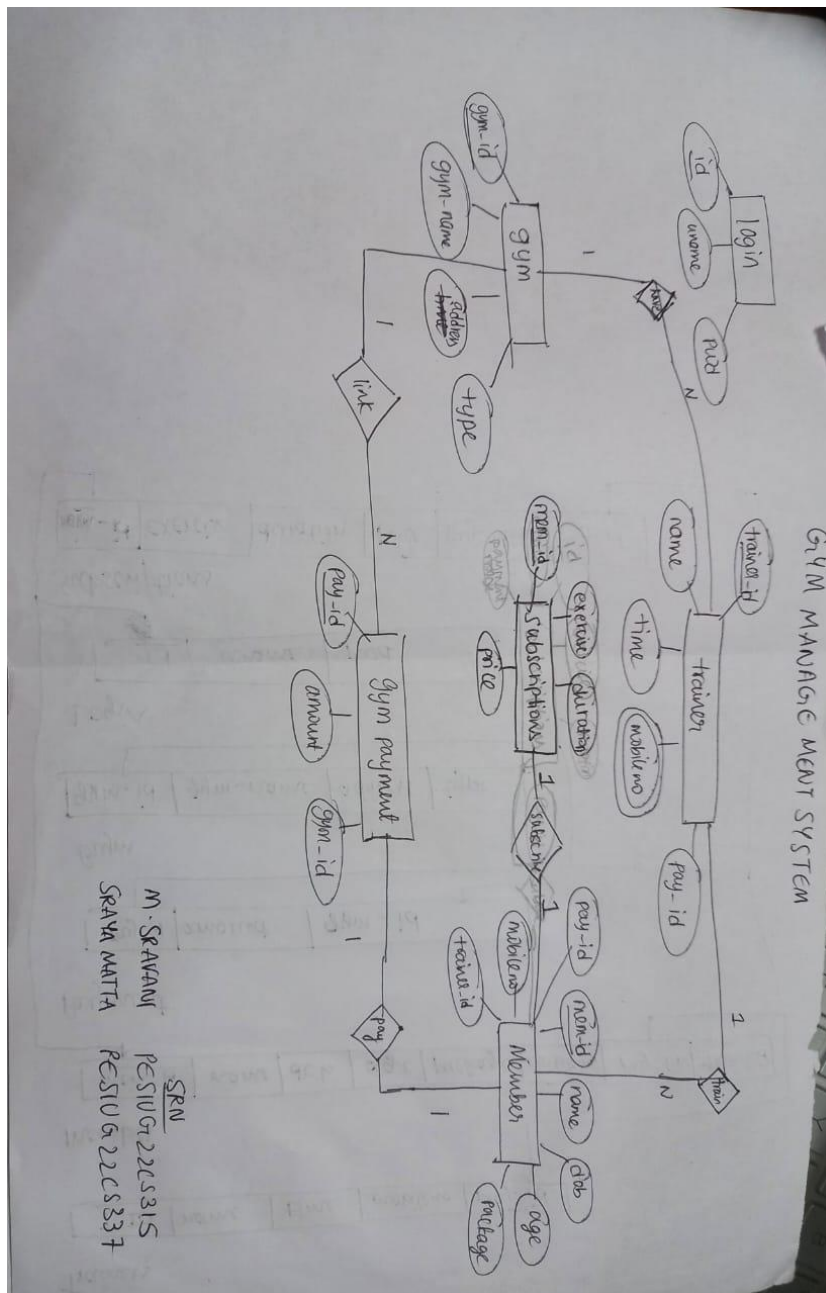


DBMS PROJECT FINAL REPORT

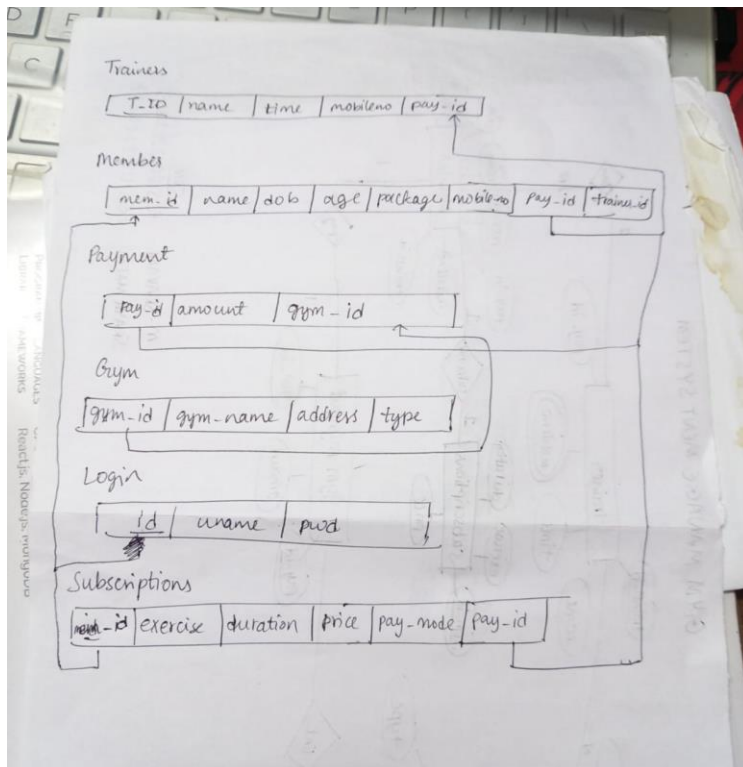
PROJECT TITLE :GYM MANAGEMENT SYSTEM

TEAM MEMBERS	SRN
M.SRAVANI	PES1UG22CS315
SRAVYA MATTA	PES1UG22CS337

E-R Diagram



Relational Schema



Code snippets and screenshots for queries

Create Commands

```
CREATE TABLE `gym` (
  `gym_id` varchar(20) NOT NULL,
  `gym_name` varchar(30) NOT NULL,
  `address` varchar(150) NOT NULL,
  `type` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `gym`
--

INSERT INTO `gym` (`gym_id`, `gym_name`, `address`, `type`) VALUES
('GYM1', 'GYM LAND', 'Shiv Nagar', 'men'),
('GYM2', 'TARGET ZONE', 'Shanthi Nagar', 'unisex'),
('GYM3', 'GEORGE GYM', 'Mahesh Nagar', 'unisex'),
('GYM4', 'SUNNY GYM FITNESS STATION', 'Rupali Complex', 'women'),
('GYM5', 'A3 FITNESS GYM', 'Ramnagar Colony', 'men'),
('GYM6', 'SHAPE GYM', 'Zion Colony', 'unisex'),
('GYM7', 'TITAN GYM', 'Old City', 'women'),
('GYM8', 'TIGERS TOP GYM', 'Madival Circle', 'men');
```

```

CREATE TABLE `login` (
  `id` int(10) NOT NULL,
  `uname` varchar(30) NOT NULL,
  `pwd` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `login`
--

INSERT INTO `login` (`id`, `uname`, `pwd`) VALUES
(1, 'admin', 'admin');

```

```

CREATE TABLE `member` (
  `mem_id` varchar(20) NOT NULL,
