Project: Store Management System - SQL Database Schema

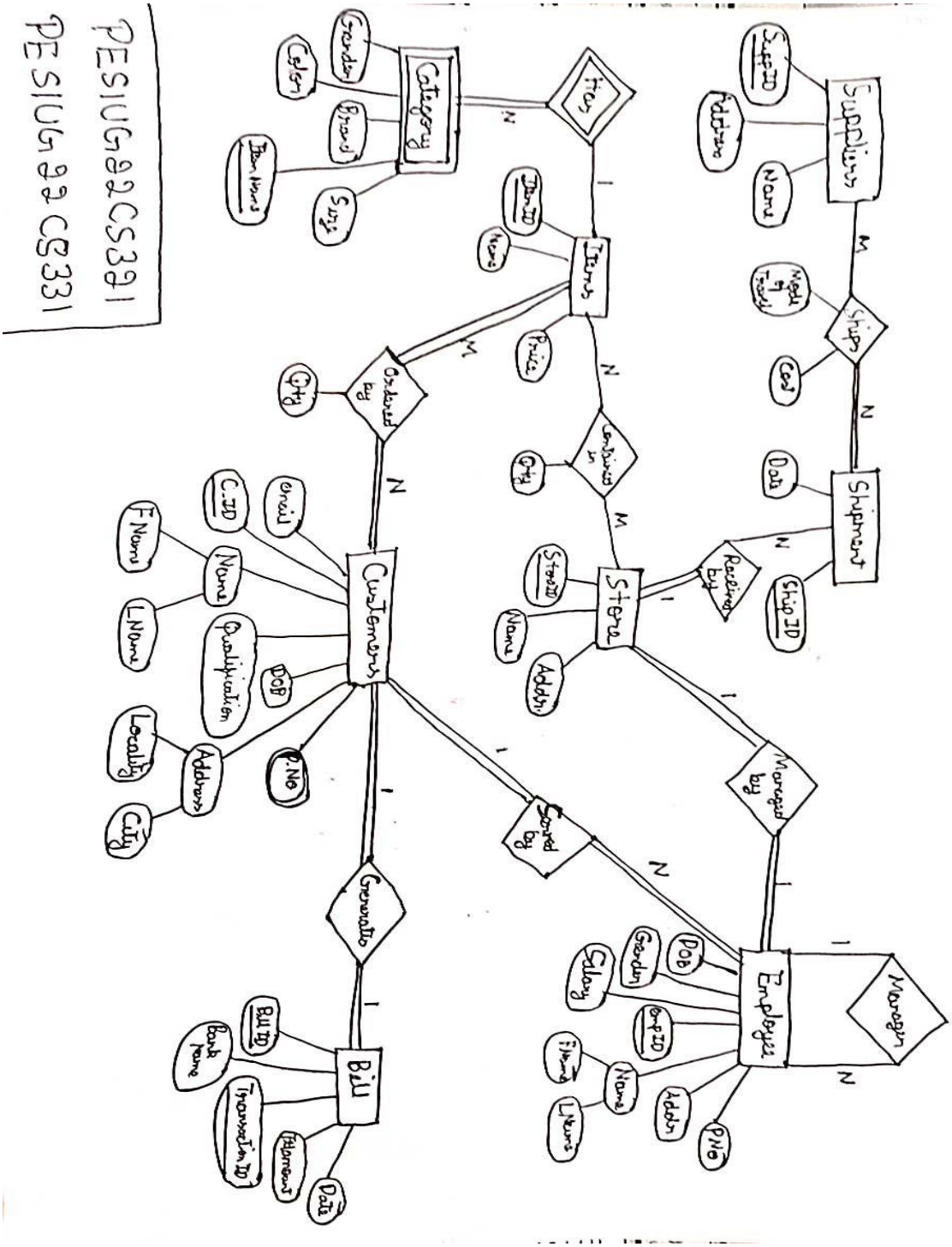
1. SHORT DESCRIPTION AND SCOPE OF THE PROJECT

Description : This is a mini project on store management system, where a complete database consists of the stores(branches), items in each store, customer records, employees present and their respective managers too. In addition, we also have a large number of items with varieties of categories and collections. Also, we have a mapping of customers buying the respective products and stores with the stock of particular items. We also have the record of how the items come to the store from the suppliers, the mode of travelling and tracking the order with the date. A couple of triggers, functions, procedures and cursors are implemented in order to make the working of database in a smooth and sophisticated way. Views are also implemented to show the dynamic changes that occurs in the database as a whole. This database handles for 4 branches of the Store and this essence are completely felt, when we perform some of join, aggregate and set operations. Overall, we get to see a miniature of real – time store management.

Scope : This mini project aims to deliver the real – time essence of store management, which is prevalent and widely developed by many software enthusiasts all over the world. The relational database is quiet deeply modelled and revealed in a very schematic way. The output of this DBMS is put out in a technical sense to bring out the effectiveness and usefulness of such a model and system. Also, a front end for this project using streamlit is made in order to access and perform CRUD

operations in simple way. This also creates a high reachability of the software. Hence these are few of the scopes of this project. Though the challenges to build the queries are high, MySQL rather supports all these in a sophistic way with the help of the reserved keywords and clauses provided by them.

2. E-R Diagram:



3. Database:

A) Tables:

- **Customers**: Stores details about customers, including their ID, name, qualification, address, contact details, and the employee handling them.
- **Employee**: Contains employee information such as ID, name, manager ID, gender, salary, date of birth, and contact details.
- **Store**: Defines the stores with store ID, name, address, and the manager responsible.
- **Items**: Catalog of items available in the store, including item ID, name, and price.
- **Orders**: Tracks customer orders with details like customer ID, item ID, quantity, and order amount.
- **Shipment**: Handles shipment details, linking with stores and shipment providers.
- **Suppliers**: Contains information about suppliers and the items they provide.
- **Bill**: Records billing information related to customer purchases.
- **Customer Backup Log**: Keeps a log of customer data for backup purposes.

