A Project on

Optical Shop's Database System

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELORS OF TECHNOLOGY

In Computers Science and Engineering

2022-2026

By,

NAVYAA KABRA

229301073

MANIPAL UNIVERSITY
JAIPUR

UNDER THE GUIDANCE OF

Dr. Neha Janu

Department of Computer Science and Engineering

School of Computer Science and Engineering

Manipal University Jaipur

Jaipur, Rajasthan

Table of Contents

Introduction

Welcome to the database system for Kabra Optical House, a small-scale optical shop dedicated to providing high-quality eyewear products and exceptional customer service. This database serves as the backbone of Kabra Optical House's operations, facilitating the management of various aspects of the business, from inventory control to customer relations.

The primary objective of this database project is to streamline and optimize the management of Kabra Optical House's operations. By centralizing essential data and providing efficient means for data retrieval and manipulation, this system aims to enhance productivity, improve decision-making processes, and ultimately contribute to the overall success of the business.

With its robust database structure and comprehensive data management capabilities, the Kabra Optical House database system is poised to revolutionize the way the optical shop conducts its business operations. By harnessing the power of data-driven insights and automation, Kabra Optical House aims to deliver unparalleled value to its customers while achieving sustainable growth and success in the competitive eyewear market.

Formal Specification

This document explains about the entities we are using in the project, reason for choosing the entities, their attributes, how they will work, data type, primary key, foreign key, constraints (if exists) and flow of data.

Basically, this is designed to make it easier and more understandable for everyone.

Entities:

1. contactlenses
2. customers
3. employees
4. frames
5. inventory
6. orderdetails
7. orders
8. roles
9. sunglasses
10. suppliers
11. transactions
12. users

Contact Lenses:

Reason: Contact lenses are a key product offering in an optical shop like Kabra Optical House.

Attributes:

- LensID (Primary Key, int)
- Brand (varchar(100))
- LensType (varchar(50))
- Color (varchar(50))
- Power (decimal(5,2))

- Price (decimal(10,2))

Constraints: LensID as the primary key ensures uniqueness for each contact lens. Other attributes can have constraints like NOT NULL if necessary.

Data Flow: Data flows into this entity when new contact lenses are added to the inventory. It's accessed when customers inquire about or purchase contact lenses.

Customers:

Reason: Customers are the primary stakeholders in the business.

Attributes:

- CustomerID (Primary Key, int)
- FirstName (varchar(50))
- LastName (varchar(50))
- Email (varchar(100))
- Phone (varchar(20))

Constraints: CustomerID as the primary key ensures uniqueness. Email and Phone may have constraints like UNIQUE if each customer should have a unique email or phone number.

Data Flow: Data flows into this entity when new customers register or make purchases. It's accessed when managing customer accounts or processing orders.

Employees:

Reason: Employees are essential for managing day-to-day operations.

Attributes:

- EmployeeID (Primary Key, int)
- FirstName (varchar(50))
- LastName (varchar(50))
- Position (varchar(100))
- Email (varchar(100))
- Phone (varchar(20))

Constraints: EmployeeID as the primary key ensures uniqueness. Email and Phone may have constraints like UNIQUE if each employee should have a unique email or phone number.

Data Flow: Data flows into this entity when new employees are hired. It's accessed when managing employee information, scheduling, and payroll.

Frames:

Reason: Frames are a key product offering in an optical shop.

Attributes:

- FrameID (Primary Key, int)
- FrameType (varchar(100))
- FrameColor (varchar(50))
- Size (varchar(20))
- Material (varchar(100))
- Price (decimal(10,2))

Constraints: FrameID as the primary key ensures uniqueness. Other attributes can have constraints like NOT NULL if necessary.

Data Flow: Data flows into this entity when new frames are added to the inventory. It's accessed when customers inquire about or purchase frames.

Inventory:

Reason: Tracks available products and quantities in stock.

Attributes:

- ProductID (Primary Key, int)
- ProductName (varchar(100))
- ProductType (varchar(50))
- Price (decimal(10,2))
- Quantity (int)

Constraints: ProductID as the primary key ensures uniqueness. Other attributes may have constraints like NOT NULL or CHECK constraints.

