Ayberk Savaş

DATABASE APPLICATION PROJECT FOR AN ONLINE WATCH STORE

Project Overview

I decided to create a database model specifically for an online watch market because I am personally interested in watches and curious about how online businesses handle sales, inventory, and customer relations. This project aims to cover the essential information needed to manage watch products, customers, and orders in a single, well-structured system.

Motivation

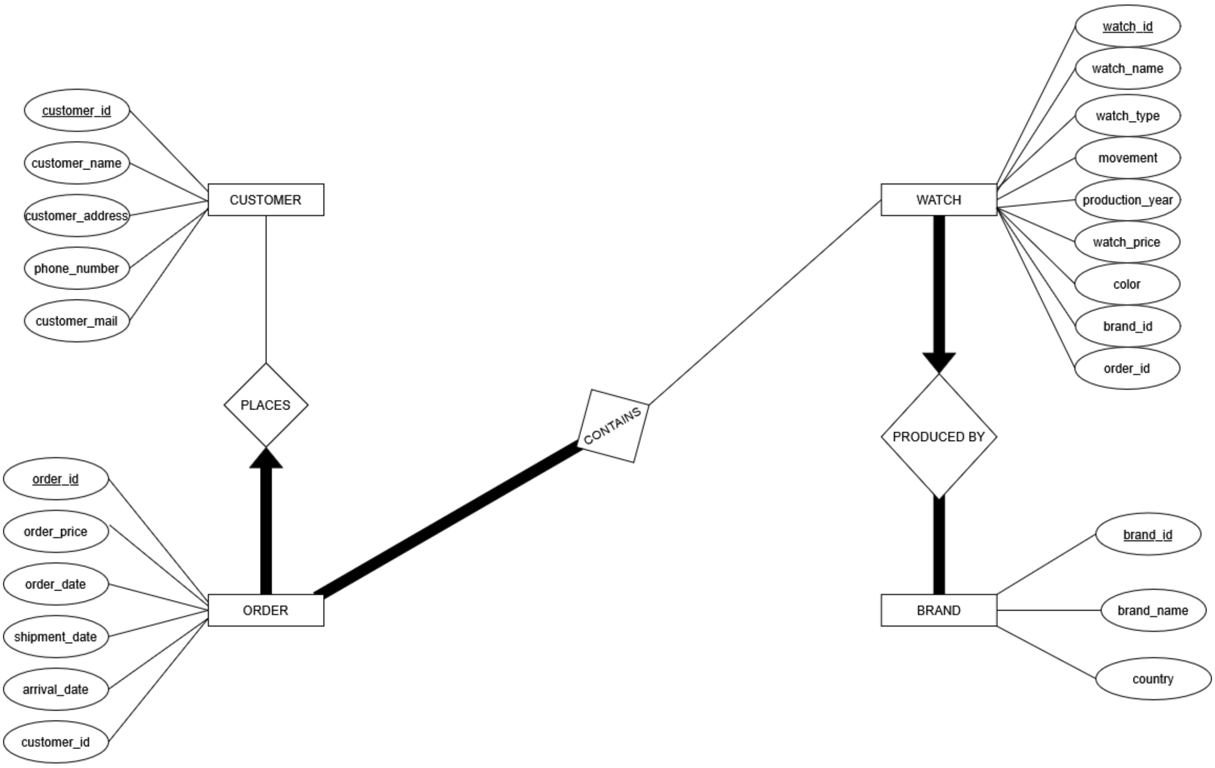
Personal Curiosity in Watches: My interest in different watch brands, types, and features inspired me to build a database that can handle details like watch movements, production years, and prices.

Learning About Online Business Operations: I wanted to gain a better understanding of how e-commerce stores track orders, customer details, and inventory. By designing the database schema, I can see how data flows in a real-world online store setting.

Overall, this project combines my personal enthusiasm for watches with an exploration of how online businesses manage orders, inventory, and customer data. It serves as a practical way to understand both database design principles and real-world e-commerce operations.

Tools and Platforms: I drew the diagram with Draw.io and I am using MySQL for the database applications.

ER DIAGRAM



ENTITIES AND RELATIONSHIPS

ER diagram includes four main entities:

Customer, Order, Watch, Brand

These entities are connected through three relationships:

- **PLACES:** A customer places an order.
- **CONTAINS:** An order contains one or more watches.
- **PRODUCED BY:** A brand produces a watch.

Constraints:

- Every order is placed by a single customer (key and participation constraint from ORDER to CUSTOMER).
- Every order contains at least one watch (participation constraint from ORDER to WATCH).
- Every watch must be produced by a brand (only one brand to be specific so key constraint and participation constraint from WATCH to BRAND).
- A brand must have at least one watch to be considered a brand.