B)Creating tables and inserting data:

```
CREATE TABLE `bill` (
  `B_ID` int(11) NOT NULL,
  `BANK` varchar(20) DEFAULT NULL,
  `DATE_OF_BILL` date DEFAULT NULL,
  `TRANSACTION_ID` varchar(20) DEFAULT NULL,
  `AMOUNT` float NOT NULL,
  `C_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `bill`
--

INSERT INTO `bill` (`B_ID`, `BANK`, `DATE_OF_BILL`, `TRANSACTION_ID`,
`AMOUNT`, `C_ID`) VALUES
(9001, '', '2022-12-09', '', 25997, 2001),
(9002, 'HDFC', '2022-10-16', '98765432102', 8348, 2002),
(9003, 'SBI', '2022-10-25', '34676876876', 20800, 2003),
(9004, 'HDFC', '2022-08-22', '46548767588', 9500, 2004),
(9005, 'SBI', '2022-10-09', '89876543876', 4250, 2005),
(9006, 'HDFC', '2022-10-07', '42637267463', 22592, 2006),
(9007, 'ICICI', '2022-08-21', '12348765454', 30797, 2007),
(9008, 'HDFC', '2022-07-19', '43644783600', 28200, 2008),
(9009, 'SBI', '2022-08-31', '52343477809', 6500, 2009),
(9010, 'ICICI', '2022-08-02', '32356654896', 7350, 2010),
(9011, 'SBI', '2022-10-07', '32456487876', 15588, 2011),
(9012, 'HDFC', '2022-10-17', '34678871313', 14995, 2012),
(9013, 'ICICI', '2022-09-18', '54467945789', 10250, 2013),
(9014, 'ICICI', '2022-08-29', '32869248768', 11200, 2014),
(9015, 'SBI', '2022-10-30', '34557897968', 20500, 2015),
(9016, 'HDFC', '2022-08-07', '45467884334', 29300, 2016),
(9017, 'HDFC', '2022-09-23', '34879565998', 32250, 2017),
(9018, 'SBI', '2022-10-27', '12350767426', 14396, 2018),
(9019, 'ICICI', '2022-11-23', '65748392102', 19600, 2019),
(9020, 'ICICI', '2022-08-14', '34365587090', 24900, 2020),
(9021, 'HDFC', '2022-11-23', '89076543245', 18197, 2021),
(9022, 'NA', '2022-11-23', 'Paid Through Cash', 10597, 2022);

-- --------------------------------------------------------

--
-- Stand-in structure for view `bills`
-- (See below for the actual view)
--
CREATE TABLE `bills` (
`C_ID` int(11)
```

```
,`First_Name` varchar(50)
,`Last_Name` varchar(50)
,`Item_ID` int(11)
,`Item_Name` varchar(30)
,`Brand` varchar(20)
,`Size` varchar(10)
,`Quantity` int(5)
,`TOTAL` double
,`DOP` date
);

-- --------------------------------------------------------

--
-- Table structure for table `contains`
--

CREATE TABLE `contains` (
  `STORE_ID` int(11) NOT NULL,
  `ITEM_ID` int(11) NOT NULL,
  `Quantity` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `contains`
--

INSERT INTO `contains` (`STORE_ID`, `ITEM_ID`, `Quantity`) VALUES
(6001, 4002, 32),
(6001, 4011, 32),
(6001, 4041, 30),
(6001, 4050, 29),
(6001, 4081, 29),
(6001, 4082, 34),
(6001, 4090, 26),
(6001, 4100, 37),
(6001, 4101, 32),
(6001, 4111, 33),
(6001, 4130, 22),
(6001, 4161, 25),
(6001, 4162, 26),
(6001, 4170, 27),
(6001, 4181, 36),
(6001, 4201, 24),
(6001, 4300, 30),
(6002, 4010, 46),
(6002, 4030, 43),
(6002, 4042, 43),
```

```
(6002, 4061, 44),
(6002, 4083, 41),
(6002, 4090, 46),
(6002, 4092, 44),
(6002, 4100, 36),
(6002, 4110, 39),
(6002, 4120, 47),
(6002, 4122, 41),
(6002, 4140, 46),
(6002, 4163, 40),
(6002, 4180, 38),
(6002, 4200, 48),
(6002, 4201, 38),
(6003, 4002, 25),
(6003, 4010, 27),
(6003, 4060, 21),
(6003, 4061, 38),
(6003, 4070, 25),
(6003, 4071, 35),
(6003, 4080, 40),
(6003, 4091, 28),
(6003, 4092, 35),
(6003, 4100, 32),
(6003, 4102, 31),
(6003, 4121, 36),
(6003, 4122, 40),
(6003, 4130, 28),
(6004, 4000, 27),
(6004, 4092, 22),
(6004, 4112, 34),
(6004, 4131, 31),
(6004, 4150, 33),
(6004, 4151, 23),
(6004, 4160, 34),
(6004, 4163, 28),
(6004, 4171, 31),
(6004, 4190, 35),
(6004, 4400, 40);

--
-- Triggers `contains`
--
DELIMITER $$
CREATE TRIGGER `Item_Add` BEFORE INSERT ON `contains` FOR EACH ROW BEGIN
    DECLARE error_msg VARCHAR(255);
    SET error_msg = ('The new quantity cannot be greater than 50');
    IF new.Quantity > 50 THEN
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = error_msg;
    END IF;
END
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `customers`
--

CREATE TABLE `customers` (
  `C_ID` int(11) NOT NULL,
  `First_Name` varchar(50) DEFAULT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Qualification` varchar(20) DEFAULT NULL,
  `ADDRESS` varchar(50) DEFAULT NULL,
  `Locality` varchar(20) DEFAULT NULL,
  `CITY` varchar(20) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL,
  `Phone_NO` varchar(10) DEFAULT NULL,
  `DOP` date NOT NULL,
  `E_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `customers`
--

INSERT INTO `customers` (`C_ID`, `First_Name`, `Last_Name`, `Qualification`,
`ADDRESS`, `Locality`, `CITY`, `Email`, `Phone_NO`, `DOP`, `E_ID`) VALUES
(2001, 'Sujatha', 'Mohan', 'Doctor', 'Vijay Apartment', 'Bilekahalli',
'Bangalore', 'Sujatha2022  @gmail.com', '1234567890', '2022-09-12', 1002),
(2002, 'Jhanavhi', '', 'Teacher', 'Jahnavi Enclave', 'Begur', 'Bangalore',
'Jhanavhi2022  @gmail.com', '9876543210', '2022-10-16', 1013),
(2003, 'Mohan ', 'Raj', 'Engineer', 'Vishwas Apartments', 'Hongasandra',
'Bangalore', 'Mohan 2022  @gmail.com', '8654432318', '2022-10-25', 1001),
(2004, 'Adarsh', 'Liju', 'Software Engineer', 'Vashist Apartments',
'Gottigere', 'Bangalore', 'Adarsh2022  @gmail.com', '8763313499', '2022-08-
22', 1003),
(2005, 'Vignesh', 'Sheshadri', 'Teacher', 'Shuddha Shelters', 'Arikere',
'Bangalore', 'Vignesh2022  @gmail.com', '2454676898', '2022-10-09', 1005),
