LITTLE FLOWER SCHOOL DATABASE

1.1 INTRODUCTION :

Little Flower School is a nurturing educational institution dedicated to fostering academic excellence and personal growth in a supportive environment. Located in salem, the school provides a holistic education that blends rigorous academics with a focus on character development and extracurricular enrichment.

1.2 Little Flower School Database

The Little Flower School Database is designed to efficiently manage and organize the core aspects of school operations.

It encompasses various tables that collectively support the school's administrative and operational needs. The primary tables in the database are:

Student : Contains detailed records of students, including personal information, class assignments, and parent contact details.

This table is crucial for tracking student performance and attendance.

Employee : Stores information about school staff, including teachers, administrators, and support personnel.

This table manages employee details such as names, positions, salaries, and contact information.

Class Room: Manages data related to classrooms, including classroom IDs, in-charge details, class assignments, and capacity.

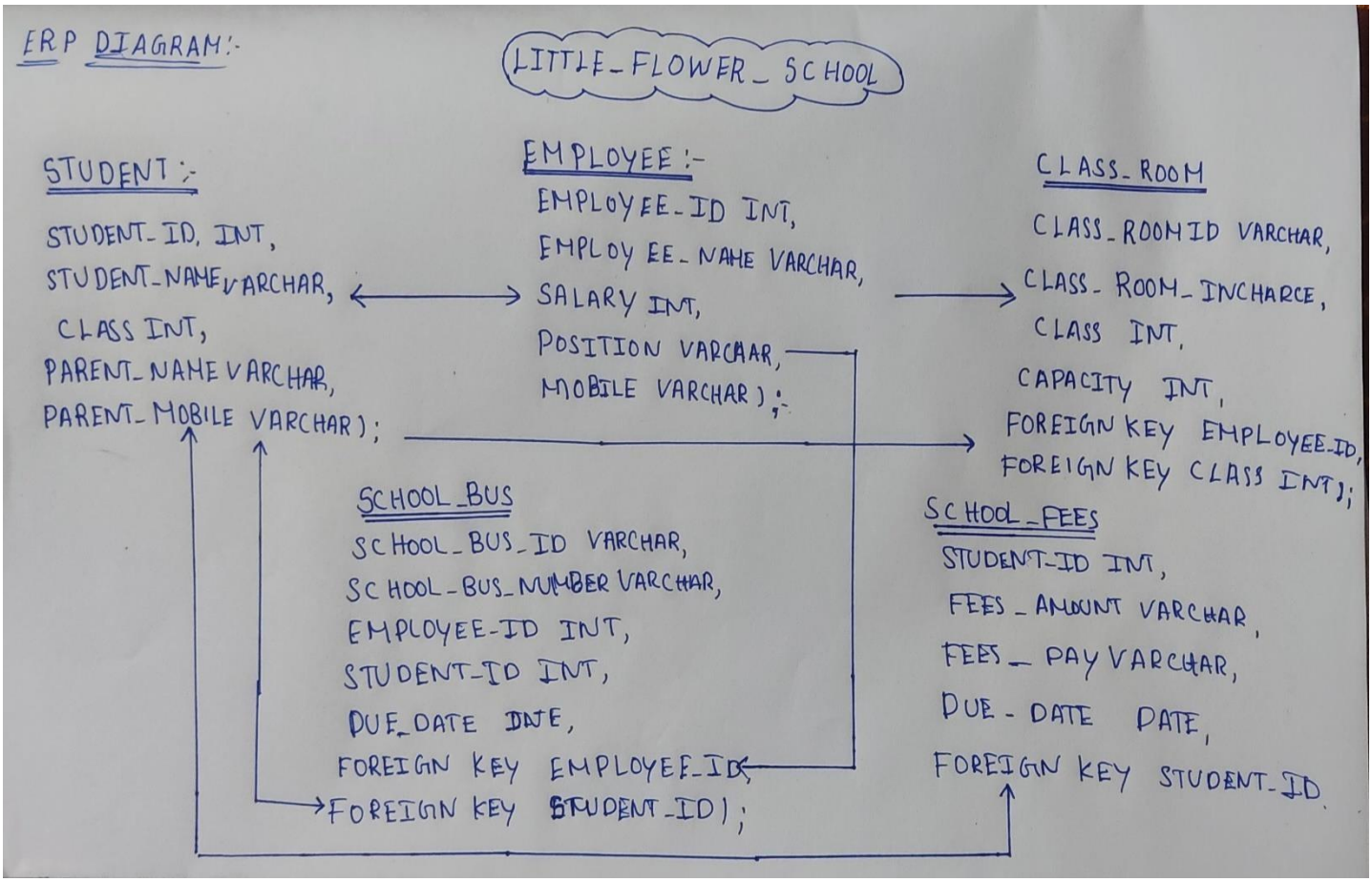
This table helps in organizing and assigning rooms for different classes and activities.

Student Fees: Records fee-related information for students, including fee amounts, payments made, and due dates. This table is essential for tracking financial transactions and managing fee collections.

School Bus: Maintains information about the school's transportation system, including bus IDs, numbers, routes, and the students assigned to each bus.