Cardinalities in Relationships:

CUSTOMER-ORDER: There is one to many relationship.

ORDER-WATCH: There is one to many relationship.

WATCH-BRAND: There is many to one relationship.

Attributes of the Entities:

Customer

- **customer_id:** Unique identifier for each customer (Primary Key).
- **customer_name:** Name of the customer.
- **customer_address:** Address of the customer.
- **phone_number:** Contact phone number.
- **customer_mail:** Email address.

Order

- **order_id:** Unique identifier for each order (Primary Key).
- **order_price:** Total price of the order.
- **order_date:** Date when the order was placed.
- **shipment_date:** Date when the order was shipped.
- **arrival_date:** Expected arrival date.
- **customer_id:** Foreign key linking to the CUSTOMER table.

Watch

- **watch_id:** Unique identifier for each watch (Primary Key).
- **watch_name:** Name of the watch model.
- **watch_type:** Display type of the watch.
- **movement:** Type of movement (ex: Quartz, Automatic).
- **production_year:** Year the watch was produced.
- **watch_price:** Price of the watch.
- **color:** Color of the watch.
- **brand_id:** Foreign key linking to the BRAND table.
- **order_id:** Foreign key linking to the ORDER table (nullable to handle unsold watches).

Brand

- **brand_id:** Unique identifier for each brand (Primary Key).
- **brand_name:** Name of the brand (Unique and NOT NULL).
- **country:** Country of origin of the brand.

RELATIONAL MODEL

Database Design Highlights

- **CUSTOMER Table:** Stores customer information with unique customer IDs and essential contact details.
- **BRAND Table:** Stores brand-specific data such as brand name and country of origin.
- **WATCH Table:** Contains details of each watch, including type, movement, and brand association.
- **ORDER Table:** Tracks orders placed by customers and their lifecycle (order date, shipment date, and arrival date).

How It All Comes Together

Each **customer** can place multiple **orders**, linking to the specific **watches** they've purchased.

The **_BRAND_** table helps categorize watches and keep brand information consistent.

The **_ORDER_** table's reference to **_CUSTOMER_** provides a clear way to track who bought which watch and when it was shipped and delivered.

