MAKING TABLES

CUSTOMER

```
CREATE TABLE CUSTOMER (
   customerID INT PRIMARY KEY,
   fname VARCHAR(50),
   lname VARCHAR(50),
   email VARCHAR(100),
   phone VARCHAR(20),
   gender CHAR(1),
   dob DATE,
   joindate DATE,
   classbalance INT
);
```

CLASS

```
CREATE TABLE CLASS (
   classID INT PRIMARY KEY,
   classname VARCHAR(100),
   classtemp DECIMAL(5, 2),
   startTime TIME,
   endTime TIME,
   classDate DATE,
   spotsAvailable INT,
   instructorID INT,
   FOREIGN KEY (instructorID) REFERENCES INSTRUCTOR(instructorID);
```

CLASSPACKAGE

```
CREATE TABLE CLASSPACKAGE (
   classpackageID INT PRIMARY KEY,
   classpackagedesc VARCHAR(255),
   classpackageprice DECIMAL(10, 2),
   numClasses INT
);
```

INSTRUCTOR

```
CREATE TABLE INSTRUCTOR (
   instructorID INT PRIMARY KEY,
   instfname VARCHAR(50),
   instlname VARCHAR(50),
   instphone VARCHAR(20),
   instdob DATE,
   inststartdate DATE,
   instgender CHAR(1)
);
```

SALE

```
CREATE TABLE SALE (
    saleID INT PRIMARY KEY,
    customerID INT,
    classpackageID INT,
    saleDate DATE,
    paymentType VARCHAR(50),
    FOREIGN KEY (customerID) REFERENCES CUSTOMER(customerID),
    FOREIGN KEY (classpackageID) REFERENCES CLASSPACKAGE(classpackageID)
);
```

REGISTER

```
CREATE TABLE REGISTER (
    registerDateID INT PRIMARY KEY,
    customerID INT,
    classID INT,
    registerDate DATE,
    FOREIGN KEY (customerID) REFERENCES CUSTOMER(customerID),
    FOREIGN KEY (classID) REFERENCES CLASS(classID)
);
```

TEACHES

```
CREATE TABLE INSTRUCTORPAYMENT (
    paymentID INT PRIMARY KEY AUTO_INCREMENT,
    instructorID INT,
    classID INT,
    paymentAmount DECIMAL(10, 2),
    paymentDate DATE,
    FOREIGN KEY (instructorID) REFERENCES INSTRUCTOR(instructorID),
    FOREIGN KEY (classID) REFERENCES CLASS(classID)
);
```

INSTRUCTORPAYMENT

```
CREATE TABLE INSTRUCTORPAYMENT (
    paymentID INT PRIMARY KEY AUTO_INCREMENT,
    instructorID INT,
    classID INT,
    paymentAmount DECIMAL(10, 2),
    paymentDate DATE,
    FOREIGN KEY (instructorID) REFERENCES INSTRUCTOR(instructorID),
    FOREIGN KEY (classID) REFERENCES CLASS(classID)
);
```

TABLE RELATIONS

1. **HANDLES INSTRUCTOR PAYMENT**

```
CREATE TRIGGER PayInstructorAfterTeach
AFTER INSERT ON TEACHES
FOR EACH ROW
BEGIN
   DECLARE payment DECIMAL(10, 2);
   SET payment = 50.00;  -- Fixed payment amount per class, modify as needed

   INSERT INTO INSTRUCTORPAYMENT (instructorID, classID, paymentAmount,
paymentDate)
   VALUES (NEW.instructorID, NEW.classID, payment, CURDATE());
END;
```

2. **UPDATES CUSTOMER CLASS BALANCE ON SALE**

```
CREATE TRIGGER UpdateClassBalanceAfterSale
AFTER INSERT ON SALE
FOR EACH ROW
BEGIN
   UPDATE CUSTOMER
   SET classbalance = classbalance + (SELECT numClasses FROM CLASSPACKAGE
WHERE classpackageID = NEW.classpackageID)
   WHERE customerID = NEW.customerID;
END;
```

3. **SUBTRACTS CUSTOMER CLASS BALANCE ON REGISTER AND DELETES AVAILABLE SPOTS FROM CLASS**

```
DELIMITER //

CREATE TRIGGER UpdateClassBalanceAndSpotsAfterRegister
AFTER INSERT ON REGISTER
FOR EACH ROW
BEGIN
   DECLARE spots INT;

   -- Decrease classbalance for the customer
   UPDATE CUSTOMER
   SET classbalance = classbalance - 1
```

```
    WHERE customerID = NEW.customerID;

    -- Decrease spotsAvailable for the class
    SELECT spotsAvailable INTO spots FROM CLASS WHERE classID = NEW.classID;

    IF spots > 0 THEN
        UPDATE CLASS
        SET spotsAvailable = spots - 1
        WHERE classID = NEW.classID;
    ELSE
        -- Handle no spots available error
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No spots available for registration';
    END IF;
END //

DELIMITER ;
```

