Lucy Njuguna

SQL Project

Overview

This project demonstrates the structured design and implementation of a normalized relational database that models a course management system. Developed for an academic environment, the system includes detailed entity-relationship modeling and SQL table creation to enforce referential integrity. While originally modeled for academic scheduling, the schema structure and relationship mapping simulate real-world systems used in regulated industries such as education, healthcare, and financial compliance.

This project is a strong demonstration of both **data normalization principles** and **secure database structuring** that underpins identity management, audit logging, and regulatory reporting.

Entity-Relationship Diagram (ERD)

Entities and Relationships

The ERD includes the following relationships:

A Course can be offered in many Sections

Each Section is taught by one Instructor

A Room can be used by multiple Sections

Each Section has one MeetingTime

A Student can enroll in multiple Sections, and each Section can have many Students

These relationships are enforced through foreign keys and normalized table structures, helping ensure data integrity and efficiency.

Relational Schema

Course(CourseID, CourseName)

Instructor(InstructorID, InstructorName)

Room(RoomID, Capacity)

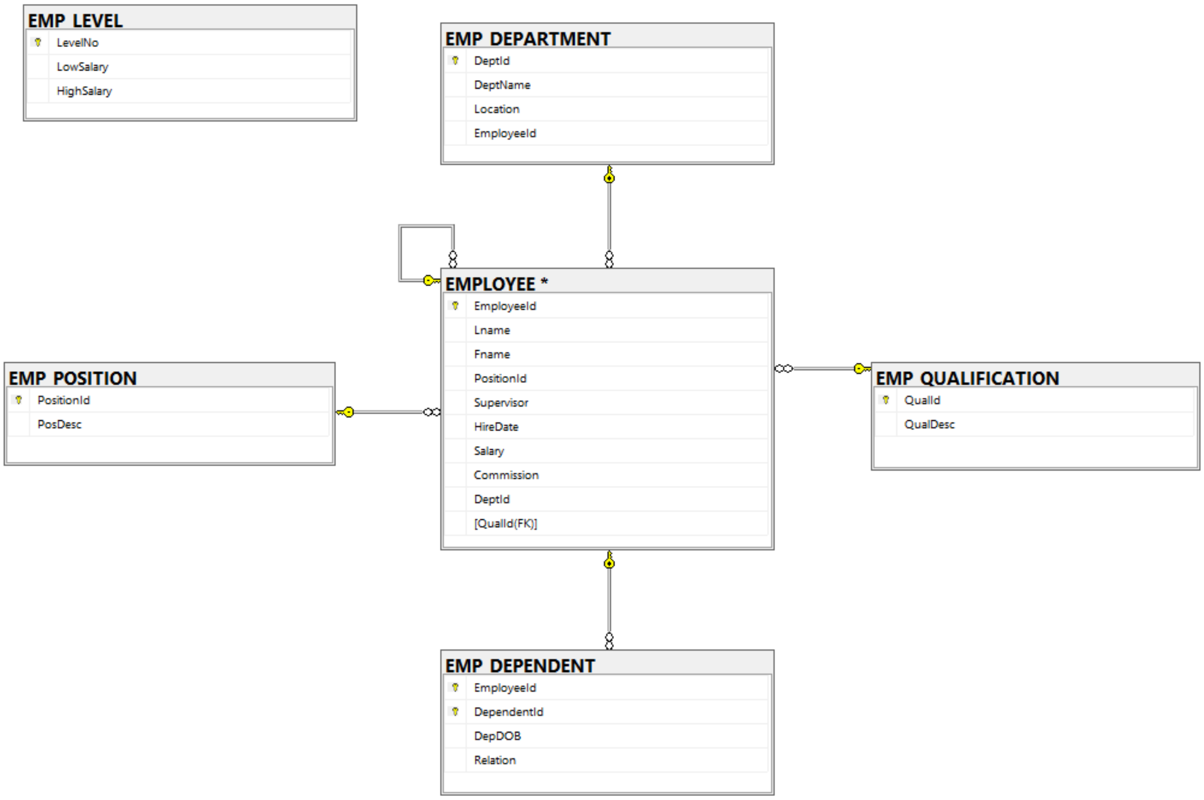
MeetingTime(MeetingTimeID, TimeSlot)