  `name` varchar(30) DEFAULT NULL,
  `dob` varchar(20) DEFAULT NULL,
  `age` varchar(20) DEFAULT NULL,
  `package` varchar(10) DEFAULT NULL,
  `mobilen0` varchar(10) DEFAULT NULL,
  `pay_id` varchar(20) DEFAULT NULL,
  `trainer_id` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `member`
--

INSERT INTO `member` (`mem_id`, `name`, `dob`, `age`, `package`, `mobilen0`, `pay_id`, `trainer_id`) VALUES
('M1', 'Aditya', '18/08/1994', '26', '5200', '8888888888', 'Payment1', 'T1'),
('M2', 'Karan', '26/06/1998', '21', '4800', '9988998899', 'Payment2', 'T2'),
('M3', 'Chirag', '22/07/1997', '22', '6400', '9977997799', 'Payment3', 'T3'),
('M4', 'Abhishek', '21/08/1998', '21', '5400', '9966996699', 'Payment4', 'T4'),
('M5', 'Veeresh', '24/06/1999', '20', '6000', '9955995599', 'Payment5', 'T5');

```

```

CREATE TABLE `payment` (
  `pay_id` varchar(20) NOT NULL,
  `amount` varchar(20) DEFAULT NULL,
  `gym_id` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `payment`
--

INSERT INTO `payment` (`pay_id`, `amount`, `gym_id`) VALUES
('Payment1', '5200', 'GYM1'),
('Payment2', '4800', 'GYM2'),
('Payment3', '6400', 'GYM3'),
('Payment4', '5400', 'GYM4'),
('Payment5', '6000', 'GYM5'),
('Payment6', '4500', 'GYM6'),
('Payment7', '5500', 'GYM7'),
('Payment8', '6100', 'GYM8');

