key)

VARCHAR2(50) payment\_method

Status can be 'CREDIT\_CARD', 'DEBIT\_CARD', 'PAYPAL', 'COD'

**Payment\_Status\_Type**

———————————————————

NUMBER payment\_status\_id(Primary Key)

VARCHAR2(20) payment\_status

Status can be 'PENDING', 'COMPLETED', 'FAILED', 'REFUNDED'

**Addresses**

—————————

NUMBER address\_id(Primary Key)

NUMBER order\_id(Foreign Key to Orders. order\_id)

NUMBER user\_id(Foreign Key to Users.user\_id)

VARCHAR2(255) address\_line1

VARCHAR2(255) address\_line2

VARCHAR2(100) city

VARCHAR2(100) state

VARCHAR2(20) zip\_code

VARCHAR2(100) country

TIMESTAMP created\_at

TIMESTAMP updated\_at

Relationships:

• 1 User Can Have Many Addresses (1:N)

**Reviews**

——————————

NUMBER review\_id(Primary Key)

NUMBER user\_id(Foreign Key to Users.user\_id)

NUMBER product\_id(Foreign Key to Products.product\_id)

NUMBER rating

CLOB comment

TIMESTAMP created\_at

Relationships:

• 1 User Can Write Many Reviews (1:N)

• 1 Product Can Have Many Reviews (1:N)

You can add additional tables like Coupons & Discounts, Whishlists when those features are need to be implemented.