Section(SectionID, CourseID, InstructorID, RoomID, MeetingTimeID)

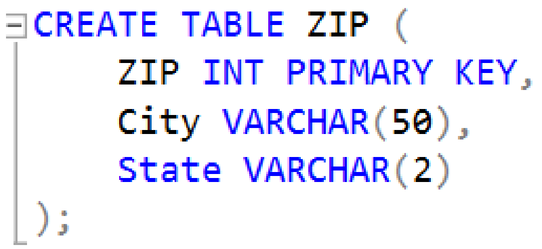
Student(StudentID, StudentName)

Enrollment(StudentID, SectionID)

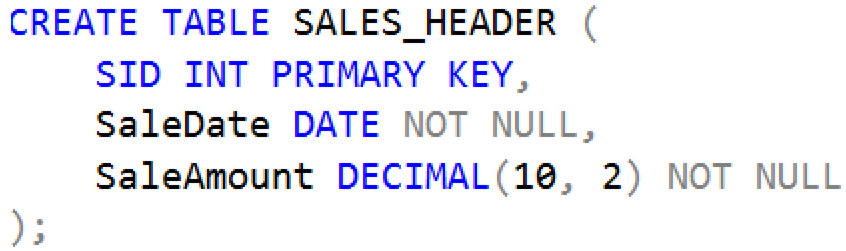
Entity-Relationship Diagram (ERD) based on the information.



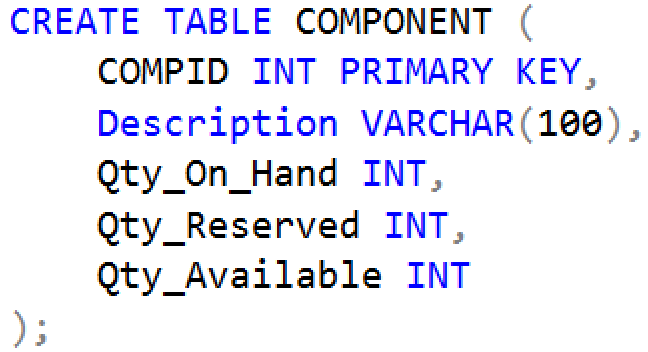
SQL scripts used to create all tables, relationships, primary keys, and foreign keys



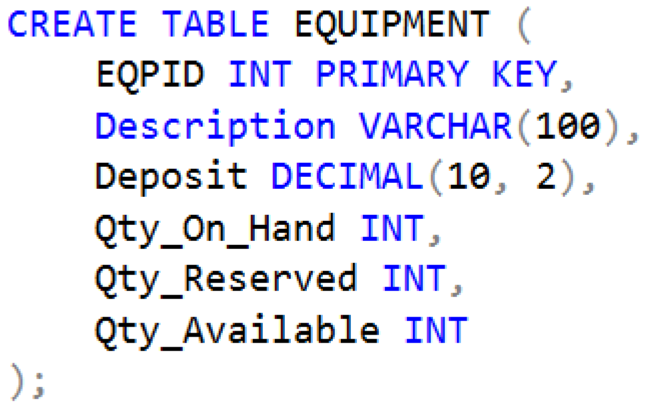
```
CREATE TABLE ZIP (
    ZIP INT PRIMARY KEY,
    City VARCHAR(50),
    State VARCHAR(2)
);
```