Each table is interlinked to provide a comprehensive view of the school's operations, supporting effective management and decision-making.

EER DIAGRAM :



1.3 CREATE DATABASE & TABLE

Create the data base :

```
CREATE DATABASE little_flower_school ;
```

Use the created data base :

```
USE little_flower_school ;
```

CREATE STUDENT TABLE ;

```
CREATE TABLE STUDENT (
   STUDENT_ID INT PRIMARY KEY,
   STUDENT_NAME VARCHAR(100),
   CLASS INT,
   PARENT_NAME VARCHAR(100),
   PARENT_MOBILE VARCHAR(100)
)
;
```

CREATE EMPLOYEE TABLE ;

```
CREATE TABLE Employees (
   EMPLOYEE_ID INT PRIMARY KEY,
   EMPLOYEE_NAME VARCHAR(100),
   POSITION VARCHAR(100),
   SALARY INT,
   SALARY_DATE DATE,
   EMPLOYEE_MOBILE VARCHAR(100)
)
;
```

CREATE CLASS ROOM ;

```
CREATE TABLE ClassRoom (
   Class_RoomID VARCHAR(100),
   ClassRoom_InchargeID INT,
   Class INT,
   ClassRoom_Capacity INT,
```

```
   FOREIGN KEY (ClassRoom_InchargeID) REFERENCES Employees(EMPLOYEE_ID),
FOREIGN KEY (Class) REFERENCES STUDENT(CLASS)
)
;
```

CREATE THE TABLE SCHOOL BUS ;

```
CREATE TABLE School_Bus (
   SCHOOL_BUS_ID VARCHAR(100),
   SCHOOL_BUS_NUMBER VARCHAR(100),
   EMPLOYEE_ID INT,
   STUDENT_ID INT,
   ROUTE VARCHAR(100),
   FOREIGN KEY (EMPLOYEE_ID) REFERENCES Employees(EMPLOYEE_ID),
   FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT(STUDENT_ID)
)
;
```

CRATE STUDENT FEES ;

```
CREATE TABLE student_fees (
   STUDENT_ID INT,
   FEES_AMOUNT VARCHAR(100),
   FEES_PAY VARCHAR(50),
   DUE_DATE DATE,
   FOREIGN KEY (STUDENT_ID) REFERENCES STUDENT (STUDENT_ID)
);
```

1.4 INSERT INTO VALUES ;

INSERT DATA STUDENT TABLE

```
INSERT INTO  STUDENT (STUDENT_ID, STUDENT_NAME, CLASS, PARENT_NAME,

PARENT_MOBILE)

VALUES

(1001, 'Aiden', 1, 'John Doe', '9712345678'),
```