Data Flow: Data flows into this entity when new products are added to the inventory or when quantities are updated. It's accessed when managing inventory levels and fulfilling orders.

Order Details:

Reason: Stores details of items included in each order.

Attributes:

- OrderDetailID (Primary Key, int)
- OrderID (Foreign Key, int)
- ProductID (Foreign Key, int)
- Quantity (int)
- UnitPrice (decimal(10,2))

Constraints: OrderDetailID as the primary key ensures uniqueness. Foreign key constraints ensure data integrity between this table and the orders and inventory tables.

Data Flow: Data flows into this entity when customers place orders. It's accessed when retrieving details of specific orders.

Orders:

Reason: Stores general information about orders placed by customers.

Attributes:

- OrderID (Primary Key, int)
- CustomerID (Foreign Key, int)
- OrderDate (date)
- TotalAmount (decimal(10,2))

Constraints: OrderID as the primary key ensures uniqueness. Foreign key constraints ensure data integrity with the customers table.

Data Flow: Data flows into this entity when customers place orders. It's accessed when managing order information and generating reports.

Roles:

Reason: Defines different roles within the organization.

Attributes:

- RoleID (Primary Key, int)
- RoleName (varchar(50))

Constraints: RoleID as the primary key ensures uniqueness. RoleName may have a UNIQUE constraint if each role name should be unique.

Data Flow: Data flows into this entity when defining roles for employees. It's accessed when assigning permissions and responsibilities to users.

Sunglasses:

Reason: Sunglasses are another key product offering in an optical shop.

Attributes:

- SunglassesID (Primary Key, int)
- Style (varchar(100))
- LensColor (varchar(50))
- FrameMaterial (varchar(100))
- Price (decimal(10,2))

Constraints: SunglassesID as the primary key ensures uniqueness. Other attributes can have constraints like NOT NULL if necessary.

Data Flow: Data flows into this entity when new sunglasses are added to the inventory. It's accessed when customers inquire about or purchase sunglasses.

Suppliers:

Reason: Stores information about the suppliers providing products to Kabra Optical House.

Attributes:

- SupplierID (Primary Key, int)
- SupplierName (varchar(100))
- ContactInfo (varchar(255))

Constraints: SupplierID as the primary key ensures uniqueness. Other attributes can have constraints like NOT NULL if necessary.

Data Flow: Data flows into this entity when new suppliers are added. It's accessed when managing supplier relationships and procurement.

Transactions:

Reason: Records all transactions processed by Kabra Optical House.

Attributes:

- TransactionID (Primary Key, int)
- OrderID (Foreign Key, int)
- EmployeeID (Foreign Key, int)
- TransactionDate (datetime)

Constraints: TransactionID as the primary key ensures uniqueness. Foreign key constraints ensure data integrity with the orders and employees tables.

Data Flow: Data flows into this entity when transactions occur, such as sales or returns. It's accessed when analyzing sales data and tracking transaction history.

Users:

Reason: Manages user accounts and access privileges for the database system.