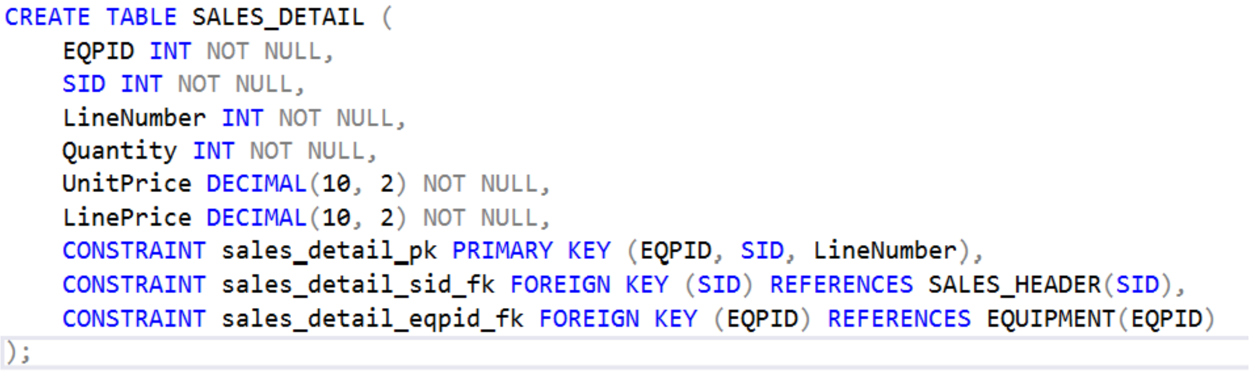
```
CREATE TABLE SALES_HEADER (
    SID INT PRIMARY KEY,
    SaleDate DATE NOT NULL,
    SaleAmount DECIMAL(10, 2) NOT NULL
);
```



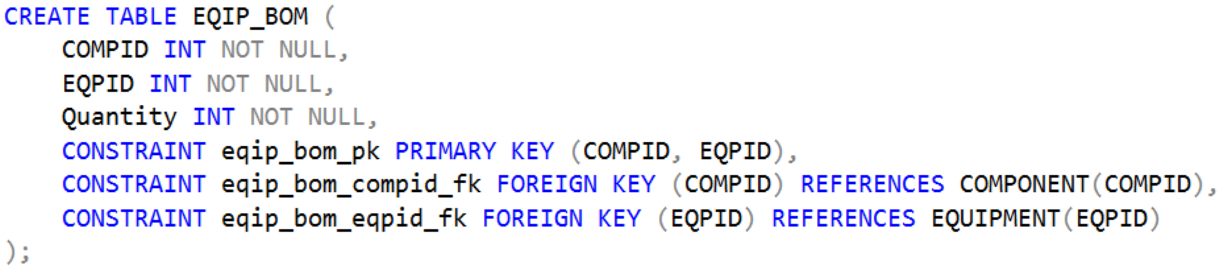
```
CREATE TABLE COMPONENT (
    COMPID INT PRIMARY KEY,
    Description VARCHAR(100),
    Qty_On_Hand INT,
    Qty_Reserved INT,
    Qty_Available INT
);
```



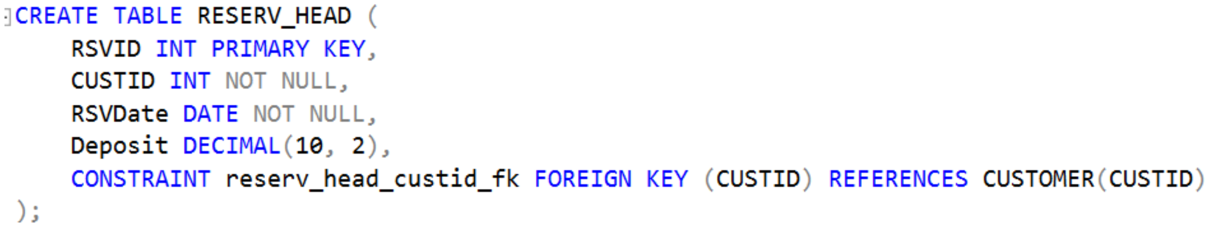
```
CREATE TABLE EQUIPMENT (
    EQPID INT PRIMARY KEY,
    Description VARCHAR(100),
    Deposit DECIMAL(10, 2),
    Qty_On_Hand INT,
    Qty_Reserved INT,
    Qty_Available INT
);
```