(1002, 'Emma', 1, 'Jane Doe', '9723456789'),

(1003, 'Noah', 1, 'Michael Smith', '9734567890'),

(1004, 'Olivia', 1, 'Emily Smith', '9745678901'),

(1005, 'Liam', 1, 'Robert Johnson', '9756789012'),

(1006, 'Ava', 1, 'Linda Johnson', '9767890123'),

(1007, 'Mason', 1, 'James Williams', '9778901234'),

(1008, 'Sophia', 1, 'Patricia Williams', '9789012345'),

(1009, 'Jacob', 1, 'Daniel Brown', '9790123456'),

(1010, 'Isabella', 1, 'Nancy Brown', '9701234567'),

(1011, 'Ethan', 1, 'Thomas Jones', '9712345679'),

(1012, 'Mia', 1, 'Margaret Jones', '9723456790'),

(1013, 'Michael', 1, 'Charles Garcia', '9734567901'),

(1014, 'Charlotte', 1, 'Barbara Garcia', '9745679012'),

(1015, 'Alexander', 1, 'Christopher Martinez', '9756789123'),

(1016, 'Amelia', 1, 'Jessica Martinez', '9767890234'),

(1017, 'William', 1, 'David Hernandez', '9778901345'),

(1018, 'Harper', 1, 'Sarah Hernandez', '9789012456'),

(1019, 'James', 1, 'Joseph Wilson', '9790123567'),

(1020, 'Ella', 1, 'Karen Wilson', '9701234678') ,

(2001, 'Arun', 2, 'Rajesh Kumar', '9712345678'),

(2002, 'Priya', 2, 'Sangeetha Kumar', '9723456789'),

(2003, 'Karthik', 2, 'Venkatesh Reddy', '9734567890'),

(2004, 'Anitha', 2, 'Lakshmi Reddy', '9745678901'),

(2005, 'Suresh', 2, 'Mani Subramanian', '9756789012'),

(2006, 'Jaya', 2, 'Meena Subramanian', '9767890123'),

(2007, 'Ravi', 2, 'Rajendran Narayanan', '9778901234'),

(2008, 'Sita', 2, 'Sujatha Narayanan', '9789012345'),

(2009, 'Vijay', 2, 'Gopalakrishnan Iyer', '9790123456'),

(2010, 'Kaviya', 2, 'Saraswathi Iyer', '9701234567'),

(2011, 'Selvan', 2, 'Murugan Pillai', '9712345679'),

(2012, 'Madhavi', 2, 'Nandini Pillai', '9723456790'),

(2013, 'Ganesh', 2, 'Arumugam Srinivasan', '9734567901'),

(2014, 'Lakshmi', 2, 'Rathnam Srinivasan', '9745679012'),

(2015, 'Sanjay', 2, 'Vijayalakshmi Ravi', '9756789123'),

(2016, 'Nila', 2, 'Rajkumar Ravi', '9767890234'),

(2017, 'Manoj', 2, 'Karthikeyan Balasubramanian', '9778901345'),

(2018, 'Anjali', 2, 'Geetha Balasubramanian', '9789012456'),

(2019, 'Arvind', 2, 'Kumarasamy Chandran', '9790123567'),

(2020, 'Divya', 2, 'Rukmini Chandran', '9701234678'),

(3001, 'Indhu', 3, 'Rajendran Kumar', '9712345678'),

(3002, 'Musilim', 3, 'Lakshmi Kumar', '9723456789'),

(3003, 'Cristhaan', 3, 'Suresh Reddy', '9734567890'),

(3004, 'Aarav', 3, 'Anitha Reddy', '9745678901'),

(3005, 'Meera', 3, 'Venkatesh Nair', '9756789012'),

(3006, 'Ravi', 3, 'Geetha Nair', '9767890123'),

(3007, 'Nandini', 3, 'Murugan Subramanian', '9778901234'),

(3008, 'Kumar', 3, 'Vijayalakshmi Subramanian', '9789012345'),

(3009, 'Divya', 3, 'Ganesh Iyer', '9790123456'),

(3010, 'Arjun', 3, 'Lakshmi Iyer', '9701234567'),

(3011, 'Kavya', 3, 'Ramesh Pillai', '9712345679'),

(3012, 'Sanjay', 3, 'Saraswathi Pillai', '9723456790'),

(3013, 'Harini', 3, 'Rajkumar Srinivasan', '9734567901'),

(3014, 'Vikram', 3, 'Meena Srinivasan', '9745679012'),

(3015, 'Sita', 3, 'Karthikeyan Ravi', '9756789123'),

(3016, 'Ranjith', 3, 'Geetha Ravi', '9767890234'),

(3017, 'Ananya', 3, 'Rajendran Balasubramanian', '9778901345'),

(3018, 'Ravi', 3, 'Nandhini Balasubramanian', '9789012456'),

(3019, 'Karthik', 3, 'Arumugam Chandran', '9790123567'),

(3020, 'Lakshmi', 3, 'Sita Chandran', '9701234678');

(4001, 'Arun', 4, 'Ramesh Kumar', '9112345678'),

(4002, 'Priya', 4, 'Asha Kumar', '9123456789'),

(4003, 'Karthik', 4, 'Rajesh Iyer', '9134567890'),

(4004, 'Anitha', 4, 'Sanjay Iyer', '9145678901'),

(4005, 'Vijay', 4, 'Mohammed Ali', '9156789012'),

(4006, 'Sita', 4, 'Fathima Ali', '9167890123'),

(4007, 'Ganesh', 4, 'Suresh Patel', '9178901234'),

(4008, 'Lakshmi', 4, 'Meena Patel', '9189012345'),

(4009, 'Nila', 4, 'John Peter', '9190123456'),

(4010, 'Ravi', 4, 'Mary Peter', '9101234567'),

(4011, 'Aarav', 4, 'Venkatesh Rajan', '9112345679'),

(4012, 'Meera', 4, 'Lakshmi Rajan', '9123456790'),

(4013, 'Kaviya', 4, 'Arumugam Subramanian', '9134567901'),

(4014, 'Sanjay', 4, 'Nandini Subramanian', '9145679012'),

(4015, 'Harini', 4, 'Gurpreet Singh', '9156789123'),

(4016, 'Vikram', 4, 'Aarti Singh', '9167890234'),

(4017, 'Divya', 4, 'Rajendran Murugan', '9178901345'),

(4018, 'Sanjana', 4, 'Kavitha Murugan', '9189012456'),

(4019, 'Karthik', 4, 'Suresh Kumar', '9190123567'),

(4020, 'Lakshmi', 4, 'Rukmini Kumar', '9101234678');