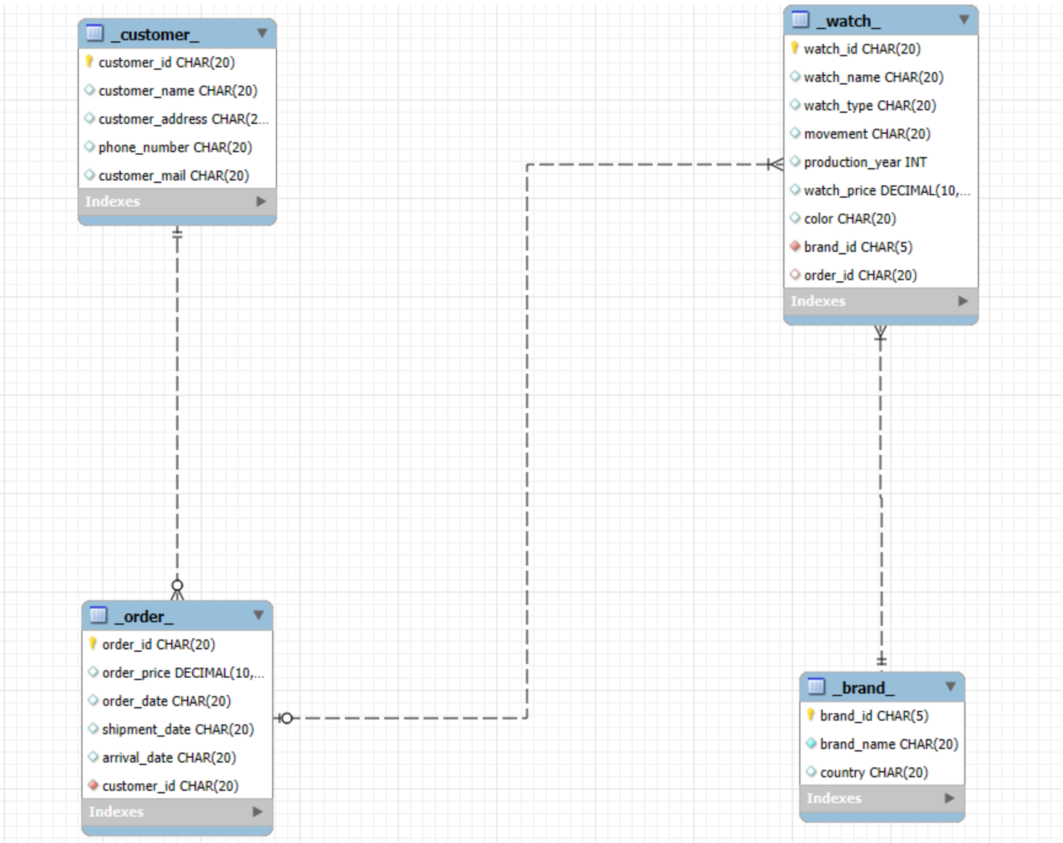
Benefits and Future Possibilities

Clear Data Relationships: By separating customers, brands, watches, and orders, the design is easier to maintain and expand in the future.

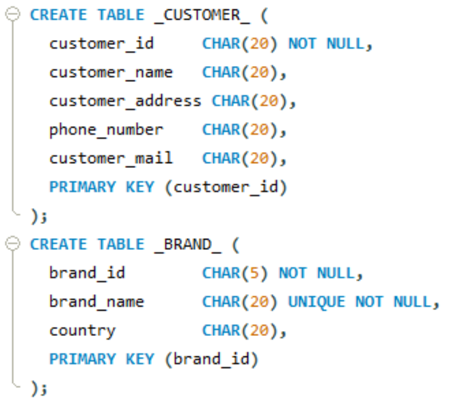
Scalability: More tables could be added (ex: payment, shipping) as the business grows or to track more aspects of the online store.

Business Insights: Storing data in well-defined tables opens opportunities for reporting on sales trends, customer preferences, and brand performance.

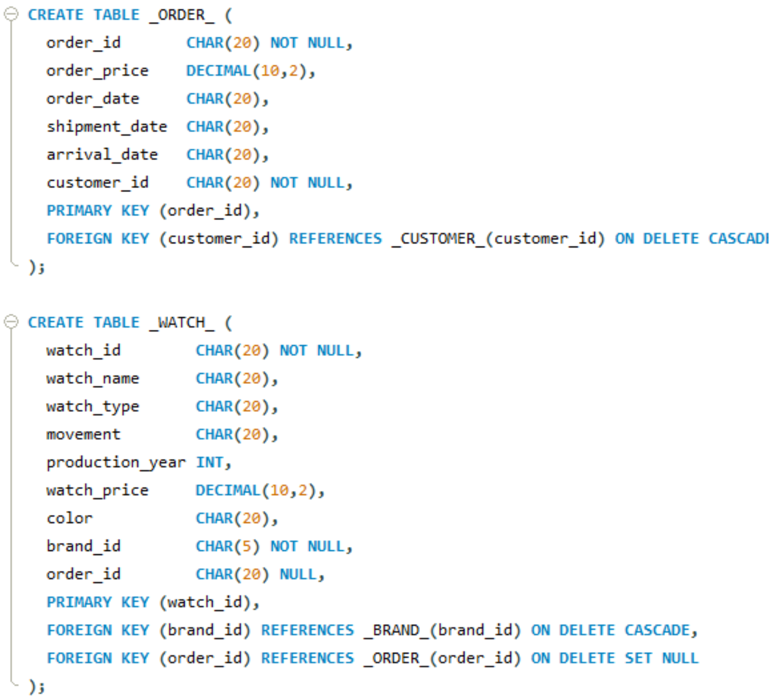
ER Diagram on MySQL:



CREATE TABLE Statements:

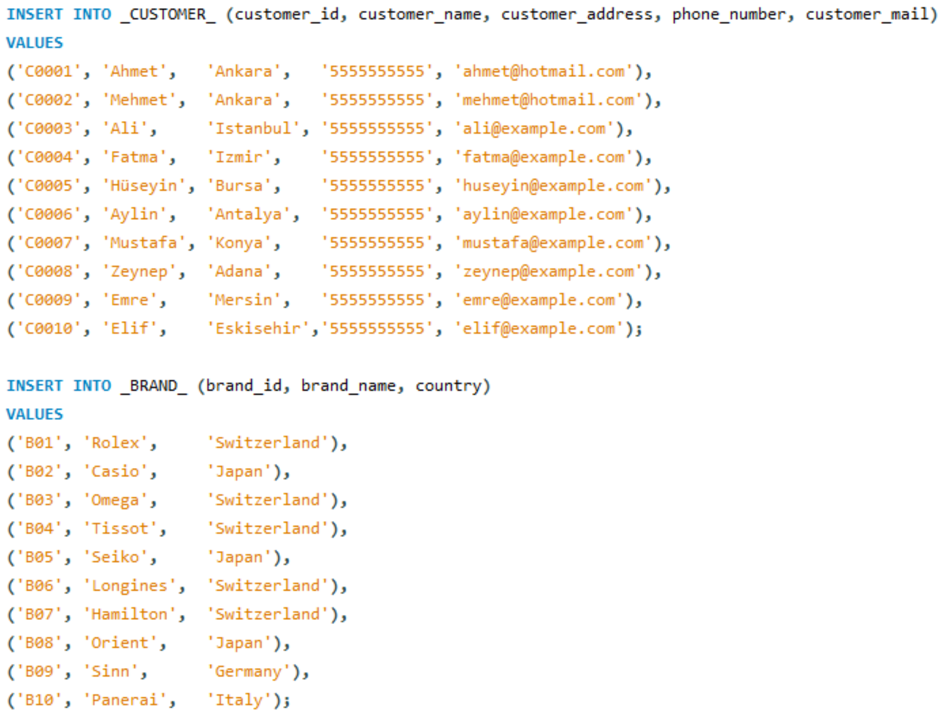


```
CREATE TABLE _CUSTOMER_ (
    customer_id      CHAR(20) NOT NULL,
    customer_name    CHAR(20),
    customer_address CHAR(20),
    phone_number     CHAR(20),
    customer_mail    CHAR(20),
    PRIMARY KEY (customer_id)
);
CREATE TABLE _BRAND_ (
    brand_id         CHAR(5) NOT NULL,
    brand_name       CHAR(20) UNIQUE NOT NULL,
    country          CHAR(20),
    PRIMARY KEY (brand_id)
);
```

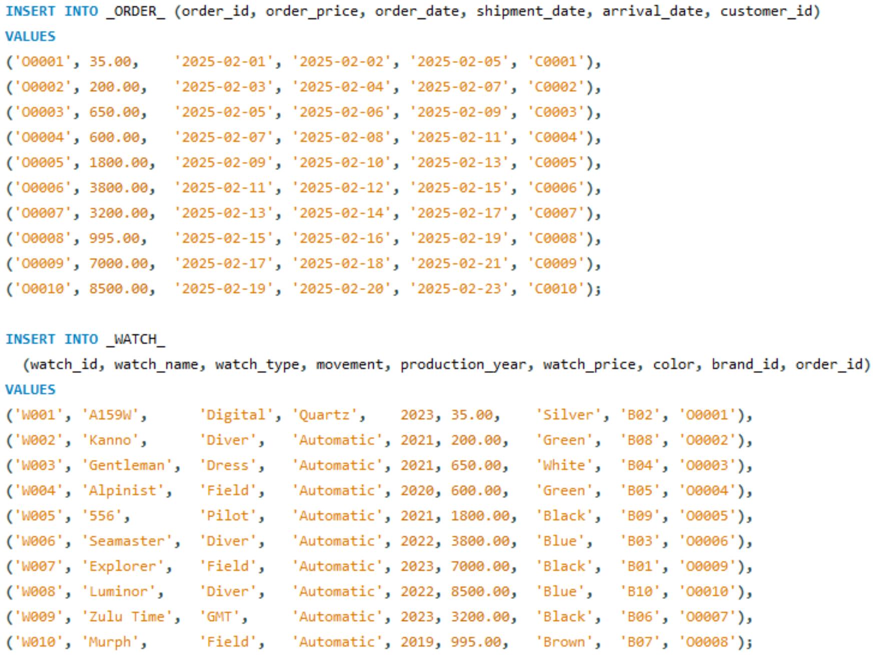


```
CREATE TABLE _ORDER_ (
    order_id      CHAR(20) NOT NULL,
    order_price   DECIMAL(10,2),
    order_date    CHAR(20),
    shipment_date CHAR(20),
    arrival_date  CHAR(20),
    customer_id   CHAR(20) NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (customer_id) REFERENCES _CUSTOMER_(customer_id) ON DELETE CASCADI
);

CREATE TABLE _WATCH_ (
    watch_id        CHAR(20) NOT NULL,
    watch_name      CHAR(20),
    watch_type      CHAR(20),
    movement        CHAR(20),
    production_year INT,
    watch_price     DECIMAL(10,2),
    color           CHAR(20),
    brand_id        CHAR(5) NOT NULL,
    order_id        CHAR(20) NULL,
    PRIMARY KEY (watch_id),
    FOREIGN KEY (brand_id) REFERENCES _BRAND_(brand_id) ON DELETE CASCADE,
    FOREIGN KEY (order_id) REFERENCES _ORDER_(order_id) ON DELETE SET NULL
);
```