(2006, 'Harsh ', 'Chowdhary', 'Doctor', 'Vishwas Apartments', 'Hulimavu',
'Bangalore', 'Harsh 2022  @gmail.com', '7364837638', '2022-10-07', 1006),
(2007, 'Rohith', 'Jain', 'Software Engineer', 'Hareetas', 'Hongasandra',
'Bangalore', 'Rohith2022  @gmail.com', '5365384899', '2022-08-21', 1007),
```

```
(2008, 'Himanshu', '', 'Engineer', 'Nandana Greens', 'Bilekahalli',
'Bangalore', 'Himanshu2022  @gmail.com', '7259966769', '2022-07-19', 1013),
(2009, 'Sutharsan', 'Raj', 'Student', 'Vijay apartment', 'Begur', 'Bangalore',
'Sutharsan2022  @gmail.com', '8396364290', '2022-08-31', 1004),
(2010, 'Kavi', 'Priya', 'Teacher', 'Vishwas Apartments', 'Hongasandra',
'Bangalore', 'Kavi2022  @gmail.com', '7678347343', '2022-08-02', 1013),
(2011, 'Varuna', 'Shree', 'Student', 'Hasmitha Nandana', 'Gottigere',
'Bangalore', 'Varuna2022  @gmail.com', '2646747346', '2022-10-07', 1013),
(2012, 'Gopinath', 'Gokul', 'Doctor', 'Prestige Song of South', 'Arekere',
'Bangalore', 'Gopinath2022  @gmail.com', '8464987736', '2022-10-17', 1009),
(2013, 'Krishna', 'Kumar', 'Student', 'Uday Apartments', 'Arekere',
'Bangalore', 'Krishna2022  @gmail.com', '4567893210', '2022-09-18', 1011),
(2014, 'Divya', 'Shree', 'Doctor', 'Prestige Song of South', 'Hulimavu',
'Bangalore', 'Divya2022  @gmail.com', '8664313546', '2022-08-29', 1012),
(2015, 'Siddharth', 'Seetharaman', 'Engineer', 'Pride Apartments', 'Hulimavu',
'Bangalore', 'Siddharth2022  @gmail.com', '8765432345', '2022-10-30', 1005),
(2016, 'Gokul', 'Sreenath', 'Student', 'Hasmitha Nandana', 'Begur',
'Bangalore', 'Gokul2022  @gmail.com', '6432469890', '2022-08-07', 1013),
(2017, 'Ramesh', 'Agarwal', 'Activist', 'Anugraha', 'Hongasandra',
'Bangalore', 'Ramesh2022  @gmail.com', '8632145805', '2022-09-23', 1004),
(2018, 'Suresh', 'Sathish', 'Teacher', 'Phoenix One', 'Bilekahalli',
'Bangalore', 'Suresh2022  @gmail.com', '7332668789', '2022-10-27', 1001),
(2019, 'Om', 'Katkam', 'Driver', 'Brigade Millenium', 'Gottigere',
'Bangalore', 'Om2022  @gmail.com', '1298765235', '2022-09-26', 1003),
(2020, 'Shashank', 'Singh', 'Driver', 'Pride Apartments', 'Arekere',
'Bangalore', 'Shashank2022  @gmail.com', '9876542345', '2022-08-14', 1006),
(2021, 'Rama', 'Krishna', 'Musician', 'Dwaraka Nilaya', 'Girinagar',
'Bangalore', 'ramakrishna2002@gmail.com', '8907564321', '2022-11-23', 1004),
(2022, 'Jenny', 'Meow', 'Pilot', 'Jenny Enclave', 'Buckingham Palace',
'Paris', 'jenny2022@gmail.com', '7259907510', '2022-11-19', 1007);

--
-- Triggers `customers`
--
DELIMITER $$
CREATE TRIGGER `Before_Delete_CustomerInfo` BEFORE DELETE ON `customers` FOR
EACH ROW BEGIN
CALL Delete_Customer(OLD.C_ID);
END
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `customer_backuplog`
--
```

```sql
CREATE TABLE `customer_backuplog` (
  `C_ID` int(11) NOT NULL,
  `First_Name` varchar(50) DEFAULT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Qualification` varchar(20) DEFAULT NULL,
  `ADDRESS` varchar(50) DEFAULT NULL,
  `Locality` varchar(20) DEFAULT NULL,
  `CITY` varchar(20) DEFAULT NULL,
  `Email` varchar(50) DEFAULT NULL,
  `Phone_NO` varchar(10) DEFAULT NULL,
  `DOP` date DEFAULT NULL,
  `E_ID` int(11) DEFAULT NULL,
  `Store_ID` int(11) DEFAULT NULL,
  `Item_ID` int(11) DEFAULT NULL,
  `Quantity` int(11) DEFAULT NULL,
  `O_Amount` float DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--
-- Table structure for table `employee`
--

CREATE TABLE `employee` (
  `E_ID` int(11) NOT NULL,
  `First_Name` varchar(30) DEFAULT NULL,
  `Last_Name` varchar(30) NOT NULL,
  `MGR_ID` int(11) DEFAULT NULL,
  `GENDER` varchar(1) NOT NULL,
  `SALARY` float NOT NULL,
  `DOB` date NOT NULL,
  `Address` varchar(50) DEFAULT NULL,
  `Phone_no` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `employee`
--

INSERT INTO `employee` (`E_ID`, `First_Name`, `Last_Name`, `MGR_ID`, `GENDER`,
`SALARY`, `DOB`, `Address`, `Phone_no`) VALUES
(1001, 'Akshay', 'Kumar', 1010, 'M', 69709, '1980-01-23', 'MG Road',
'8596092928'),
(1002, 'Akarsh', 'Poojari', 0, 'M', 112526, '1970-06-12', 'Basavangudi',
'7152011668'),
(1003, 'Bhanu', 'Preetham', 1002, 'M', 77901, '1990-03-15', 'Banashankari',
'7479776634'),
```

```
(1004, 'Bhavya ', 'Shree', 1007, 'F', 67891, '1984-10-17', 'RR Nagar',
'3391172715'),
(1005, 'Milan', '', 1002, 'M', 79103, '1987-09-27', 'Basavangudi',
'5605646750'),
(1006, 'Meenakshi', 'Saravanan', 1007, 'F', 59795, '1986-10-08',
'Banashankari', '2015362539'),
(1007, 'Ramya', 'Pandian', 0, 'F', 115297, '1971-02-19', 'Banashankari',
'9525847821'),
(1008, 'Nisha ', 'Advani', 1007, 'F', 55859, '1989-02-28', 'Basavangudi',
'5884750721'),
(1009, 'Surendra', 'Jain', 1013, 'M', 63712, '1992-03-14', 'RR Nagar',
'5002946135'),
(1010, 'Shina', 'Sudhir', 0, 'F', 109979, '1974-11-23', 'RR Nagar',
'8954874497'),
(1011, 'Tissa', 'Varghese', 1013, 'F', 75312, '1988-12-04', 'MG Road',
'7654873190'),
(1012, 'Navaneeth', 'Purohit', 1010, 'M', 60593, '1993-12-07', 'MG Road',
'1480171904'),
(1013, 'Yuvraj', 'Singh', 0, 'M', 118701, '1975-05-23', 'Banashankari',
'3791768076'),
(1014, 'Meow', 'Jenny', 1013, 'F', 200000, '2000-03-31', 'Meow Enclave',
'9876543210'),
(1015, 'Lithik', 'Raj', 1010, 'M', 80000, '1990-06-12', 'Netaji street',