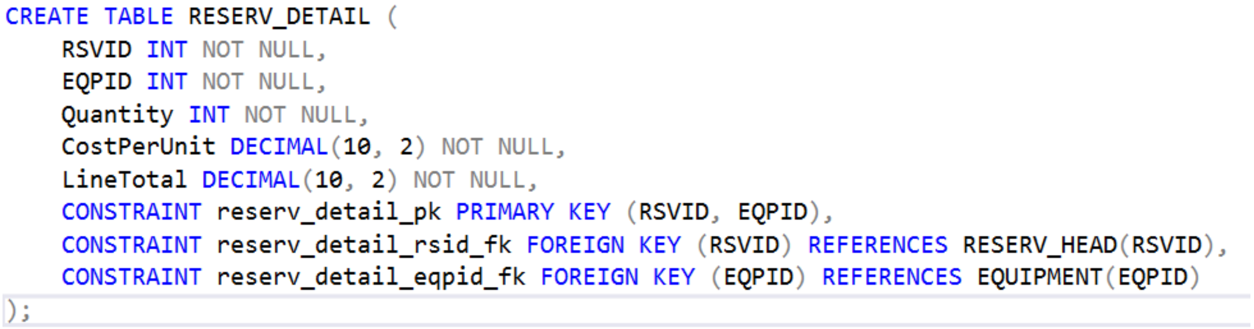
```sql
CREATE TABLE SALES_DETAIL (
    EQPID INT NOT NULL,
    SID INT NOT NULL,
    LineNumber INT NOT NULL,
    Quantity INT NOT NULL,
    UnitPrice DECIMAL(10, 2) NOT NULL,
    LinePrice DECIMAL(10, 2) NOT NULL,
    CONSTRAINT sales_detail_pk PRIMARY KEY (EQPID, SID, LineNumber),
    CONSTRAINT sales_detail_sid_fk FOREIGN KEY (SID) REFERENCES SALES_HEADER(SID),
    CONSTRAINT sales_detail_eqpid_fk FOREIGN KEY (EQPID) REFERENCES EQUIPMENT(EQPID)
);
```



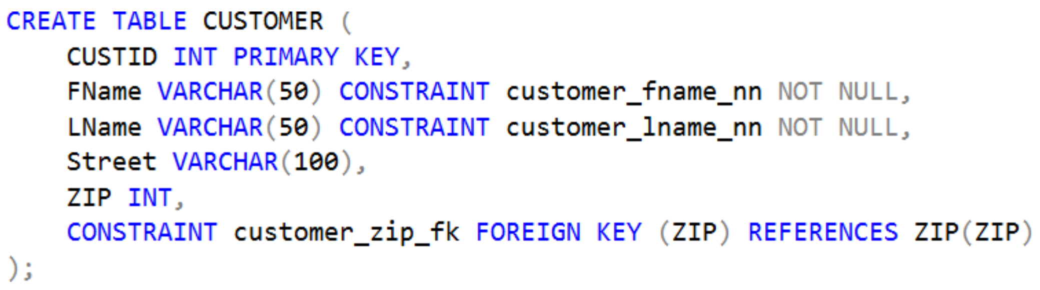
```sql
CREATE TABLE EQIP_BOM (
    COMPID INT NOT NULL,
    EQPID INT NOT NULL,
    Quantity INT NOT NULL,
    CONSTRAINT eqip_bom_pk PRIMARY KEY (COMPID, EQPID),
    CONSTRAINT eqip_bom_compid_fk FOREIGN KEY (COMPID) REFERENCES COMPONENT(COMPID),
    CONSTRAINT eqip_bom_eqpid_fk FOREIGN KEY (EQPID) REFERENCES EQUIPMENT(EQPID)
);
```