```

```

-----

CREATE TABLE `trainer` (
  `trainer_id` varchar(20) NOT NULL,
  `name` varchar(20) DEFAULT NULL,
  `time` varchar(10) DEFAULT NULL,
  `mobilen0` varchar(10) DEFAULT NULL,
  `pay_id` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `trainer`
--

INSERT INTO `trainer` (`trainer_id`, `name`, `time`, `mobilen0`, `pay_id`) VALUES
('T1', 'George', '5:00 AM', '9999999999', 'Payment1'),
('T2', 'Tanveer', '9:00 AM', '8888888888', 'Payment2'),
('T3', 'Wong Lee', '11:00 AM', '7777777777', 'Payment3'),
('T4', 'Kiran Das', '1:00 PM', '6666666666', 'Payment6'),
('T5', 'Harry Styles', '3:00 PM', '6655665566', 'Payment5'),
('T6', 'James Corden', '5:00 PM', '6677667766', 'Payment6'),
('T7', 'Jimmy Kimmel', '7:00 PM', '6688668866', 'Payment7'),
('T8', 'Ray Berlin', '9:00 PM', '6699669966', 'Payment8');

```

```
CREATE TABLE `subscriptions` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `exercise` varchar(50) NOT NULL,  
  `duration` int(11) NOT NULL,  
  `price` decimal(10,2) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
INSERT INTO `subscriptions` (`exercise`, `duration`, `price`) VALUES  
(`Zumba`, 1, 10000.00),  
(`HRX Workout`, 3, 25000.00),  
(`Boxing`, 6, 57000.00),  
(`Weightlifting`, 12, 100000.00),  
(`Yoga`, 1, 8000.00),  
(`HRX Workout`, 1, 12000.00),  
(`HRX Workout`, 1, 10000.00),  
(`Zumba`, 1, 10000.00),  
(`Boxing`, 3, 28350.00),  
(`Zumba`, 1, 10000.00),  
(`HRX Workout`, 6, 60000.00),  
(`Boxing`, 1, 10500.00),  
(`Zumba`, 1, 10000.00),  
(`Zumba`, 1, 10000.00),
```

```
--  
-- Indexes for table `gym`  
--  
ALTER TABLE `gym`  
  ADD PRIMARY KEY (`gym_id`);  
  
--  
-- Indexes for table `login`  
--  
ALTER TABLE `login`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- Indexes for table `member`  
--  
ALTER TABLE `member`  
  ADD PRIMARY KEY (`mem_id`),  
  ADD KEY `pay_id` (`pay_id`),  
  ADD KEY `trainer_id` (`trainer_id`);  
  
--
```

```
-- Indexes for table `member`
--
ALTER TABLE `member`
  ADD PRIMARY KEY (`mem_id`),
  ADD KEY `pay_id` (`pay_id`),
  ADD KEY `trainer_id` (`trainer_id`);

--
-- Indexes for table `payment`
--
ALTER TABLE `payment`
  ADD PRIMARY KEY (`pay_id`),
  ADD KEY `gym_id` (`gym_id`);

--
-- Indexes for table `trainer`
--
ALTER TABLE `trainer`
  ADD PRIMARY KEY (`trainer_id`),
  ADD KEY `pay_id` (`pay_id`);

--
```

```

-- AUTO_INCREMENT for table login
--
ALTER TABLE `login`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `member`
--
ALTER TABLE `member`
  ADD CONSTRAINT `member_ibfk_1` FOREIGN KEY (`pay_id`) REFERENCES `payment` (`pay_id`),
  ADD CONSTRAINT `member_ibfk_2` FOREIGN KEY (`trainer_id`) REFERENCES `trainer` (`trainer_id`);

--
-- Constraints for table `payment`
--
ALTER TABLE `payment`
  ADD CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`gym_id`) REFERENCES `gym` (`gym_id`);

--
-- Constraints for table `payment`
--
ALTER TABLE `payment`
  ADD CONSTRAINT `payment_ibfk_1` FOREIGN KEY (`gym_id`) REFERENCES `gym` (`gym_id`);

--
-- Constraints for table `trainer`
--
ALTER TABLE `trainer`
  ADD CONSTRAINT `trainer_ibfk_1` FOREIGN KEY (`pay_id`) REFERENCES `payment` (`pay_id`);
COMMIT;