(5001, 'Gokul', 5, 'Ravi Kumar', '9112345678'),

(5002, 'Vimal', 5, 'Sita Kumar', '9123456789'),

(5003, 'Muthu', 5, 'Suresh Nair', '9134567890'),

(5004, 'Preetha', 5, 'Lakshmi Nair', '9145678901'),

(5005, 'Anamiga', 5, 'Ramesh Raj', '9156789012'),

(5006, 'David', 5, 'Anitha Raj', '9167890123'),

(5007, 'Ram', 5, 'Vijay Kumar', '9178901234'),

(5008, 'Meena', 5, 'Kavitha Kumar', '9189012345'),

(5009, 'Karthik', 5, 'Arumugam Pillai', '9190123456'),

(5010, 'Nila', 5, 'Rani Pillai', '9101234567'),

(5011, 'Raj', 5, 'Ganesh Subramanian', '9112345679'),

(5012, 'Sanjana', 5, 'Sangeetha Subramanian', '9123456790'),

(5013, 'Harini', 5, 'Ravi Iyer', '9134567901'),

(5014, 'Suresh', 5, 'Madhavi Iyer', '9145679012'),

(5015, 'Kavya', 5, 'Kumaravelu Reddy', '9156789123'),

(5016, 'Anjali', 5, 'Lakshmi Reddy', '9167890234'),

(5017, 'Ganesh', 5, 'Rajesh Ramasamy', '9178901345'),

(5018, 'Lakshmi', 5, 'Jaya Ramasamy', '9189012456'),

(5019, 'Ravi', 5, 'Shankar Balasubramanian', '9190123567'),

(5020, 'Divya', 5, 'Priya Balasubramanian', '9101234678');

-- INSERT RECORDS INTO THE EMPLOYEE TABLE

INSERT INTO Employees (EMPLOYEE_ID, EMPLOYEE_NAME, POSITION, SALARY,

SALARY_DATE, EMPLOYEE_MOBILE)

VALUES

(101, 'Rajesh Kumar', 'Principal', 35000, '2024-08-01', '9712345678'),

(102, 'Sita Devi', 'Vice Principal', 30000, '2024-08-01', '9723456789'),

(103, 'Arun Reddy', 'Tamil Staff', 25000, '2024-08-01', '9734567890'),

(104, 'Anita Sharma', 'English Staff', 26000, '2024-08-01', '9745678901'),

(105, 'Muthu Kumar', 'Maths Staff', 27000, '2024-08-01', '9756789012'),

(106, 'Preetha', 'Science Staff', 28000, '2024-08-01', '9767890123'),

(107, 'Kumar Raj', 'SS Staff', 19000, '2024-08-01', '9778901234'),

(108, 'Sanjay Menon', 'PET Master', 20000, '2024-08-01', '9789012345'),

(109, 'Meena', 'Administrator', 12000, '2024-08-01', '9790123456'),

(110, 'Ravi', 'Incharge', 15000, '2024-08-01', '9701234567'),

(111, 'Deepak', 'Bus Driver', 15000, '2024-08-01', '9712345679'),

(112, 'Lathan', 'Bus Driver', 15000, '2024-08-01', '9723456790'),

(113, 'Karthik', 'Bus Driver', 15000, '2024-08-01', '9734567901'),

(114, 'Nilan', 'Bus Driver', 15000, '2024-08-01', '9745679012'),

(115, 'Arjun', 'Bus Driver', 15000, '2024-08-01', '9756789123'),

(116, 'Suresh', 'Watch Men', 12000, '2024-08-01', '9767890234'),

(117, 'Sundar', 'Watch Men', 12000, '2024-08-01', '9778901345'),

(118, 'Harini', 'House Keeping', 8000, '2024-08-01', '9789012456'),

(119, 'Kavitha', 'House Keeping', 8000, '2024-08-01', '9790123567');

-- INSERT RECORDS INTO THE CLASSROOM TABLE

```
INSERT INTO ClassRoom (Class_RoomID, ClassRoom_InchargeID, Class, ClassRoom_Capacity)
VALUES
('1class', 103, 1, 25),
('2class', 105, 2, 25),
('3class', 106, 3, 25),
('4class', 104, 4, 25),
('5class', 107, 5, 25);
```

-- INSERT RECORDS INTO THE SCHOOL_BUS TABLE

```
INSERT INTO school_bus (SCHOOL_BUS_ID, SCHOOL_BUS_NUMBER, EMPLOYEE_ID,
STUDENT_ID, ROUTE)
VALUES
('1A', 'TN70M2611', 111, 1001, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 1002, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 1003, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 1004, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 2001, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 2002, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 2003, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 2004, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 3001, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 3002, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 3003, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 3004, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 4001, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 4002, 'SALEM OUTER'),
('1A', 'TN70M2611', 111, 4003, 'SALEM OUTER'),
```