'7689045321');

-- --------------------------------------------------------

--
-- Table structure for table `items`
--

CREATE TABLE `items` (
  `ITEM_ID` int(11) NOT NULL,
  `Item_Name` varchar(30) DEFAULT NULL,
  `PRICE` float DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `items`
--

INSERT INTO `items` (`ITEM_ID`, `Item_Name`, `PRICE`) VALUES
(4000, 'Shirt', 900),
(4001, 'Shirt', 1499),
(4002, 'Shirt', 999),
(4010, 'Pant', 2999),
(4011, 'Pant', 4000),
(4030, 'Coat', 15000),
```

```
(4041, 'Gown', 7999),
(4042, 'Gown', 4999),
(4050, 'Cap/Hat', 499),
(4060, 'Sweater', 999),
(4061, 'Sweater', 1799),
(4070, 'Jacket', 900),
(4071, 'Jacket', 600),
(4080, 'Legging', 400),
(4081, 'Legging', 349),
(4082, 'Legging', 499),
(4083, 'Legging', 550),
(4090, 'Jeggings', 700),
(4091, 'Jeggings', 599),
(4092, 'Jeggings', 445),
(4100, 'Tops', 250),
(4101, 'Tops', 399),
(4102, 'Tops', 549),
(4110, 'Saree', 1749),
(4111, 'Saree', 2499),
(4112, 'Saree', 3999),
(4120, 'Chudidhar', 799),
(4121, 'Chudidhar', 999),
(4122, 'Chudidhar', 1599),
(4130, 'Frock', 1199),
(4131, 'Frock', 1499),
(4140, 'Lehenga', 3000),
(4150, 'Dhoti', 400),
(4151, 'Dhoti', 500),
(4160, 'Tshirt', 799),
(4161, 'Tshirt', 699),
(4162, 'Tshirt', 1000),
(4163, 'Tshirt', 900),
(4170, 'Shorts', 500),
(4171, 'Shorts', 599),
(4180, 'Skirt', 699),
(4181, 'Skirt', 700),
(4190, 'Pyjama', 650),
(4200, 'Kurta', 999),
(4201, 'Kurta', 799),
(4300, 'Palazzo', 749),
(4400, 'Cigar Pant', 1999);

-- --------------------------------------------------------

--
-- Table structure for table `item_category`
--
```

```sql
CREATE TABLE `item_category` (
  `ITEM_ID` int(11) NOT NULL,
  `Item_Name` varchar(30) NOT NULL,
  `Gender` varchar(1) NOT NULL,
  `BRAND` varchar(20) DEFAULT 'SMS Textiles',
  `COLOUR` varchar(20) DEFAULT NULL,
  `SIZE` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `item_category`
--

INSERT INTO `item_category` (`ITEM_ID`, `Item_Name`, `Gender`, `BRAND`,
`COLOUR`, `SIZE`) VALUES
(4000, 'Shirt', 'M', 'Ramraj', 'Red', '38'),
(4001, 'Shirt', 'M', 'Allen Solly', 'Blue', '40'),
(4002, 'Shirt', 'M', 'Peterson', 'Green', '42'),
(4010, 'Pant', 'M', 'Buffallo', 'Black', '38'),
(4011, 'Pant', 'M', 'Polo', 'Brown', '40'),
(4030, 'Coat', 'F', 'Raymond', 'Grey', 'XXL'),
(4041, 'Gown', 'F', 'Chennai Silks', 'Pink', 'XXL'),
(4042, 'Gown', 'M', 'ARRS Silks', 'Green', 'XL'),
(4050, 'Cap/Hat', 'M', 'MAX', 'Cyan', ''),
(4060, 'Sweater', 'F', 'Deccathlon', 'Navyblue', 'XL'),
(4061, 'Sweater', 'M', 'Deccathlon', 'Light_Brown', 'L'),
(4070, 'Jacket', 'M', 'Kumaran Tex', 'Red', 'L'),
(4071, 'Jacket', 'M', 'Vittal Dresses', 'Brown', 'XXL'),
(4080, 'Legging', 'F', 'Chennai Silks', 'Maroon', 'XL'),
(4081, 'Legging', 'F', 'ARRS Silks', 'White', 'M'),
(4082, 'Legging', 'F', 'Kumaran Tex', 'Black', 'XXL'),
(4083, 'Legging', 'F', 'Pothys', 'Dark_Blue', 'XXXL'),
(4090, 'Jeggings', 'F', 'Chennai Silks', 'Purple', 'XXL'),
(4091, 'Jeggings', 'F', 'ARRS Silks', 'Grey', 'XL'),
(4092, 'Jeggings', 'F', 'Kumaran Tex', 'Yellow', 'L'),
(4100, 'Tops', 'F', 'Pothys', 'Black', 'XL'),
(4101, 'Tops', 'F', 'Trends', 'Red', 'XL'),
(4102, 'Tops', 'F', 'Trends', 'Blue', 'XXL'),
(4110, 'Saree', 'F', 'Chennai Silks', 'Pink', ''),
(4111, 'Saree', 'F', 'Kanchipuram Textiles', 'Purple', ''),
(4112, 'Saree', 'F', 'Kanchipuram Textiles', 'Blue', ''),
(4120, 'Chudidhar', 'F', 'Chennai Silks', 'Yellow', 'L'),
(4121, 'Chudidhar', 'F', 'ARRS Silks', 'Orange', 'XL'),
(4122, 'Chudidhar', 'F', 'Pothys', 'Green', 'XXL'),
(4130, 'Frock', 'F', 'Ramraj', 'Rainbow', '38'),
(4131, 'Frock', 'F', 'MAX', 'Violet', '42'),
(4140, 'Lehenga', 'F', 'Raymond', 'Baby_Pink', 'XL'),
(4150, 'Dhoti', 'M', 'Ramraj', 'White', '40'),
```

```
(4151, 'Dhoti', 'M', 'Pothys', 'Light_Brown', '50'),
(4160, 'Tshirt', 'M', 'Van Hussen', 'Red', '42'),
(4161, 'Tshirt', 'M', 'Van Hussen', 'Black', '44'),
(4162, 'Tshirt', 'M', 'MAX', 'Brown', '40'),
(4163, 'Tshirt', 'M', 'Polo', 'Grey', '38'),
(4170, 'Shorts', 'M', 'Polo', 'Green', '36'),
(4171, 'Shorts', 'M', 'Buffallo', 'Blue', '38'),
(4180, 'Skirt', 'F', 'Kanchipuram Tex', 'Light_Green', '42'),
(4181, 'Skirt', 'F', 'Kumaran Tex', 'Pink', '38'),
(4190, 'Pyjama', 'M', 'Chennai Silks', 'Grey', '40'),
(4200, 'Kurta', 'M', 'Raymond', 'White', '42'),
(4201, 'Kurta', 'M', 'Trends', 'Red', '36'),
(4300, 'Palazzo', 'F', 'Prisma', 'Cyan', 'XXL'),
(4400, 'Cigar Pant', 'F', 'Twinbirds', 'Maroon', 'XL');

-- --------------------------------------------------------

--
-- Table structure for table `orders`
--

CREATE TABLE `orders` (
  `C_ID` int(11) NOT NULL,
  `ITEM_ID` int(11) NOT NULL,
  `Price` float DEFAULT NULL,
  `QUANTITY` int(5) NOT NULL,
  `O_Date` date DEFAULT NULL,
  `O_Amount` float DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `orders`
--

INSERT INTO `orders` (`C_ID`, `ITEM_ID`, `Price`, `QUANTITY`, `O_Date`,
`O_Amount`) VALUES
(2001, 4002, 999, 3, '2022-12-09', 2997),
(2001, 4011, 4000, 4, '2022-12-09', 16000),
(2001, 4090, 700, 10, '2022-12-09', 7000),
(2002, 4002, 999, 2, '2022-10-16', 2000),
(2002, 4080, 400, 4, '2022-08-16', 1600),
(2002, 4102, 549, 5, '2022-10-16', 2750),
(2002, 4121, 999, 2, '2022-10-16', 1998),