CONSTRAINTS

```
ALTER TABLE SALE
ADD CONSTRAINT fk_sale_customer FOREIGN KEY (customerID) REFERENCES
CUSTOMER(customerID);

ALTER TABLE SALE
ADD CONSTRAINT fk_sale_classpackage FOREIGN KEY (classpackageID) REFERENCES
CLASSPACKAGE(classpackageID);

ALTER TABLE REGISTER
ADD CONSTRAINT fk_register_customer FOREIGN KEY (customerID) REFERENCES
CUSTOMER(customerID);

ALTER TABLE REGISTER
ADD CONSTRAINT fk_register_class FOREIGN KEY (classID) REFERENCES
CLASS(classID);

ALTER TABLE TEACHES
ADD CONSTRAINT fk_teaches_instructor FOREIGN KEY (instructorID) REFERENCES
INSTRUCTOR(instructorID);

ALTER TABLE TEACHES
ADD CONSTRAINT fk_teaches_class FOREIGN KEY (classID) REFERENCES
CLASS(classID);

ALTER TABLE INSTRUCTORPAYMENT
ADD CONSTRAINT fk_instructorpayment_instructor FOREIGN KEY (instructorID)
REFERENCES INSTRUCTOR(instructorID);

ALTER TABLE INSTRUCTORPAYMENT
```

```
ADD CONSTRAINT fk_instructorpayment_class FOREIGN KEY (classID) REFERENCES
CLASS(classID);
```