('1A', 'TN70M2611', 111, 4004, 'SALEM OUTER'),

('1A', 'TN70M2611', 111, 5001, 'SALEM OUTER'),

('1A', 'TN70M2611', 111, 5002, 'SALEM OUTER'),

('1A', 'TN70M2611', 111, 5003, 'SALEM OUTER'),

('1A', 'TN70M2611', 111, 5004, 'SALEM OUTER'),

('3C', 'TN70M2002', 112, 1005, 'SALEM'),

('3C', 'TN70M2002', 112, 1006, 'SALEM'),

('3C', 'TN70M2002', 112, 1007, 'SALEM'),

('3C', 'TN70M2002', 112, 1008, 'SALEM'),

('3C', 'TN70M2002', 112, 1009, 'SALEM'),

('3C', 'TN70M2002', 112, 2005, 'SALEM'),

('3C', 'TN70M2002', 112, 2006, 'SALEM'),

('3C', 'TN70M2002', 112, 2007, 'SALEM'),

('3C', 'TN70M2002', 112, 2008, 'SALEM'),

('3C', 'TN70M2002', 112, 2009, 'SALEM'),

('3C', 'TN70M2002', 112, 3005, 'SALEM'),

('3C', 'TN70M2002', 112, 3006, 'SALEM'),

('3C', 'TN70M2002', 112, 3007, 'SALEM'),

('3C', 'TN70M2002', 112, 3008, 'SALEM'),

('3C', 'TN70M2002', 112, 3009, 'SALEM'),

('3C', 'TN70M2002', 112, 4005, 'SALEM'),

('3C', 'TN70M2002', 112, 4006, 'SALEM'),

('3C', 'TN70M2002', 112, 4007, 'SALEM'),

('3C', 'TN70M2002', 112, 4008, 'SALEM'),

('3C', 'TN70M2002', 112, 4009, 'SALEM'),

('3C', 'TN70M2002', 112, 5005, 'SALEM'),

('3C', 'TN70M2002', 112, 5006, 'SALEM'),

('3C', 'TN70M2002', 112, 5007, 'SALEM'),

('3C', 'TN70M2002', 112, 5008, 'SALEM'),

('3C', 'TN70M2002', 112, 5009, 'SALEM'),

('11C', 'TN70M1234', 113, 1010, 'OMALUR'),

('11C', 'TN70M1234', 113, 1011, 'OMALUR'),

('11C', 'TN70M1234', 113, 1012, 'OMALUR'),

('11C', 'TN70M1234', 113, 2010, 'OMALUR'),

('11C', 'TN70M1234', 113, 2011, 'OMALUR'),

('11C', 'TN70M1234', 113, 2012, 'OMALUR'),

('11C', 'TN70M1234', 113, 3010, 'OMALUR'),

('11C', 'TN70M1234', 113, 3011, 'OMALUR'),

('11C', 'TN70M1234', 113, 3012, 'OMALUR'),

('11C', 'TN70M1234', 113, 4010, 'OMALUR'),

('11C', 'TN70M1234', 113, 4011, 'OMALUR'),

('11C', 'TN70M1234', 113, 4012, 'OMALUR'),

('11C', 'TN70M1234', 113, 5010, 'OMALUR'),

('11C', 'TN70M1234', 113, 5011, 'OMALUR'),

('11C', 'TN70M1234', 113, 5012, 'OMALUR'),

('15C', 'TN70M7788', 114, 1013, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 1014, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 1015, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 1016, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 2013, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 2014, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 2015, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 2016, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 3013, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 3014, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 3015, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 3016, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 4013, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 4014, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 4015, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 4016, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 5013, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 5014, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 5015, 'VALAPHADY'),

('15C', 'TN70M7788', 114, 5016, 'VALAPHADY'),

('15A', 'TN70M9999', 115, 1017, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 1018, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 1019, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 1020, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 2017, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 2018, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 2019, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 2020, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 3017, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 3018, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 3019, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 3020, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 4017, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 4018, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 4019, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 4020, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 5017, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 5018, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 5019, 'SALEM BUS STAND'),

('15A', 'TN70M9999', 115, 5020, 'SALEM BUS STAND');

-- INSERT RECORDS INTO THE STUDENT_FEES TABLE

INSERT INTO student_fees (STUDENT_ID, FEES_AMOUNT, FEES_PAY, DUE_DATE)