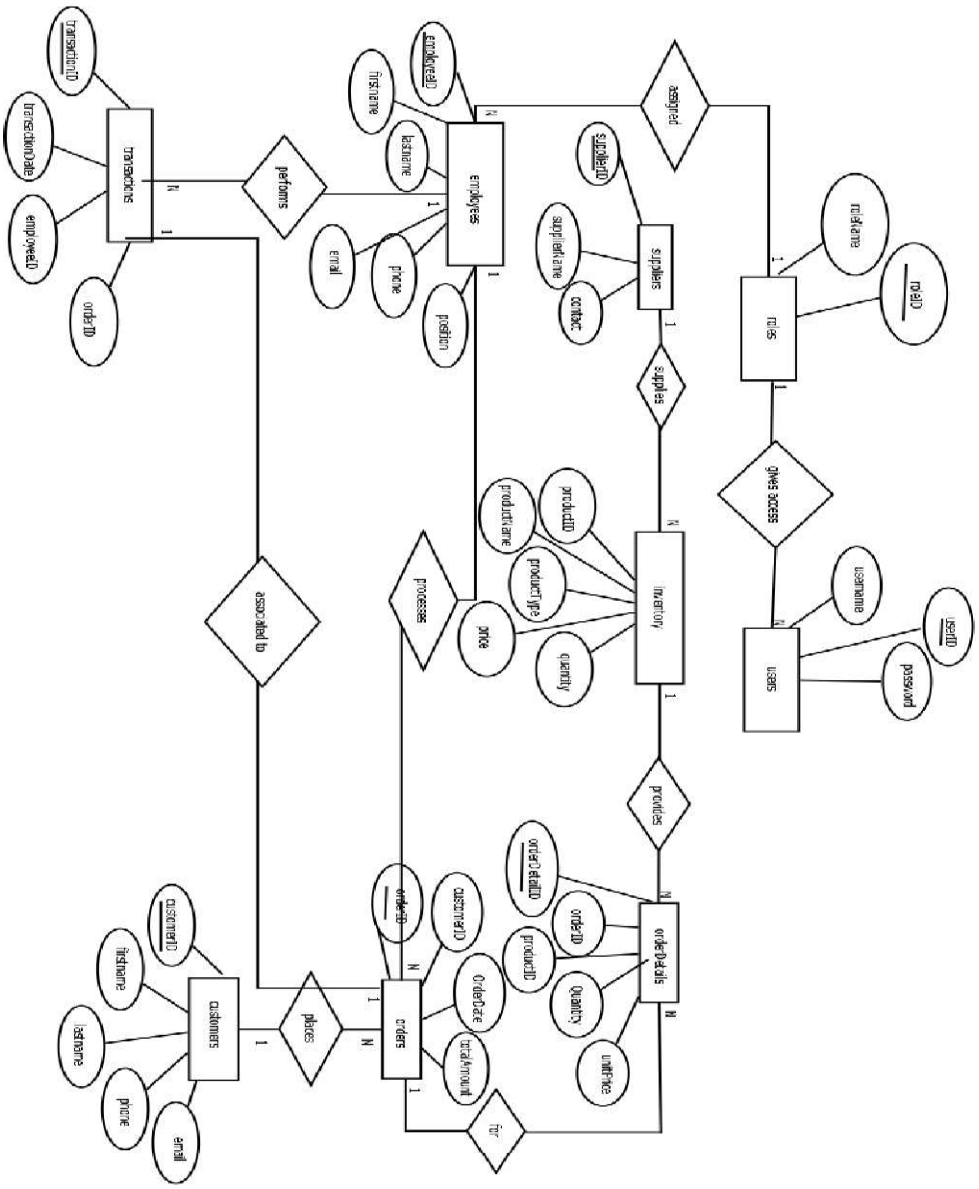
Attributes:

- UserID (Primary Key, int)
- Username (varchar(50))
- Password (varchar(255))
- RoleID (Foreign Key, int)

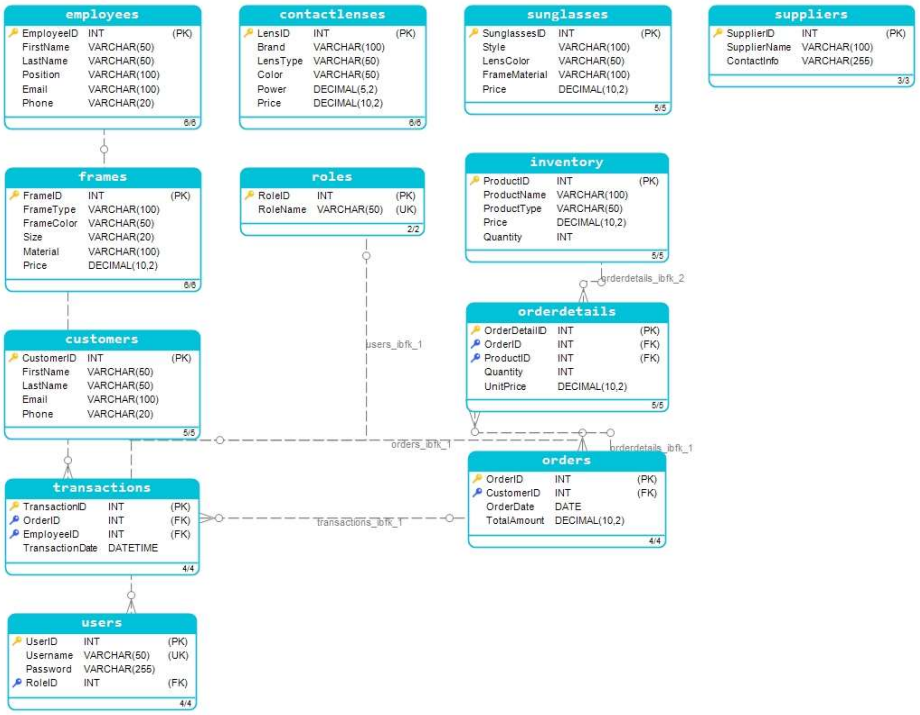
Constraints: UserID as the primary key ensures uniqueness. Username may have a UNIQUE constraint to ensure each username is unique. RoleID as a foreign key links to the roles table, ensuring referential integrity.