DATA ENTRY

CUSTOMERS

```
-- Sample data for inserting customers
INSERT INTO CUSTOMER (customerID, fname, lname, email, phone, gender, dob, joindate,
classbalance)
VALUES
  (1, 'John', 'Doe', 'john.doe@example.com', '1234567890', 'M', '1990-01-01', '2020-01-15', 5),
  (2, 'Jane', 'Smith', 'jane.smith@example.com', '0987654321', 'F', '1992-05-15', '2020-02-10',
10),
  (3, 'Michael', 'Johnson', 'michael.johnson@example.com', '5678901234', 'M', '1985-09-20',
'2019-12-20', 7),
  (4, 'Emily', 'Williams', 'emily.williams@example.com', '9876543210', 'F', '1988-03-12',
'2020-03-05', 3),
  (5, 'Daniel', 'Brown', 'daniel.brown@example.com', '2345678901', 'M', '1993-07-25',
'2020-04-18', 8),
  (6, 'Olivia', 'Davis', 'olivia.davis@example.com', '8765432109', 'F', '1991-11-30', '2020-05-02',
12),
  (7, 'William', 'Miller', 'william.miller@example.com', '3456789012', 'M', '1987-02-28',
'2020-06-08', 6),
  (8, 'Sophia', 'Wilson', 'sophia.wilson@example.com', '7654321098', 'F', '1995-04-17',
'2020-07-12', 4),
  (9, 'James', 'Moore', 'james.moore@example.com', '4567890123', 'M', '1989-08-10',
'2020-08-25', 9),
  (10, 'Emma', 'Taylor', 'emma.taylor@example.com', '6543210987', 'F', '1994-12-05',
'2020-09-01', 15),
  (11, 'Benjamin', 'Anderson', 'benjamin.anderson@example.com', '5432109876', 'M',
'1996-02-18', '2020-10-03', 11),
  (12, 'Mia', 'Thomas', 'mia.thomas@example.com', '4321098765', 'F', '1986-06-22', '2020-11-14',
7),
  (13, 'Elijah', 'Jackson', 'elijah.jackson@example.com', '3210987654', 'M', '1997-10-08',
'2020-12-30', 6),
  (14, 'Charlotte', 'White', 'charlotte.white@example.com', '2109876543', 'F', '1998-04-03',
'2021-01-05', 9),
  (15, 'Lucas', 'Harris', 'lucas.harris@example.com', '1098765432', 'M', '1990-11-19',
'2021-02-18', 8),
  (16, 'Ava', 'Martinez', 'ava.martinez@example.com', '0987654321', 'F', '1992-07-15',
'2021-03-21', 13),
  (17, 'Alexander', 'Garcia', 'alexander.garcia@example.com', '9876543210', 'M', '1988-03-12',
'2021-04-25', 5),
  (18, 'Madison', 'Lopez', 'madison.lopez@example.com', '8765432109', 'F', '1991-11-30',
'2021-05-30', 10),
  (19, 'Jacob', 'King', 'jacob.king@example.com', '7654321098', 'M', '1995-04-17', '2021-06-12',
7),
  (20, 'Isabella', 'Perez', 'isabella.perez@example.com', '6543210987', 'F', '1989-08-10',
'2021-07-18', 14),
  (21, 'William', 'Rivera', 'william.rivera@example.com', '5432109876', 'M', '1996-02-18',
'2021-08-22', 9),
  (22, 'Sophia', 'Young', 'sophia.young@example.com', '4321098765', 'F', '1986-06-22',
'2021-09-05', 6),
  (23, 'Ethan', 'Wright', 'ethan.wright@example.com', '3210987654', 'M', '1997-10-08',
'2021-10-10', 11),
  (24, 'Amelia', 'Scott', 'amelia.scott@example.com', '2109876543', 'F', '1998-04-03',
'2021-11-15', 8),
  (25, 'Oliver', 'Green', 'oliver.green@example.com', '1098765432', 'M', '1990-11-19',
'2021-12-20', 7);

-- Sample data for inserting classes
INSERT INTO CLASS (classID, classname, classtemp, startTime, endTime, classDate,
spotsAvailable)
VALUES
  -- Day 1: June 25, 2024
  (1, 'Yoga Basics', 32.50, '08:00:00', '09:00:00', '2024-06-25', 20),
  (2, 'Vinyasa Flow', 34.00, '10:00:00', '11:00:00', '2024-06-25', 15),
  (3, 'Hatha Yoga', 31.00, '12:00:00', '13:00:00', '2024-06-25', 18),
  (4, 'Power Yoga', 35.00, '14:00:00', '15:00:00', '2024-06-25', 22),
  (5, 'Restorative Yoga', 30.00, '16:00:00', '17:00:00', '2024-06-25', 25),

  -- Day 2: June 26, 2024
  (6, 'Yin Yoga', 33.00, '08:00:00', '09:00:00', '2024-06-26', 18),
  (7, 'Ashtanga Yoga', 36.00, '10:00:00', '11:00:00', '2024-06-26', 20),
  (8, 'Kundalini Yoga', 34.50, '12:00:00', '13:00:00', '2024-06-26', 15),
  (9, 'Chair Yoga', 28.00, '14:00:00', '15:00:00', '2024-06-26', 12),
  (10, 'Aerial Yoga', 37.00, '16:00:00', '17:00:00', '2024-06-26', 10),

  -- Day 3: June 27, 2024
  (11, 'Hot Yoga', 35.50, '08:00:00', '09:00:00', '2024-06-27', 22),
  (12, 'Iyengar Yoga', 32.00, '10:00:00', '11:00:00', '2024-06-27', 18),
  (13, 'AcroYoga', 36.50, '12:00:00', '13:00:00', '2024-06-27', 15),
  (14, 'Prenatal Yoga', 29.00, '14:00:00', '15:00:00', '2024-06-27', 20),
  (15, 'Yoga for Seniors', 27.50, '16:00:00', '17:00:00', '2024-06-27', 18),

  -- Day 4: June 28, 2024
  (16, 'Core Power Yoga', 34.00, '08:00:00', '09:00:00', '2024-06-28', 25),
  (17, 'Yoga Nidra', 30.50, '10:00:00', '11:00:00', '2024-06-28', 15),
  (18, 'Mindfulness Yoga', 31.50, '12:00:00', '13:00:00', '2024-06-28', 20),
  (19, 'Pilates', 33.00, '14:00:00', '15:00:00', '2024-06-28', 18),
  (20, 'Gentle Flow Yoga', 29.50, '16:00:00', '17:00:00', '2024-06-28', 22),

  -- Day 5: June 29, 2024
  (21, 'Beginner Yoga', 28.00, '08:00:00', '09:00:00', '2024-06-29', 20),
  (22, 'Advanced Yoga', 36.00, '10:00:00', '11:00:00', '2024-06-29', 15),
  (23, 'Yoga Therapy', 32.50, '12:00:00', '13:00:00', '2024-06-29', 18),
  (24, 'Yoga Sculpt', 35.00, '14:00:00', '15:00:00', '2024-06-29', 22),
  (25, 'Meditation Class', 25.00, '16:00:00', '17:00:00', '2024-06-29', 25);
```

Class Packages Examples

```
-- Sample data for inserting class packages
INSERT INTO CLASSPACKAGE (classpackageID, classpackagedesc, classpackageprice,
numClasses)
VALUES
  (1, 'Single Class', 18.00, 1),
  (2, '5 Class Pass', 80.00, 5),
  (3, '10 Class Pass', 140.00, 10);
```