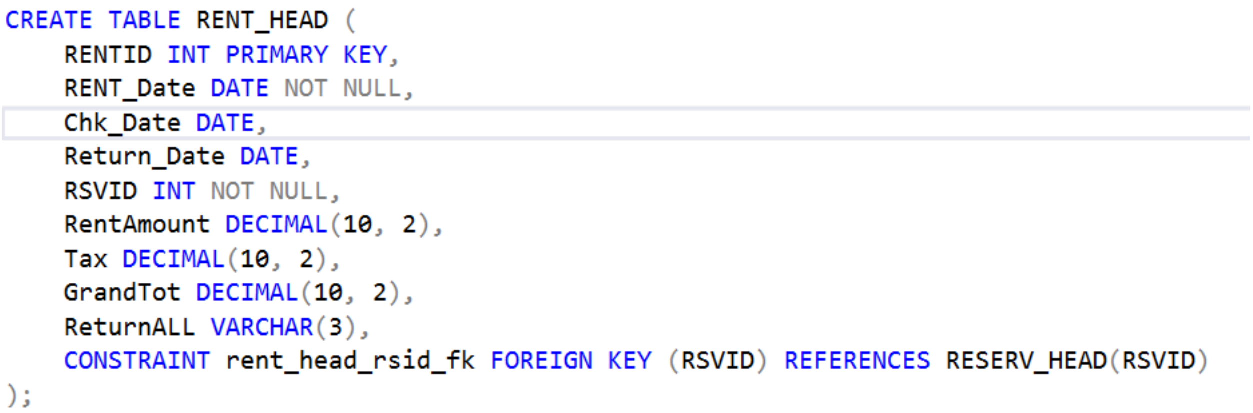
```sql
CREATE TABLE RESERV_HEAD (
    RSVID INT PRIMARY KEY,
    CUSTID INT NOT NULL,
    RSVDate DATE NOT NULL,
    Deposit DECIMAL(10, 2),
    CONSTRAINT reserv_head_custid_fk FOREIGN KEY (CUSTID) REFERENCES CUSTOMER(CUSTID)
);
```



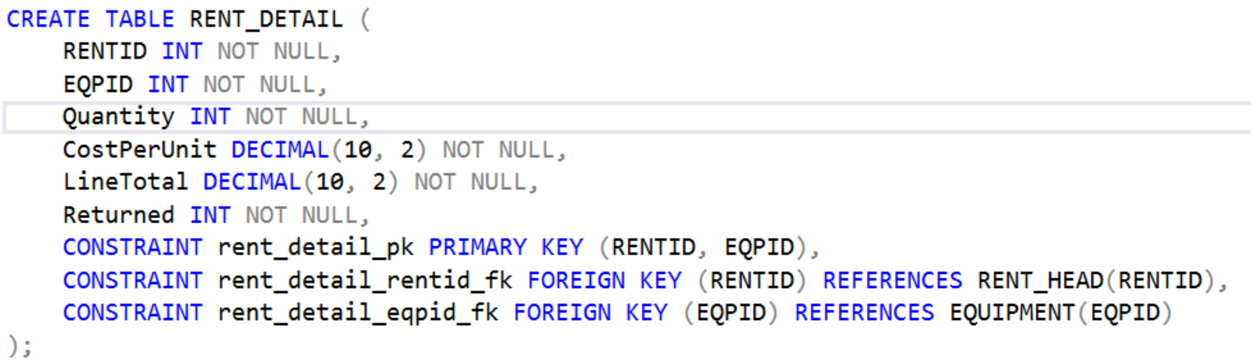
```sql
CREATE TABLE RESERV_DETAIL (
    RSVID INT NOT NULL,
    EQPID INT NOT NULL,
    Quantity INT NOT NULL,
    CostPerUnit DECIMAL(10, 2) NOT NULL,
    LineTotal DECIMAL(10, 2) NOT NULL,
    CONSTRAINT reserv_detail_pk PRIMARY KEY (RSVID, EQPID),
    CONSTRAINT reserv_detail_rsid_fk FOREIGN KEY (RSVID) REFERENCES RESERV_HEAD(RSVID),
    CONSTRAINT reserv_detail_eqpid_fk FOREIGN KEY (EQPID) REFERENCES EQUIPMENT(EQPID)
);
```