VALUES

(1001, '20000', '15000', '2024-09-01'),

(1002, '20000', '14500', '2024-09-05'),

(1003, '20000', '12000', '2024-09-10'),

(1004, '20000', '13000', '2024-09-15'),

(1005, '20000', '14000', '2024-09-20'),

(1006, '20000', '11000', '2024-09-25'),

(1007, '20000', '10500', '2024-10-01'),

(1008, '20000', '11500', '2024-10-05'),

(1009, '20000', '12500', '2024-10-10'),

(1010, '20000', '13500', '2024-10-15'),

(1011, '20000', '14500', '2024-10-20'),

(1012, '20000', '10000', '2024-10-25'),

(1013, '20000', '12000', '2024-11-01'),

(1014, '20000', '15000', '2024-11-05'),

(1015, '20000', '14000', '2024-11-10'),

(1016, '20000', '12500', '2024-11-15'),

(1017, '20000', '11000', '2024-11-20'),

(1018, '20000', '13500', '2024-11-25'),

(1019, '20000', '14500', '2024-12-01'),

(1020, '20000', '10000', '2024-12-05'),

(2001, '30000', '25000', '2024-09-01'),

(2002, '30000', '22000', '2024-09-05'),

(2003, '30000', '21000', '2024-09-10'),

(2004, '30000', '20000', '2024-09-15'),

(2005, '30000', '18000', '2024-09-20'),

(2006, '30000', '16000', '2024-09-25'),

(2007, '30000', '15000', '2024-10-01'),

(2008, '30000', '14000', '2024-10-05'),

(2009, '30000', '13000', '2024-10-10'),

(2010, '30000', '12000', '2024-10-15'),

(2011, '30000', '11000', '2024-10-20'),

(2012, '30000', '10500', '2024-10-25'),

(2013, '30000', '26000', '2024-11-01'),

(2014, '30000', '24000', '2024-11-05'),

(2015, '30000', '23000', '2024-11-10'),

(2016, '30000', '22000', '2024-11-15'),

(2017, '30000', '21000', '2024-11-20'),

(2018, '30000', '20000', '2024-11-25'),

(2019, '30000', '19000', '2024-12-01'),

(2020, '30000', '18000', '2024-12-05'),

(3001, '30000', '25000', '2024-12-10'),

(3002, '30000', '24000', '2024-12-15'),

(3003, '30000', '23000', '2024-12-20'),

(3004, '30000', '22000', '2024-12-25'),

(3005, '30000', '21000', '2025-01-01'),

(3006, '30000', '20000', '2025-01-05'),

(3007, '30000', '19000', '2025-01-10'),

(3008, '30000', '18000', '2025-01-15'),

(3009, '30000', '17000', '2025-01-20'),

(3010, '30000', '16000', '2025-01-25'),

(3011, '30000', '15000', '2025-02-01'),

(3012, '30000', '14000', '2025-02-05'),

(3013, '30000', '13000', '2025-02-10'),

(3014, '30000', '12000', '2025-02-15'),

(3015, '30000', '11000', '2025-02-20'),

(3016, '30000', '10000', '2025-02-25'),

(3017, '30000', '15000', '2025-03-01'),

(3018, '30000', '14000', '2025-03-05'),

(3019, '30000', '13000', '2025-03-10'),

(3020, '30000', '12000', '2025-03-15'),

(4001, '40000', '35000', '2024-09-01'),

(4002, '40000', '34000', '2024-09-05'),

(4003, '40000', '33000', '2024-09-10'),

(4004, '40000', '32000', '2024-09-15'),

(4005, '40000', '31000', '2024-09-20'),

(4006, '40000', '30000', '2024-09-25'),

(4007, '40000', '29000', '2024-10-01'),

(4008, '40000', '28000', '2024-10-05'),

(4009, '40000', '27000', '2024-10-10'),

(4010, '40000', '26000', '2024-10-15'),

(4011, '40000', '25000', '2024-10-20'),

(4012, '40000', '24000', '2024-10-25'),

(4013, '40000', '23000', '2024-11-01'),

(4014, '40000', '22000', '2024-11-05'),

(4015, '40000', '21000', '2024-11-10'),

(4016, '40000', '20000', '2024-11-15'),

(4017, '40000', '19000', '2024-11-20'),

(4018, '40000', '18000', '2024-11-25'),

(4019, '40000', '17000', '2024-12-01'),

(4020, '40000', '16000', '2024-12-05'),

(5001, '40000', '35000', '2024-12-10'),

(5002, '40000', '34000', '2024-12-15'),

(5003, '40000', '33000', '2024-12-20'),

(5004, '40000', '32000', '2024-12-25'),

(5005, '40000', '31000', '2025-01-01'),

(5006, '40000', '30000', '2025-01-05'),

(5007, '40000', '29000', '2025-01-10'),

(5008, '40000', '28000', '2025-01-15'),

(5009, '40000', '27000', '2025-01-20'),

(5010, '40000', '26000', '2025-01-25'),

(5011, '40000', '25000', '2025-02-01'),

(5012, '40000', '24000', '2025-02-05'),

(5013, '40000', '23000', '2025-02-10'),

(5014, '40000', '22000', '2025-02-15'),

(5015, '40000', '21000', '2025-02-20'),

(5016, '40000', '20000', '2025-02-25'),

(5017, '40000', '19000', '2025-03-01'),

(5018, '40000', '18000', '2025-03-05'),

(5019, '40000', '17000', '2025-03-10'),

(5020, '40000', '16000', '2025-03-15');

1.5 CASE STUDY

1) select all students name and list?

```
select * from student;
```

2) What are the names of all students in class 5?

```
SELECT STUDENT_NAME
FROM STUDENT
WHERE CLASS = 5;
```

3)What is the mobile number of the parent of student with ID 1001?

```
SELECT PARENT_MOBILE
FROM STUDENT
WHERE STUDENT_ID = 1001;
```

4) What is the total salary paid to all employees?

```
SELECT SUM(SALARY) AS TOTAL_SALARY
FROM Employees;
```

5)Which school bus routes are handled by employee with ID 112?

```
SELECT ROUTE
```

```
FROM School_Bus
WHERE EMPLOYEE_ID = 112;
```

6)What are the names and mobile numbers of employees who earn more than $30,000?

```
SELECT EMPLOYEE_NAME, EMPLOYEE_MOBILE
FROM Employees
WHERE SALARY > 30000;
```

7)What is the fee amount due for student ID 2002?

```
SELECT FEES_AMOUNT
FROM student_fees
WHERE STUDENT_ID = 2002;
```

8)Which students are assigned to bus number 'TN70M2611'?

```
SELECT s.STUDENT_NAME
FROM STUDENT s
JOIN School_Bus b ON s.STUDENT_ID = b.STUDENT_ID
WHERE b.SCHOOL_BUS_NUMBER = 'TN70M2611';
```