(2003, 4000, 900, 10, '2022-10-25', 8000),
(2003, 4090, 700, 7, '2022-10-25', 4900),
(2003, 4151, 500, 5, '2022-10-25', 2500),
(2003, 4171, 599, 9, '2022-10-25', 5400),
(2004, 4082, 499, 12, '2022-08-22', 6000),
```

```
(2004, 4170, 500, 7, '2022-08-22', 3500),
(2005, 4002, 999, 1, '2022-09-10', 1000),
(2005, 4081, 349, 5, '2022-09-10', 1750),
(2005, 4082, 499, 3, '2022-09-10', 1500),
(2006, 4120, 799, 7, '2022-07-10', 5600),
(2006, 4140, 3000, 3, '2022-07-10', 9000),
(2006, 4201, 799, 8, '2022-01-08', 7992),
(2007, 4042, 4999, 1, '2022-08-21', 5000),
(2007, 4122, 1599, 3, '2022-08-21', 4797),
(2007, 4140, 3000, 7, '2022-08-21', 21000),
(2008, 4002, 999, 3, '2022-07-19', 3000),
(2008, 4010, 2999, 6, '2022-07-19', 18000),
(2008, 4061, 1799, 4, '2022-07-19', 7200),
(2009, 4083, 550, 6, '2022-08-31', 3300),
(2009, 4200, 999, 4, '2022-08-31', 3200),
(2010, 4071, 600, 4, '2022-02-08', 2400),
(2010, 4092, 445, 7, '2022-02-08', 3150),
(2011, 4122, 1599, 3, '2022-07-10', 4797),
(2011, 4130, 1199, 7, '2022-07-10', 8393),
(2012, 4060, 999, 1, '2022-10-17', 1000),
(2012, 4061, 1799, 3, '2022-10-17', 5400),
(2012, 4091, 599, 6, '2022-10-17', 3600),
(2012, 4121, 999, 5, '2022-10-17', 4995),
(2013, 4070, 900, 1, '2022-09-18', 900),
(2013, 4102, 549, 3, '2022-09-18', 1650),
(2014, 4160, 799, 8, '2022-08-29', 6400),
(2015, 4002, 999, 5, '2022-10-30', 5000),
(2015, 4011, 4000, 2, '2022-10-30', 8000),
(2015, 4111, 2499, 3, '2022-10-30', 7500),
(2016, 4030, 15000, 1, '2022-07-08', 15000),
(2016, 4061, 1799, 3, '2022-07-08', 3600),
(2016, 4180, 699, 5, '2022-07-08', 3500),
(2016, 4200, 999, 9, '2022-07-08', 7200),
(2017, 4030, 15000, 2, '2022-09-23', 30000),
(2017, 4100, 250, 9, '2022-09-23', 2250),
(2018, 4131, 1499, 4, '2022-10-27', 5996),
(2018, 4160, 799, 5, '2022-10-27', 4000),
(2018, 4190, 650, 11, '2022-10-27', 4400),
(2019, 4041, 7999, 2, '2022-09-26', 16000),
(2019, 4050, 499, 3, '2022-09-26', 1500),
(2019, 4090, 700, 3, '2022-09-26', 2100),
(2020, 4100, 250, 15, '2022-08-14', 3750),
(2020, 4110, 1749, 9, '2022-08-14', 15750),
(2020, 4163, 900, 6, '2022-08-14', 5400),
(2021, 4030, 15000, 1, '2022-11-23', 15000),
(2021, 4061, 1799, 1, '2022-11-23', 1799),
(2021, 4161, 699, 2, '2022-11-23', 1398),
(2022, 4080, 400, 4, '2022-11-19', 1600),
```

```
(2022, 4121, 999, 3, '2022-11-20', 2997),
(2022, 4140, 3000, 2, '2022-11-19', 6000);

-- --------------------------------------------------------

--
-- Table structure for table `shipment`
--

CREATE TABLE `shipment` (
  `SHIP_ID` int(11) NOT NULL,
  `DATE_OF_SHIPMENT` date DEFAULT NULL,
  `STORE_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `shipment`
--

INSERT INTO `shipment` (`SHIP_ID`, `DATE_OF_SHIPMENT`, `STORE_ID`) VALUES
(8001, '2022-05-22', 6001),
(8002, '2022-06-30', 6001),
(8003, '2022-09-05', 6001),
(8004, '2022-12-05', 6002),
(8005, '2022-04-21', 6004),
(8006, '2022-06-06', 6003),
(8007, '2022-06-25', 6004),
(8008, '2022-06-16', 6002),
(8009, '2022-05-31', 6003),
(8010, '2022-04-27', 6004);

-- --------------------------------------------------------

--
-- Table structure for table `ships`
--

CREATE TABLE `ships` (
  `COST_OF_SHIPPING` float DEFAULT NULL,
  `MODE_OF_TRAVELLING` varchar(255) DEFAULT NULL,
  `SUPP_ID` int(11) NOT NULL,
  `SHIP_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `ships`
--
```

```sql
INSERT INTO `ships` (`COST_OF_SHIPPING`, `MODE_OF_TRAVELLING`, `SUPP_ID`,
`SHIP_ID`) VALUES
(5489, 'Roadways', 7001, 8005),
(4000, 'Airways', 7001, 8006),
(6597, 'Railways', 7002, 8001),
(870, 'Railways', 7002, 8009),
(5500, 'Railways', 7003, 8004),
(5500, 'Railways', 7003, 8008),
(7000, 'Airways', 7004, 8003),
(3500, 'Roadways', 7004, 8007),
(6290, 'Roadways', 7005, 8002),
(6290, 'Waterways', 7005, 8010);

-- --------------------------------------------------------

--
-- Table structure for table `store`
--

CREATE TABLE `store` (
  `STORE_ID` int(11) NOT NULL,
  `NAME` varchar(50) DEFAULT NULL,
  `ADDRESS` varchar(50) DEFAULT NULL,
  `MGR_ID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `store`
--

INSERT INTO `store` (`STORE_ID`, `NAME`, `ADDRESS`, `MGR_ID`) VALUES
(6001, 'Bannerghatta', 'BG Road', 1002),
(6002, 'Jayanagar', 'Near Cool Joint', 1007),
(6003, 'Rajajinagar', 'Opposite to Rajarajeshwari Medical College', 1013),
(6004, 'Malleshwaram', 'Near Railway Station', 1010);

-- --------------------------------------------------------

--
-- Table structure for table `suppliers`
--

CREATE TABLE `suppliers` (
  `SUPP_ID` int(11) NOT NULL,
  `NAME` varchar(50) DEFAULT NULL,
  `ADDRESS` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```sql
--
-- Dumping data for table `suppliers`
--

INSERT INTO `suppliers` (`SUPP_ID`, `NAME`, `ADDRESS`) VALUES
(7001, 'Sukeerthan', 'Haritasa Apartments'),
(7002, 'Monisha', 'Mahaveer Marvel'),
(7003, 'Kavana', 'Tvs Emarold jordin'),
(7004, 'Roshan', 'Thayappa Garden'),
(7005, 'Satvik', 'Bhavani Apartments');

-- --------------------------------------------------------

--
-- Stand-in structure for view `total_orders`
-- (See below for the actual view)
--
CREATE TABLE `total_orders` (
`Item_ID` int(11)
,`C_ID` int(11)
,`Price` float
,`Quantity` int(5)
,`O_Amount` double
);
```

