**QUERIES USED TO GET DATA THESE ARE DML QUERIES**

GET EMPLOYEE DATA (THIS REPRESENTS WHO APPROVED THE MEMBER?

SELECT \* FROM `employee`;

GET LOCATION DATA ( THIS FETCHES FOR THE LOCATION )

SELECT \* FROM `location`

INSERT NEW MEMBER

CHECKS IF THE MEMBER EXISTS

SELECT \* FROM member WHERE fname = '" + fname + "' AND lname = '" + lname + "'

IF MEMBER EXISTS IT ALERTS USER

INSERTS INTO MEMBER TABLE

INSERT INTO member(phone, fname, lname, locationID, dob, age, address) VALUES('" + phone + "', '" + fname + "', '" + lname + "', " + location + ", '" + dob + "', " + age + ", '" + address + "')

INSERTS INTO MEMBERSHIP TYPE TABLE

INSERT INTO `membership\_type` (`name`, `employeeID`, `memberID`, `date\_created`, `start\_date`, `end\_date`, `type`, `status`) VALUES ('" + plan + "', '" + approved\_by + "', '" + memberID + "', NOW(), '" + start\_date + "', '" + end\_date + "', '" + payment\_type + "', 'Active')

INSERT INTO FEE TABLE

INSERT INTO `fee` (`member\_typeID`, `amount`, `payment\_type`, `due\_date`, `invoice\_number`, `discount`) VALUES ('" + member\_typeID + "', '" + fee + "', '" + payment\_type + "', '" + due\_date + "', '" + invoice\_number + "', '" + discount + "')

FETCH AND DISPLAY MEMBERS IN A TABLE

SELECT membership\_type.memberID, member.fname AS member\_fname, member.lname AS memberlname, member.phone, member.dob, member.age, member.address, name AS membership\_type,employee.fname AS employee\_fname, employee.lname AS employee\_lname, start\_date, end\_date, type AS payment\_type, fee.amount, fee.due\_date, fee.invoice\_number, fee.discount FROM `membership\_type` LEFT JOIN employee ON membership\_type.employeeID = employee.employeeID LEFT JOIN member ON membership\_type.memberID = member.memberID LEFT JOIN fee on fee.member\_typeID = membership\_type.member\_typeID

**CREATE TABLE QUERIES THESE ARE DDL QUERIES**

**CREATE MEMBERSHIP TYPE TABLE**

CREATE TABLE `membership\_type` (  `member\_typeID` int(11) NOT NULL AUTO\_INCREMENT,  `name` varchar(20) NOT NULL,  `employeeID` int(11) NOT NULL,  `memberID` int(11) NOT NULL,  `date\_created` varchar(20) NOT NULL,  `start\_date` varchar(20) NOT NULL,  `end\_date` varchar(20) NOT NULL,  `type` varchar(255) NOT NULL,  `status` varchar(20) NOT NULL,  PRIMARY KEY (`member\_typeID`),  KEY `employeeID` (`employeeID`),  KEY `memberID` (`memberID`),  CONSTRAINT `membership\_type\_ibfk\_1` FOREIGN KEY (`employeeID`) REFERENCES `employee` (`employeeID`),  CONSTRAINT `membership\_type\_ibfk\_2` FOREIGN KEY (`memberID`) REFERENCES `member` (`memberID`) )

**CREATE EMPLOYEE TABLE**

CREATE TABLE `employee` (  `employeeID` int(11) NOT NULL AUTO\_INCREMENT,  `fname` varchar(20) NOT NULL,  `lname` varchar(20) NOT NULL,  PRIMARY KEY (`employeeID`) )

**CREATE FEE TABLE**

CREATE TABLE `fee` (  `feeID` int(11) NOT NULL AUTO\_INCREMENT,  `member\_typeID` int(11) NOT NULL,  `amount` double(10,2) NOT NULL,  `payment\_type` varchar(20) NOT NULL,  `due\_date` varchar(20) NOT NULL,  `invoice\_number` varchar(255) NOT NULL,  `discount` double(10,2) NOT NULL,  PRIMARY KEY (`feeID`),  KEY `member\_typeID` (`member\_typeID`),  CONSTRAINT `fee\_ibfk\_1` FOREIGN KEY (`member\_typeID`) REFERENCES `membership\_type` (`member\_typeID`) )

**CREATE TABLE location**

CREATE TABLE `location` (  `locationID` int(11) NOT NULL AUTO\_INCREMENT,  `street\_number` int(20) NOT NULL,  `street\_name` varchar(255) NOT NULL,  `city` varchar(20) NOT NULL,  `postal\_code` varchar(20) NOT NULL,  `phone\_number` int(10) NOT NULL,  `employee\_number` varchar(20) NOT NULL,  PRIMARY KEY (`locationID`) ) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8mb

**CREATE TABLE MEMBER**

CREATE TABLE `member` (  `memberID` int(11) NOT NULL AUTO\_INCREMENT,  `phone` varchar(20) NOT NULL,  `fname` varchar(20) NOT NULL,  `lname` varchar(20) NOT NULL,  `locationID` int(11) NOT NULL,  `dob` varchar(10) NOT NULL,  `age` int(100) NOT NULL,  `address` varchar(200) NOT NULL,  PRIMARY KEY (`memberID`),  KEY `locationID` (`locationID`),  CONSTRAINT `member\_ibfk\_1` FOREIGN KEY (`locationID`) REFERENCES `location` (`locationID`) )