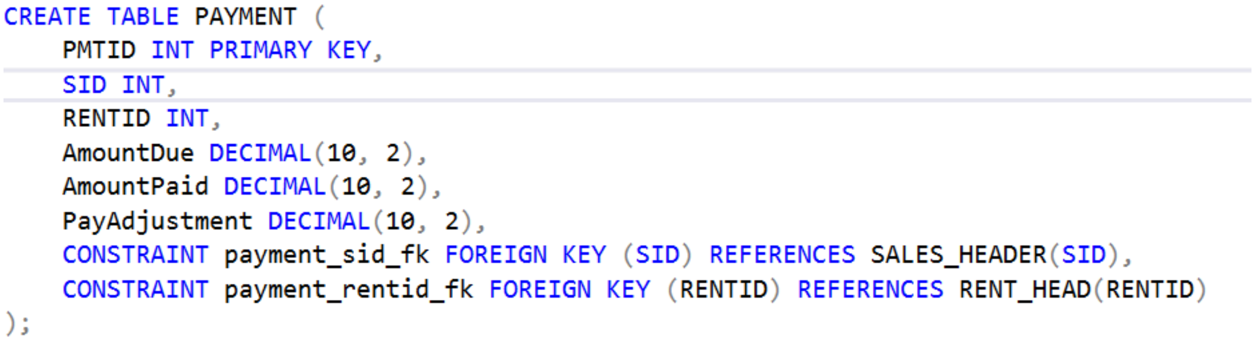
```sql
CREATE TABLE CUSTOMER (
    CUSTID INT PRIMARY KEY,
    FName VARCHAR(50) CONSTRAINT customer_fname_nn NOT NULL,
    LName VARCHAR(50) CONSTRAINT customer_lname_nn NOT NULL,
    Street VARCHAR(100),
    ZIP INT,
    CONSTRAINT customer_zip_fk FOREIGN KEY (ZIP) REFERENCES ZIP(ZIP)
);
```



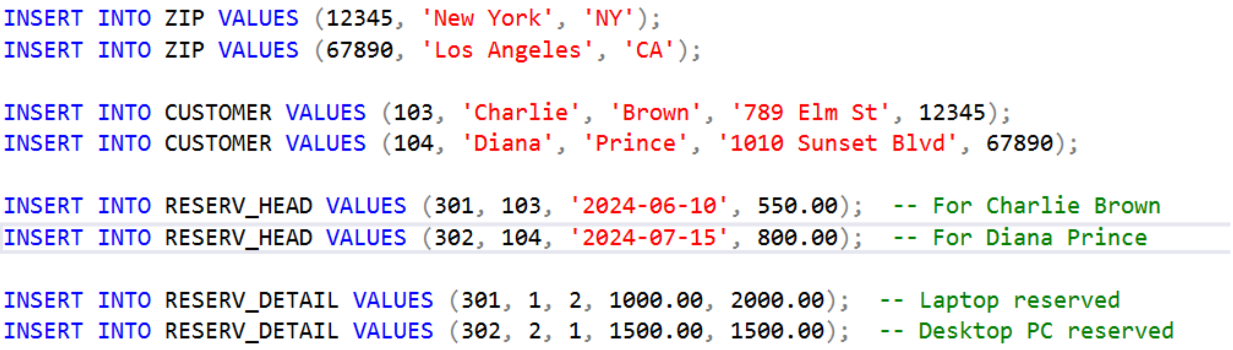
```sql
CREATE TABLE RENT_HEAD (
    RENTID INT PRIMARY KEY,
    RENT_Date DATE NOT NULL,
    Chk_Date DATE,
    Return_Date DATE,
    RSVID INT NOT NULL,
    RentAmount DECIMAL(10, 2),
    Tax DECIMAL(10, 2),
    GrandTot DECIMAL(10, 2),
    ReturnALL VARCHAR(3),
    CONSTRAINT rent_head_rsid_fk FOREIGN KEY (RSVID) REFERENCES RESERV_HEAD(RSVID)
);
```