C)CONSTRAINTS OF PK AND FK:

```
ALTER TABLE `bill`
  ADD PRIMARY KEY (`B_ID`),
  ADD KEY `c_fid` (`C_ID`);

--
-- Indexes for table `contains`
--
ALTER TABLE `contains`
  ADD PRIMARY KEY (`STORE_ID`,`ITEM_ID`),
  ADD KEY `i_fid1` (`ITEM_ID`);

--
-- Indexes for table `customers`
--
ALTER TABLE `customers`
  ADD PRIMARY KEY (`C_ID`),
  ADD KEY `e_fid` (`E_ID`);

--
-- Indexes for table `employee`
--
ALTER TABLE `employee`
  ADD PRIMARY KEY (`E_ID`);

--
-- Indexes for table `items`
--
ALTER TABLE `items`
  ADD PRIMARY KEY (`ITEM_ID`);

--
-- Indexes for table `item_category`
--
ALTER TABLE `item_category`
  ADD PRIMARY KEY (`ITEM_ID`);

--
-- Indexes for table `orders`
--
ALTER TABLE `orders`
  ADD PRIMARY KEY (`C_ID`,`ITEM_ID`),
  ADD KEY `item_fid2` (`ITEM_ID`);

--
-- Indexes for table `shipment`
--
```

```sql
ALTER TABLE `shipment`
  ADD PRIMARY KEY (`SHIP_ID`),
  ADD KEY `store_fid` (`STORE_ID`);

--
-- Indexes for table `ships`
--
ALTER TABLE `ships`
  ADD PRIMARY KEY (`SUPP_ID`,`SHIP_ID`),
  ADD KEY `ship_fid` (`SHIP_ID`);

--
-- Indexes for table `store`
--
ALTER TABLE `store`
  ADD PRIMARY KEY (`STORE_ID`),
  ADD KEY `mgr_fid` (`MGR_ID`);

--
-- Indexes for table `suppliers`
--
ALTER TABLE `suppliers`
  ADD PRIMARY KEY (`SUPP_ID`);

--
-- Constraints for dumped tables
--


--
-- Constraints for table `bill`
--
ALTER TABLE `bill`
  ADD CONSTRAINT `c_fid` FOREIGN KEY (`C_ID`) REFERENCES `customers` (`C_ID`)
ON DELETE CASCADE;

--
-- Constraints for table `contains`
--
ALTER TABLE `contains`
  ADD CONSTRAINT `i_fid1` FOREIGN KEY (`ITEM_ID`) REFERENCES `items`
(`ITEM_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `store_fid1` FOREIGN KEY (`STORE_ID`) REFERENCES `store`
(`STORE_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `customers`
--
ALTER TABLE `customers`
```

```
  ADD CONSTRAINT `e_fid` FOREIGN KEY (`E_ID`) REFERENCES `employee` (`E_ID`)
ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `item_category`
--
ALTER TABLE `item_category`
  ADD CONSTRAINT `item_fid` FOREIGN KEY (`ITEM_ID`) REFERENCES `items`
(`ITEM_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `orders`
--
ALTER TABLE `orders`
  ADD CONSTRAINT `c_fid2` FOREIGN KEY (`C_ID`) REFERENCES `customers` (`C_ID`)
ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `item_fid2` FOREIGN KEY (`ITEM_ID`) REFERENCES `items`
(`ITEM_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `shipment`
--
ALTER TABLE `shipment`
  ADD CONSTRAINT `shipment_fid` FOREIGN KEY (`SHIP_ID`) REFERENCES `ships`
(`SHIP_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `store_fid` FOREIGN KEY (`STORE_ID`) REFERENCES `store`
(`STORE_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `ships`
--
ALTER TABLE `ships`
  ADD CONSTRAINT `supp_fid` FOREIGN KEY (`SUPP_ID`) REFERENCES `suppliers`
(`SUPP_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `store`
--
ALTER TABLE `store`
  ADD CONSTRAINT `mgr_fid` FOREIGN KEY (`MGR_ID`) REFERENCES `employee`
(`E_ID`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

D) Procedures and Functions:

i) Customer Deletion and Backup Procedure:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `Delete_Customer` (IN `C_ID_val`
INT)   BEGIN
DECLARE done INT DEFAULT 0;
DECLARE val int;
DECLARE `C_ID` int;
DECLARE `First_Name` varchar(50);
DECLARE `Last_Name` varchar(50);
DECLARE `Qualification` varchar(20);
DECLARE `ADDRESS` varchar(50);
DECLARE `Locality` varchar(20);
DECLARE `CITY` varchar(20);
DECLARE `Email` varchar(50);
DECLARE `Phone_NO` varchar(10);
DECLARE `DOP` date;
DECLARE `E_ID` int(11);
DECLARE `Item_ID` int;
DECLARE `Price` float;
DECLARE `Quantity` int;
DECLARE `O_Amount` float;

DECLARE cur CURSOR FOR SELECT
C.C_ID, C.First_Name, C.Last_Name, C.Qualification,
C.Address, C.Locality, C.City, C.Email, C.Phone_NO, C.DOP, C.E_ID,
O.Item_ID,O.Price, O.Quantity, O.O_Amount
FROM Customers AS C, Orders as O
WHERE O.C_ID = C_ID_val AND C.C_ID = C_ID_val;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;


OPEN cur;
LABEL:loop

FETCH cur INTO  C_ID, First_Name, Last_Name, Qualification, Address,
Locality, City, Email, Phone_NO, DOP, E_ID,
Item_ID, Price, Quantity, O_Amount;

INSERT INTO Customer_Backuplog VALUES(C_ID, First_Name, Last_Name,
Qualification, Address,
Locality, City, Email, Phone_NO, DOP, E_ID,
Item_ID, Price, Quantity, O_Amount);

IF done = 1 THEN LEAVE LABEL;
END IF;
END loop;
```

```
CLOSE cur;
SET val = (select count(*) from Customer_Backuplog where
Customer_Backuplog.C_ID = C_ID_val);
IF val > 0 THEN
    DELETE FROM Orders
    WHERE Orders.C_ID = C_ID_val;
END IF;
END$$
```