9)What is the due date for the fee payment of student ID 1005?

```
SELECT DUE_DATE
FROM student_fees
```

```
WHERE STUDENT_ID = 1005;
```

10)How many students are there in each class?

```
SELECT CLASS, COUNT(STUDENT_ID) AS TOTAL_STUDENTS
FROM STUDENT
GROUP BY CLASS;
```

11)What are the names and routes of all school buses?

```
SELECT SCHOOL_BUS_NUMBER, ROUTE
FROM School_Bus;
```

12)What are the names and salaries of employees with the position 'Administrator'?

```
SELECT EMPLOYEE_NAME, SALARY
FROM Employees
WHERE POSITION = 'Administrator';
```

13)What is the average salary of employees?

```
SELECT AVG(SALARY) AS AVERAGE_SALARY
FROM Employees;
```

14)What are the names and classes of students along with their assigned school bus numbers?

```
SELECT s.STUDENT_NAME, s.CLASS, b.SCHOOL_BUS_NUMBER
FROM STUDENT s
JOIN School_Bus b ON s.STUDENT_ID = b.STUDENT_ID;
```

15)What are the names of employees who are assigned to the same school bus as student with ID 113?

```
SELECT e.EMPLOYEE_NAME
FROM Employees e
JOIN School_Bus b ON e.EMPLOYEE_ID = b.EMPLOYEE_ID
WHERE b.STUDENT_ID = 113;
```

1.6 STORED PROCEDURE & SIMPLE QUESTION

CREATE A STORED PROCEDURE ADD STUDENT AND CALL THE PROCEDURE

```
DELIMITER //
CREATE PROCEDURE add_student (
  S_STUDENT_ID INT,
  S_STUDENT_NAME VARCHAR(100),
  S_CLASS INT,
  S_PARENT_NAME VARCHAR(100),
  S_PARENT_MOBILE VARCHAR (100) )

BEGIN
   INSERT INTO student (STUDENT_ID, STUDENT_NAME, CLASS, PARENT_NAME,
   PARENT_MOBILE)
   VALUES (S_STUDENT_ID, S_STUDENT_NAME, S_CLASS, S_PARENT_NAME,
   S_PARENT_MOBILE);
```

```
END //

DELIMITER
```

CALL THE PROCEDURE

```
CALL add_student(1021, 'John Doe', 1, 'Jane Doe', '9712345678');
```

1) CREATE A STORED PROCEDURE REMOVE STUDENT DATA FROM TABLE

```
DELIMITER //

CREATE PROCEDURE remove_student (S_STUDENT_ID INT  )

BEGIN

   DELETE FROM student
   WHERE STUDENT_ID = S_STUDENT_ID;

END //

DELIMITER
```

CALL THE PROCEDURE

```
CALL remove_student(1001);
```

2) CREATE PROCEDURE UPDATE A STUDENT DETAILS

UPDATE STUDENT ID ANSD CLASS

```
DELIMITER //

CREATE PROCEDURE update_student_name_class(
    S_STUDENT_ID INT,
    S_STUDENT_NAME VARCHAR(100),
    S_CLASS INT   )

BEGIN
```

```
   UPDATE student
   SET STUDENT_NAME = S_STUDENT_NAME,
   CLASS = S_CLASS
   WHERE STUDENT_ID = S_STUDENT_ID;
END //

DELIMITER ;
```

CALL THE PROCEDURE

```
CALL update_student_name_class(1001, 'John Smith', 3);
```

3) CREATE STORED PROCEDURE ADD EXTRA EMPLOYEE

```
DELIMITER //

CREATE PROCEDURE add_employee(
    E_EMPLOYEE_ID INT,
    E_EMPLOYEE_NAME VARCHAR(100),
    E_POSITION VARCHAR(100),
    E_SALARY INT,
    E_SALARY_DATE DATE,
    E_EMPLOYEE_MOBILE VARCHAR (100) )

BEGIN

   INSERT INTO employee (
      EMPLOYEE_ID,
      EMPLOYEE_NAME,
      POSITION,
      SALARY,
      SALARY_DATE,
      EMPLOYEE_MOBILE)

   VALUES (
      E_EMPLOYEE_ID,
      E_EMPLOYEE_NAME,
      E_POSITION,
      E_SALARY,
      E_SALARY_DATE,
```

```
    E_EMPLOYEE_MOBILE);

END //

DELIMITER ;
```

CALL THE PROCEDURE

```
CALL add_employee (120, 'Alice Johnson', 'Administrator', 50000, '2024-08-01', '9712345678');
```

4) CREATE PROCEDURE REMOVE EMPLOYEE DATA

```
 DELIMITER //

CREATE PROCEDURE remove_employee(
IN E_EMPLOYEE_ID INT)

BEGIN

   DELETE FROM employee
   WHERE EMPLOYEE_ID = E_EMPLOYEE_ID;
END //

DELIMITER ;
```