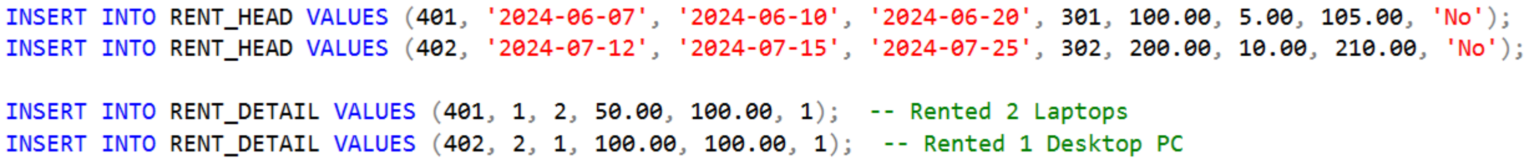
```sql
CREATE TABLE RENT_DETAIL (
    RENTID INT NOT NULL,
    EQPID INT NOT NULL,
    Quantity INT NOT NULL,
    CostPerUnit DECIMAL(10, 2) NOT NULL,
    LineTotal DECIMAL(10, 2) NOT NULL,
    Returned INT NOT NULL,
    CONSTRAINT rent_detail_pk PRIMARY KEY (RENTID, EQPID),
    CONSTRAINT rent_detail_rentid_fk FOREIGN KEY (RENTID) REFERENCES RENT_HEAD(RENTID),
    CONSTRAINT rent_detail_eqpid_fk FOREIGN KEY (EQPID) REFERENCES EQUIPMENT(EQPID)
);
```



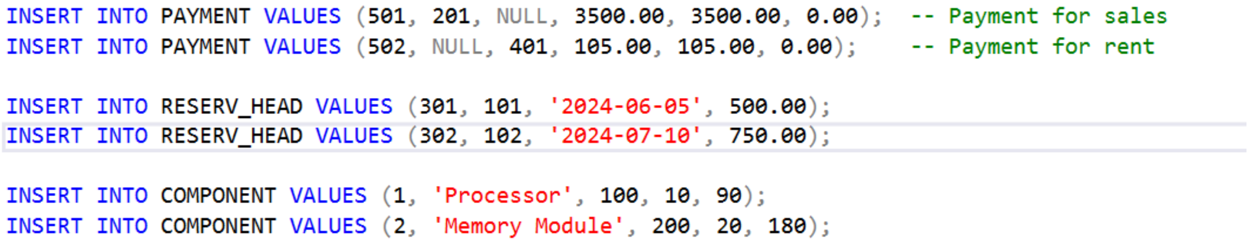
```sql
CREATE TABLE PAYMENT (
    PMTID INT PRIMARY KEY,
    SID INT,
    RENTID INT,
    AmountDue DECIMAL(10, 2),
    AmountPaid DECIMAL(10, 2),
    PayAdjustment DECIMAL(10, 2),
    CONSTRAINT payment_sid_fk FOREIGN KEY (SID) REFERENCES SALES_HEADER(SID),
    CONSTRAINT payment_rentid_fk FOREIGN KEY (RENTID) REFERENCES RENT_HEAD(RENTID)
);
```