```

UPDATE

```
-- update queries
```

```
UPDATE gym
```

```
SET address = 'New Shiv Nagar'
```

```
WHERE gym_id = 'GYM1';
```

```
UPDATE member
```

```
SET package = '6000', age = '27'
```

```
WHERE mem_id = 'M1';
```

```
UPDATE trainer
```

```
SET mobileno = '9876543210'
```

```
WHERE trainer_id = 'T1';
```

```
UPDATE subscriptions
```

```
SET price = 11000.00
```

```
WHERE exercise = 'Zumba' AND duration = 1;
```

```
DELETE FROM gym
```

DELETE

```
DELETE FROM gym
```

```
WHERE gym_id = 'GYM8';
```

```
DELETE FROM member
```

```
WHERE mem_id = 'M2';
```

```
DELETE FROM trainer
```

```
WHERE trainer_id = 'T3';
```

TRIGGER FOR CHECKING BEFORE MEMBER INSERT

```
DELIMITER //
```

```
CREATE TRIGGER before_member_insert
BEFORE INSERT ON member
FOR EACH ROW
BEGIN
    -- Check if the member is associated with any subscription
    DECLARE subscription_count INT;

    -- Ensure member has at least one subscription record
    SELECT COUNT(*) INTO subscription_count
    FROM subscriptions
    WHERE mem_id = NEW.mem_id;

    -- If no subscription exists, raise an error
    IF subscription_count = 0 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No valid subscription found for the member!';
    END IF;
END;

//
DELIMITER ;
```

```
INSERT INTO member (mem_id, name, dob, age, package, mobileno, pay_id, trainer_id)
VALUES ('M7', 'John Doe', '1995-05-15', 28, '67890', '9998887777', 'Payment7', 'T6');
```

STORED PROCEDURE FOR ADD SUBSCRIPTION

```
-- Create Stored Procedure
DELIMITER //
```

```
> CREATE PROCEDURE AddSubscription(
    IN exercise_name VARCHAR(50),
    IN duration INT,
    IN price DECIMAL(10, 2)
- )
- )
> BEGIN
    INSERT INTO subscriptions (exercise, duration, price)
    VALUES (exercise_name, duration, price);
- END;
//
DELIMITER ;

CALL AddSubscription('HRX Workout', 6, 18000.00);
```

FUNCTION TO GET PAYMENT AMOUNT WITH PAYMENT ID

```
-- FUNCTION FOR GETTING PAYMENT AMOUNT BASED ON ID
DELIMITER $$

CREATE FUNCTION get_payment_amount(payment_id VARCHAR(20))
RETURNS VARCHAR(20)
READS SQL DATA
BEGIN
    DECLARE payment_amount VARCHAR(20);

    SELECT amount INTO payment_amount
    FROM payment
    WHERE pay_id = payment_id;

    RETURN payment_amount;
END $$

DELIMITER ;
```

```
SELECT get_payment_amount('Payment1') AS PaymentAmount;
```

Nested and join

```

-- Sample Queries
-- Nested Query
SELECT *
FROM subscriptions
WHERE exercise = (
    SELECT exercise
    FROM subscriptions
    ORDER BY price DESC
    LIMIT 1
);

-- Join Query
SELECT m.name, s.exercise, s.duration, s.price
FROM member m
JOIN subscriptions s ON m.mem_id = s.mem_id;

-- JOIN QUERY WITH TABLES SUBSCRIPTIONS, TRAINER, PAYMENT
SELECT m.name, t.name, s.exercise, s.duration, s.price, p.amount
FROM member m
INNER JOIN subscriptions s ON m.mem_id = s.mem_id
INNER JOIN trainer t ON m.trainer_id = t.trainer_id
INNER JOIN payment p ON m.mem_id = p.gym_id;

-- Aggregate Query
SELECT SUM(price) AS total_revenue
FROM subscriptions;

```

PROCEDURE TO CALCULATE TOTAL COST

```

-- procedure
-- Example: Procedure to calculate the total price for a member
DELIMITER $$

CREATE PROCEDURE calculate_total_price(IN member_id INT)
BEGIN
    SELECT SUM(price) AS total_price
    FROM subscriptions
    WHERE mem_id = member_id;
END $$

DELIMITER ;

-- Call the procedure for member with ID 1
CALL calculate_total_price(1);

```

Aggregate query to get number of subscriptions(exercise type) and average price

```

- aggregate
- Get the number of subscriptions per exercise type
SELECT exercise, COUNT(*) AS number_of_subscriptions
FROM subscriptions
GROUP BY exercise;

- average
- Get the average price of subscriptions
SELECT AVG(price) AS average_price
FROM subscriptions;

```