Data Flow: Data flows into this entity when new users are registered, or user permissions are updates. It's accessed when managing user accounts, authentication, and authorization.

ER Diagram


transactions
transactionID
transactionDate
employeeID
orderID
performs
employees
firstname
lastname
email
phone
position
assigned
suppliers
supplierID
supplierName
contact
supplies
inventory
productID
productName
productType
price
quantity
processes
assembled in
rules
roleID
rolename
allows access
users
userID
username
password
provides
orderDetailID
orderID
productID
Quantity
unitPrice
order details
customers
customerID
firstname
lastname
phone
email
places
orders
orderID
customerID
OrderDate
totalAmount
for

ER to table


employees
EmployeeID INT (PK)
FirstName VARCHAR(50)
LastName VARCHAR(50)
Position VARCHAR(100)
Email VARCHAR(100)
Phone VARCHAR(20)
6/6
contactlenses
LensID INT (PK)
Brand VARCHAR(100)
LensType VARCHAR(50)
Color VARCHAR(50)
Power DECIMAL(5,2)
Price DECIMAL(10,2)
6/6
sunglasses
SunglassesID INT (PK)
Style VARCHAR(100)
LensColor VARCHAR(50)
FrameMaterial VARCHAR(100)
Price DECIMAL(10,2)
5/5
suppliers
SupplierID INT (PK)
SupplierName VARCHAR(100)
ContactInfo VARCHAR(255)
3/3
frames
FrameID INT (PK)
FrameType VARCHAR(100)
FrameColor VARCHAR(50)
Size VARCHAR(20)
Material VARCHAR(100)
Price DECIMAL(10,2)
6/6
roles
RoleID INT (PK)
RoleName VARCHAR(50) (UK)
2/2
inventory
ProductID INT (PK)
ProductName VARCHAR(100)
ProductType VARCHAR(50)
Price DECIMAL(10,2)
Quantity INT
5/5
orderdetails_ibfk_2
orderdetails
OrderDetailID INT (PK)
OrderID INT (FK)
ProductID INT (FK)
Quantity INT
UnitPrice DECIMAL(10,2)
5/5
customers
CustomerID INT (PK)
FirstName VARCHAR(50)
LastName VARCHAR(50)
Email VARCHAR(100)
Phone VARCHAR(20)
5/5
users_ibfk_1
orders_ibfk_1
orderdetails_ibfk_1
orders
OrderID INT (PK)
CustomerID INT (FK)
OrderDate DATE
TotalAmount DECIMAL(10,2)
4/4
transactions
TransactionID INT (PK)
OrderID INT (FK)
EmployeeID INT (FK)
TransactionDate DATETIME
4/4
transactions_ibfk_1
users
UserID INT (PK)
Username VARCHAR(50) (UK)
Password VARCHAR(255)
RoleID INT (FK)
4/4