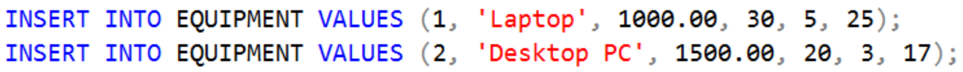
```sql
INSERT INTO ZIP VALUES (12345, 'New York', 'NY');
INSERT INTO ZIP VALUES (67890, 'Los Angeles', 'CA');

INSERT INTO CUSTOMER VALUES (103, 'Charlie', 'Brown', '789 Elm St', 12345);
INSERT INTO CUSTOMER VALUES (104, 'Diana', 'Prince', '1010 Sunset Blvd', 67890);

INSERT INTO RESERV_HEAD VALUES (301, 103, '2024-06-10', 550.00);  -- For Charlie Brown
INSERT INTO RESERV_HEAD VALUES (302, 104, '2024-07-15', 800.00);  -- For Diana Prince

INSERT INTO RESERV_DETAIL VALUES (301, 1, 2, 1000.00, 2000.00);  -- Laptop reserved
INSERT INTO RESERV_DETAIL VALUES (302, 2, 1, 1500.00, 1500.00);  -- Desktop PC reserved
```



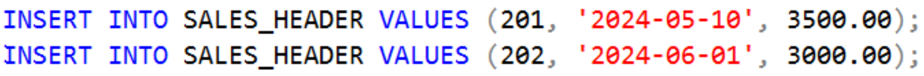
```sql
INSERT INTO RENT_HEAD VALUES (401, '2024-06-07', '2024-06-10', '2024-06-20', 301, 100.00, 5.00, 105.00, 'No');
INSERT INTO RENT_HEAD VALUES (402, '2024-07-12', '2024-07-15', '2024-07-25', 302, 200.00, 10.00, 210.00, 'No');

INSERT INTO RENT_DETAIL VALUES (401, 1, 2, 50.00, 100.00, 1);  -- Rented 2 Laptops
INSERT INTO RENT_DETAIL VALUES (402, 2, 1, 100.00, 100.00, 1);  -- Rented 1 Desktop PC
```



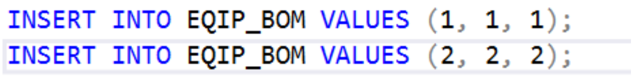
```sql
INSERT INTO PAYMENT VALUES (501, 201, NULL, 3500.00, 3500.00, 0.00);  -- Payment for sales
INSERT INTO PAYMENT VALUES (502, NULL, 401, 105.00, 105.00, 0.00);    -- Payment for rent

INSERT INTO RESERV_HEAD VALUES (301, 101, '2024-06-05', 500.00);
INSERT INTO RESERV_HEAD VALUES (302, 102, '2024-07-10', 750.00);

INSERT INTO COMPONENT VALUES (1, 'Processor', 100, 10, 90);
INSERT INTO COMPONENT VALUES (2, 'Memory Module', 200, 20, 180);
```



```sql
INSERT INTO EQUIPMENT VALUES (1, 'Laptop', 1000.00, 30, 5, 25);
INSERT INTO EQUIPMENT VALUES (2, 'Desktop PC', 1500.00, 20, 3, 17);
```



```sql
INSERT INTO SALES_HEADER VALUES (201, '2024-05-10', 3500.00);
INSERT INTO SALES_HEADER VALUES (202, '2024-06-01', 3000.00);
```



```sql
INSERT INTO EQIP_BOM VALUES (1, 1, 1);
INSERT INTO EQIP_BOM VALUES (2, 2, 2);
```

Relevance

This database structure models secure and auditable systems used in real life:

KYC & Identity Verification: Mirrors how platforms link users to services, with referential integrity preventing duplicate or fraudulent data.

Access Control & Permissions: Instructors and students represent roles with access to specific content or systems.

Audit-Ready Reporting: Structured data supports compliance tracking and monitoring.

Enterprise Systems: Similar schema used in LMS, HRIS, and AML case management tools.

Key Skills Demonstrated

Relational database design from business rules

Use of foreign keys and composite keys to enforce data rules

Many-to-many relationship resolution

SQL scripting for backend systems

Understanding of normalized, scalable data models