ii)Employee Age Calculation Procedure:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `Display_Age` (IN `UID` INT, OUT
`Age` INT)   BEGIN
    DECLARE Date_OF_Birth date;
    SELECT * FROM Employee WHERE E_ID = UID;
    SELECT E.DOB INTO Date_OF_Birth
    FROM Employee AS E
    WHERE E.E_ID = UID;
    SET Age = FLOOR(DATEDIFF(CURRENT_DATE, Date_OF_Birth)/365);
END$$
```

iii) Customer Loyalty Level Assessment Function:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `CustomerLevel` (`C_ID` INT)
RETURNS VARCHAR(20) CHARSET utf8mb4 DETERMINISTIC BEGIN
    DECLARE customerLevel VARCHAR(20);
    DECLARE credit float;
    SELECT sum(O.O_Amount) INTO credit
    FROM ORDERS AS O
    WHERE O.C_ID = C_ID
    GROUP BY O.C_ID;


    IF credit > 25000 THEN
    SET customerLevel = 'PLATINUM';
    ELSEIF (credit >= 21000 AND
      credit <= 25000) THEN
        SET customerLevel = 'GOLD';
    ELSEIF credit < 20000 THEN
        SET customerLevel = 'SILVER';
    ELSE
        SET customerLevel = 'BRONZE';
    END IF;
  -- Return the customer level
  RETURN (customerLevel);
```

```
END$$
```