Creating Tables & Inserting Data

1. ContactLenses:

```
CREATE TABLE contactlenses (
   LensID INT NOT NULL PRIMARY KEY,
   Brand VARCHAR(100),
   LensType VARCHAR(50),
   Color VARCHAR(50),
   Power DECIMAL(5,2),
   Price DECIMAL(10,2)
);
```

```
mysql> select * from contactlenses;
+--------+--------------+----------+-------+-------+---------+
| LensID | Brand        | LensType | Color | Power | Price   |
+--------+--------------+----------+-------+-------+---------+
|      1 | Acuvue       | Daily    | Clear |  2.50 |  800.00 |
|      2 | Bausch & Lomb | Monthly  | Blue  |  3.00 | 1000.00 |
|      3 | Biofinity    | Monthly  | Gray  |  2.00 | 1500.00 |
+--------+--------------+----------+-------+-------+---------+
```

2. Customers:

```
CREATE TABLE customers (
   CustomerID INT NOT NULL PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100),
   Phone VARCHAR(20)
);
```

```
mysql> select * from customers;
+------------+-----------+----------+-----------------------+------------+
| CustomerID | FirstName | LastName | Email                 | Phone      |
+------------+-----------+----------+-----------------------+------------+
|          1 | Navyaa    | Kabra    | kabranavyaa@gmail.com | 9876543210 |
|          2 | Apoorvaa  | Kabra    | apoorvaakabra@gmail.com | 8765432109 |
|          3 | Mudit     | Gupta    | mudit28@gmail.com     | 7654321098 |
+------------+-----------+----------+-----------------------+------------+
```

3. Employees:

```
CREATE TABLE employees (
   EmployeeID INT NOT NULL PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Position VARCHAR(100),
   Email VARCHAR(100),
   Phone VARCHAR(20)
);
```

```
mysql> select * from employees;
+------------+-----------+----------+-----------------+-----------------+------------+
| EmployeeID | FirstName | LastName | Position        | Email           | Phone      |
+------------+-----------+----------+-----------------+-----------------+------------+
|          1 | Amit      | Patel    | Optician        | amit@koh.com    | 9876543210 |
|          2 | Pooja     | Singh    | Sales Associate | pooja@koh.com   | 8765432109 |
|          3 | Deepali   | Kabra    | Store Manager   | deepali@koh.com | 8567439277 |
+------------+-----------+----------+-----------------+-----------------+------------+
```

4. Frames:

```
CREATE TABLE frames (
   FrameID INT NOT NULL PRIMARY KEY,
   FrameType VARCHAR(100),
   FrameColor VARCHAR(50),
   Size VARCHAR(20),
   Material VARCHAR(100),
   Price DECIMAL(10,2)
);
```

```
mysql> select * from frames;
+---------+------------+------------+--------+----------+---------+
| FrameID | FrameType  | FrameColor | Size   | Material | Price   |
+---------+------------+------------+--------+----------+---------+
|       1 | Rectangular | Black      | Medium | Plastic  | 1200.00 |
|       2 | Round      | Brown      | Small  | Metal    | 1800.00 |
|       4 | Cat Eye    | Gold       | Large  | Metal    | 3500.00 |
+---------+------------+------------+--------+----------+---------+
```

5. Inventory:

```
CREATE TABLE inventory (
   ProductID INT NOT NULL PRIMARY KEY,
```

```
    ProductName VARCHAR(100),
    ProductType VARCHAR(50),
    Price DECIMAL(10,2),
    Quantity INT
);
```

```
mysql> select * from inventory;
+-----------+---------------+---------------+---------+----------+
| ProductID | ProductName   | ProductType   | Price   | Quantity |
+-----------+---------------+---------------+---------+----------+
|         1 | Eyeglasses    | Frames        | 1500.00 |       50 |
|         2 | Sunglasses    | Sunglasses    | 2000.00 |       30 |
|         3 | Contact Lenses | Contact Lenses |  800.00 |      100 |
+-----------+---------------+---------------+---------+----------+
```

6. OrderDetails:

```
CREATE TABLE orderdetails (
    OrderDetailID INT NOT NULL PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    UnitPrice DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES inventory(ProductID)
);
```

```
mysql> select * from orderdetails;
+--------------+---------+-----------+----------+-----------+
| OrderDetailID | OrderID | ProductID | Quantity | UnitPrice |
+--------------+---------+-----------+----------+-----------+
|            1 |       1 |         1 |        2 |   1500.00 |
|            2 |       2 |         2 |        1 |   2000.00 |
|            3 |       3 |         3 |        3 |    800.00 |
+--------------+---------+-----------+----------+-----------+
```