Insert Into Statements:



```sql
INSERT INTO _CUSTOMER_ (customer_id, customer_name, customer_address, phone_number, customer_mail)
VALUES
('C0001', 'Ahmet',   'Ankara',   '5555555555', 'ahmet@hotmail.com'),
('C0002', 'Mehmet',  'Ankara',   '5555555555', 'mehmet@hotmail.com'),
('C0003', 'Ali',     'Istanbul', '5555555555', 'ali@example.com'),
('C0004', 'Fatma',   'Izmir',    '5555555555', 'fatma@example.com'),
('C0005', 'Hüseyin', 'Bursa',    '5555555555', 'huseyin@example.com'),
('C0006', 'Aylin',   'Antalya',  '5555555555', 'aylin@example.com'),
('C0007', 'Mustafa', 'Konya',    '5555555555', 'mustafa@example.com'),
('C0008', 'Zeynep',  'Adana',    '5555555555', 'zeynep@example.com'),
('C0009', 'Emre',    'Mersin',   '5555555555', 'emre@example.com'),
('C0010', 'Elif',    'Eskisehir','5555555555', 'elif@example.com');

INSERT INTO _BRAND_ (brand_id, brand_name, country)
VALUES
('B01', 'Rolex',     'Switzerland'),
('B02', 'Casio',     'Japan'),
('B03', 'Omega',     'Switzerland'),
('B04', 'Tissot',    'Switzerland'),
('B05', 'Seiko',     'Japan'),
('B06', 'Longines',  'Switzerland'),
('B07', 'Hamilton',  'Switzerland'),
('B08', 'Orient',    'Japan'),
('B09', 'Sinn',      'Germany'),
('B10', 'Panerai',   'Italy');
```



```sql
INSERT INTO _ORDER_ (order_id, order_price, order_date, shipment_date, arrival_date, customer_id)
VALUES
('O0001', 35.00,    '2025-02-01', '2025-02-02', '2025-02-05', 'C0001'),
('O0002', 200.00,   '2025-02-03', '2025-02-04', '2025-02-07', 'C0002'),
('O0003', 650.00,   '2025-02-05', '2025-02-06', '2025-02-09', 'C0003'),
('O0004', 600.00,   '2025-02-07', '2025-02-08', '2025-02-11', 'C0004'),
('O0005', 1800.00,  '2025-02-09', '2025-02-10', '2025-02-13', 'C0005'),
('O0006', 3800.00,  '2025-02-11', '2025-02-12', '2025-02-15', 'C0006'),
('O0007', 3200.00,  '2025-02-13', '2025-02-14', '2025-02-17', 'C0007'),
('O0008', 995.00,   '2025-02-15', '2025-02-16', '2025-02-19', 'C0008'),
('O0009', 7000.00,  '2025-02-17', '2025-02-18', '2025-02-21', 'C0009'),
('O0010', 8500.00,  '2025-02-19', '2025-02-20', '2025-02-23', 'C0010');

INSERT INTO _WATCH_
  (watch_id, watch_name, watch_type, movement, production_year, watch_price, color, brand_id, order_id)
VALUES
('W001', 'A159W',      'Digital', 'Quartz',    2023, 35.00,    'Silver', 'B02', 'O0001'),
('W002', 'Kanno',      'Diver',   'Automatic', 2021, 200.00,   'Green',  'B08', 'O0002'),
('W003', 'Gentleman',  'Dress',   'Automatic', 2021, 650.00,   'White',  'B04', 'O0003'),
('W004', 'Alpinist',   'Field',   'Automatic', 2020, 600.00,   'Green',  'B05', 'O0004'),
('W005', '556',        'Pilot',   'Automatic', 2021, 1800.00,  'Black',  'B09', 'O0005'),
('W006', 'Seamaster',  'Diver',   'Automatic', 2022, 3800.00,  'Blue',   'B03', 'O0006'),
('W007', 'Explorer',   'Field',   'Automatic', 2023, 7000.00,  'Black',  'B01', 'O0009'),
('W008', 'Luminor',    'Diver',   'Automatic', 2022, 8500.00,  'Blue',   'B10', 'O0010'),
('W009', 'Zulu Time',  'GMT',     'Automatic', 2023, 3200.00,  'Black',  'B06', 'O0007'),
('W010', 'Murph',      'Field',   'Automatic', 2019, 995.00,   'Brown',  'B07', 'O0008');
```