iv) Customer Discount Calculation Function:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `Discount` (`C_ID` INT) RETURNS
FLOAT DETERMINISTIC BEGIN
    DECLARE credit float;
    DECLARE discount float;
    DECLARE final_amount float;
    SELECT Amount INTO credit
    FROM Bill AS B
    WHERE B.C_ID = C_ID;


    IF credit > 25000 THEN
    SET final_amount = credit - (credit * 0.2);
        SET discount = credit * 0.2;
    ELSEIF (credit > 10000 AND credit <= 25000) THEN
        SET final_amount = credit - (credit * 0.1);
        SET discount = credit * 0.1;
    ELSEIF (credit > 5000 AND credit <= 10000) THEN
        SET final_amount = credit - (credit * 0.08);
        SET discount = credit * 0.08;
    ELSE
        SET final_amount = credit - (credit * 0.05);
        SET discount = credit * 0.05;
    END IF;
  RETURN (discount);
END$$
```

E) Triggers:

```
--
-- Triggers `contains`
--
DELIMITER $$
CREATE TRIGGER `Item_Add` BEFORE INSERT ON `contains` FOR EACH ROW BEGIN
    DECLARE error_msg VARCHAR(255);
    SET error_msg = ('The new quantity cannot be greater than 50');
    IF new.Quantity > 50 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = error_msg;
    END IF;
END
```

```
$$
DELIMITER ;
```

```
--
-- Triggers `customers`
--
DELIMITER $$
CREATE TRIGGER `Before_Delete_CustomerInfo` BEFORE DELETE ON `customers` FOR
EACH ROW BEGIN
CALL Delete_Customer(OLD.C_ID);
END
$$
DELIMITER ;
```

FRONT-END:

In the front end , the OPTIONS refer to the menu as mentioned above namely

- CUSTOMER (Combines Customers, Orders, Bill tables)
- EMPLOYEE(Combines Employee table)
- ITEMS-STOCK(Combines Items ,Item_Category, Contains, Store tables)
- SUPPLIER-SHIP(Combines Shipment, Ships, Suppliers table)

In the front end , the ACTIONS refer to the CRUD operations namely

- ADD
- READ
- UPDATE
- REMOVE