7. Orders:

```
CREATE TABLE orders (
    OrderID INT NOT NULL PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10,2),
```

```
    FOREIGN KEY (CustomerID) REFERENCES customers(CustomerID)
);
```

```
mysql> select * from orders;
+---------+------------+------------+-------------+
| OrderID | CustomerID | OrderDate  | TotalAmount |
+---------+------------+------------+-------------+
|       1 |          1 | 2024-04-01 |     1500.00 |
|       2 |          2 | 2024-04-02 |     2000.00 |
|       3 |          1 | 2024-04-03 |      800.00 |
+---------+------------+------------+-------------+
```

8. Roles:

```
CREATE TABLE roles (
    RoleID INT NOT NULL PRIMARY KEY,
    RoleName VARCHAR(50) UNIQUE
);
```

```
mysql> select * from roles;
+--------+----------+
| RoleID | RoleName |
+--------+----------+
|      2 | Employee |
|      1 | Manager  |
+--------+----------+
```

9. Sunglasses:

```
CREATE TABLE sunglasses (
    SunglassesID INT NOT NULL PRIMARY KEY,
    Style VARCHAR(100),
    LensColor VARCHAR(50),
    FrameMaterial VARCHAR(100),
    Price DECIMAL(10,2)
);
```

```
mysql> select * from sunglasses;
+--------------+---------+-----------+---------------+---------+
| SunglassesID | Style   | LensColor | FrameMaterial | Price   |
+--------------+---------+-----------+---------------+---------+
|            1 | Aviator | Green     | Metal         | 2500.00 |
|            2 | Round   | Black     | Plastic       | 2200.00 |
|            3 | Square  | Blue      | Metal         | 2800.00 |
+--------------+---------+-----------+---------------+---------+
```

10. Suppliers:

```
CREATE TABLE suppliers (
    SupplierID INT NOT NULL PRIMARY KEY,
    SupplierName VARCHAR(100),
    ContactInfo VARCHAR(255)
);
```

```
mysql> select * from suppliers;
+------------+---------------------------+-------------+
| SupplierID | SupplierName              | ContactInfo |
+------------+---------------------------+-------------+
|          1 | Indian Optical Industries | 1234567890  |
|          2 | Mumbai Frames Pvt. Ltd.   | 9876543210  |
|          3 | Delhi Eyewear Distributors | 9876543210  |
+------------+---------------------------+-------------+
```

11. Transactions:

```
CREATE TABLE transactions (
    TransactionID INT NOT NULL PRIMARY KEY,
    OrderID INT,
    EmployeeID INT,
    TransactionDate DATETIME,
    FOREIGN KEY (OrderID) REFERENCES orders(OrderID),
    FOREIGN KEY (EmployeeID) REFERENCES employees(EmployeeID)
);
```

```
mysql> select * from transactions;
+---------------+---------+------------+---------------------+
| TransactionID | OrderID | EmployeeID | TransactionDate     |
+---------------+---------+------------+---------------------+
|             1 |       1 |          1 | 2024-04-01 10:30:00 |
|             2 |       2 |          2 | 2024-04-02 11:45:00 |
|             3 |       3 |          1 | 2024-04-03 09:15:00 |
+---------------+---------+------------+---------------------+
```

12. Users:

```
CREATE TABLE users (
    UserID INT NOT NULL PRIMARY KEY,
    Username VARCHAR(50) UNIQUE,
    Password VARCHAR(255),
    RoleID INT,
    FOREIGN KEY (RoleID) REFERENCES roles(RoleID)
);
```

```
mysql> select * from users;
+--------+----------+-------------+--------+
| UserID | Username | Password    | RoleID |
+--------+----------+-------------+--------+
|      1 | jitendra | password123 |      1 |
|      2 | deepali  | password456 |      2 |
+--------+----------+-------------+--------+
```