-- Sample data for inserting instructors

```
INSERT INTO INSTRUCTOR (instructorID, instfname, instlname, instphone, instdob,
inststartdate, instgender)
VALUES
  (1, 'Alice', 'Smith', '123-456-7890', '1980-05-15', '2010-01-01', 'F'),
  (2, 'Bob', 'Johnson', '987-654-3210', '1975-08-20', '2008-06-15', 'M'),
  (3, 'Eva', 'Brown', '234-567-8901', '1983-02-10', '2012-03-20', 'F');
```

-- Sample data for inserting into SALE table

```
INSERT INTO SALE (saleID, customerID, classpackageID, saleDate, paymentType)
VALUES
  (1, 1, 2, '2024-06-25', 'Credit Card'),
  (2, 3, 3, '2024-06-26', 'Cash'),
  (3, 5, 1, '2024-06-27', 'Credit Card'),
  (4, 7, 2, '2024-06-28', 'Credit Card'),
  (5, 9, 3, '2024-06-29', 'Cash'),
  (6, 11, 2, '2024-06-30', 'Credit Card'),
  (7, 13, 1, '2024-07-01', 'Credit Card'),
  (8, 15, 3, '2024-07-02', 'Cash'),
  (9, 17, 2, '2024-07-03', 'Credit Card'),
  (10, 19, 1, '2024-07-04', 'Credit Card');
```

-- Sample data for inserting into REGISTER table

```
INSERT INTO REGISTER (registerDateID, customerID, classID, registerDate)
VALUES
  (1, 2, 6, '2024-06-25'),
  (2, 4, 8, '2024-06-25'),
  (3, 6, 10, '2024-06-26'),
  (4, 8, 12, '2024-06-26'),
  (5, 10, 14, '2024-06-27'),
  (6, 12, 16, '2024-06-27'),
  (7, 14, 18, '2024-06-28'),
  (8, 16, 20, '2024-06-28'),
  (9, 18, 22, '2024-06-29'),
  (10, 20, 24, '2024-06-29');
```

```
-- Sample data for inserting into TEACHES table.  AUTOupdates Instructor Payment
INSERT INTO TEACHES (instructorID, classID)
VALUES
  (1, 1),
  (2, 3),
  (3, 5),
  (1, 7),
  (2, 9),
  (3, 11),
  (1, 13),
  (2, 15),
  (3, 17),
  (1, 19);
```

EMPLOYEE

```
CREATE TABLE EMPLOYEE (
    employeeID INT PRIMARY KEY,
    empfname VARCHAR(50),
    emplname VARCHAR(50),
    role VARCHAR(50),
    phone VARCHAR(20)
);
```

WORK_SCHEDULE

```
CREATE TABLE WORK_SCHEDULE (
    scheduleID INT PRIMARY KEY,
    employeeID INT,
    workDate DATE,
    startTime TIME,
    endTime TIME,
    FOREIGN KEY (employeeID) REFERENCES EMPLOYEE(employeeID)
);
```

PAYMENT

```
CREATE TABLE PAYMENT (
    paymentID INT PRIMARY KEY,
    employeeID INT,
    paymentAmount DECIMAL(10, 2),
    paymentDate DATE,
    FOREIGN KEY (employeeID) REFERENCES EMPLOYEE(employeeID)
);
```

CALCULATE PAYMENTS

```
CREATE TRIGGER CalculatePayment
AFTER INSERT ON WORK_SCHEDULE
FOR EACH ROW
BEGIN
    DECLARE hoursWorked DECIMAL(5, 2);
    DECLARE hourlyRate DECIMAL(10, 2);  -- Adjust based on actual payment policies

    -- Calculate hours worked (assuming hourly rate is $15.00)
    SET hoursWorked = TIMESTAMPDIFF(HOUR, NEW.startTime, NEW.endTime);
    SET hourlyRate = 15.00;

    -- Calculate payment amount
    INSERT INTO PAYMENT (employeeID, paymentAmount, paymentDate)
    VALUES (NEW.employeeID, hoursWorked * hourlyRate, CURDATE());
END;
```

DATA EXAMPLE FOR EASIER ENTRIES

EMPLOYEES, roles are for Desk Employee or Cleaner

```
INSERT INTO EMPLOYEE (employeeID, empfname, emplname, role, phone)
VALUES
  (1, 'John', 'Smith', 'Desk Employee', '111-222-3333'),
  (2, 'Alice', 'Johnson', 'Cleaner', '555-666-7777');
```

SCHEDULE

```
-- Sample data for work schedules (adjust as per actual schedules)
INSERT INTO WORK_SCHEDULE (scheduleID, employeeID, workDate, startTime, endTime)
VALUES
  (1, 1, '2024-06-25', '08:00:00', '22:00:00'),  -- John Smith's schedule
  (2, 2, '2024-06-25', '08:00:00', '12:00:00'),  -- Alice Johnson's schedule
  (3, 2, '2024-06-25', '13:00:00', '17:00:00');  -- Alice Johnson's second shift
```