CALL THE PROCEDURE

```
CALL remove_employee (120);
```

5) CREATE STORED PROCEDURE UPDATE EMPLOYE SALARY

```
DELIMITER //

CREATE PROCEDURE update_employee_salary(
   E_EMPLOYEE_ID INT,
E_NEW_SALARY INT)

BEGIN
```

```
   UPDATE employee
   SET SALARY = E_NEW_SALARY
   WHERE EMPLOYEE_ID = E_EMPLOYEE_ID;
END //

DELIMITER ;
```

CALL THE PROCUDRE

```
CALL update_employee_salary(101, 55000);
```

8) CREATE PROCEDURE FIND THE STUDENT BALANCE FEES

```
DELIMITER //

CREATE PROCEDURE get_student_balance(
F_STUDENT_ID INT)

BEGIN

   SELECT (FEES_AMOUNT -
FEES_PAY ) AS 'BALANCE
AMOUNT '     FROM      student_fees
   WHERE
   STUDENT_ID = F_STUDENT_ID;
END //

DELIMITER   ;
```

CALL THE PROCUDURE

```
Call get_student_balance (1001);
```

9)CREATE PROCEDURE ADD NEW VEHICAL

```
DELIMITER //

CREATE PROCEDURE add_vehicle(
    S_SCHOOL_BUS_ID VARCHAR(100),
    S_SCHOOL_BUS_NUMBER VARCHAR(100),
    S_EMPLOYEE_ID INT,
```

```
    S_STUDENT_ID INT,
S_ROUTE VARCHAR (100)  )

BEGIN

   INSERT INTO School_Bus (
      SCHOOL_BUS_ID,
      SCHOOL_BUS_NUMBER,
      EMPLOYEE_ID,
      STUDENT_ID,
      ROUTE)
      VALUES (
      S_SCHOOL_BUS_ID,
      S_SCHOOL_BUS_NUMBER,
      S_EMPLOYEE_ID,
      S_STUDENT_ID,
      S_ROUTE);

END //

DELIMITER ;

CALL  THE PROCEDURE

CALL add_vehicle ('5A', 'TN70M1235', 116, 1001, 'NEW ROUTE');

10)   CREATE THE PROCEDURE FIND EMPLOYEE SALARY

 DELIMITER //

   CREATE PROCEDURE find_employee_salary (E_EMPLOYEE_ID INT)

BEGIN
   SELECT  SALARY
   FROM   Employee
   WHERE   EMPLOYEE_ID = E_EMPLOYEE_ID;
END //

DELIMITER ;

CALL  THE PROCEDURE
```

```
CALLfind_employee_salary(101);
```

1.7VIEW FUNCTIONS

1)Create a view function for view all student records and call the view function

```
CREATE VIEW Student_View AS
SELECT
   STUDENT_ID,
   STUDENT_NAME,
   CLASS,
   PARENT_NAME,
   PARENT_MOBILE
FROM
       STUDENT;
```

Call the view ;

```
SELECT * FROM Student_View;
```

2)Create the view function only view in 5 class student details ;

```
CREATE VIEW StudentClass5View AS
SELECT
   STUDENT_ID,
   STUDENT_NAME,
   CLASS,
   PARENT_NAME,
   PARENT_MOBILE
FROM
   STUDENT
WHERE
   CLASS = 10;
```

Call the view function ;

```
SELECT * FROM StudentClass5View;
```

3)Create the view function which employee handle each classroom

```
CREATE VIEW EmployeesAndClassrooms AS
SELECT
    e.EMPLOYEE_ID,
    e.EMPLOYEE_NAME,
    e.POSITION,
    e.SALARY,
    e.SALARY_DATE,
    e.EMPLOYEE_MOBILE,
    c.Class_RoomID,
    c.Class,
    c.ClassRoom_Capacity
FROM
    Employees e
LEFT JOIN
    ClassRoom c ON e.EMPLOYEE_ID = c.ClassRoom_InchargeID;
```

Call the view function :

```
SELECT * FROM EmployeesAndClassrooms;
```

4)Create the view function student and class room details ; CREATE VIEW

```
ClassroomWithStudentClass AS
SELECT
   c.Class_RoomID,
```

```
    c.Class,
    c.ClassRoom_Capacity,
    s.STUDENT_ID AS Student_ID,
    s.STUDENT_NAME AS Student_Name
FROM
    ClassRoom c
JOIN
    STUDENT s ON c.Class = s.CLASS;
```

Call the view ;

```
SELECT * FROM ClassroomWithStudentClass;
```

1.8 JOINS

1)Create the joins between student and employee table

```
SELECT
    s.STUDENT_ID,
    s.STUDENT_NAME,
    s.CLASS,
    e.EMPLOYEE_ID AS Incharge_ID,
    e.EMPLOYEE_NAME AS Incharge_Name,
    e.POSITION AS Incharge_Position,
    e.SALARY AS Incharge_Salary,
    e.SALARY_DATE AS Incharge_Salary_Date,
    e.EMPLOYEE_MOBILE AS Incharge_Mobile
FROM
    Student  s
JOIN
    classroom c ON s.CLASS = c.Class
JOIN
    Employees e ON c.ClassRoom_InchargeID = e.EMPLOYEE_ID;
```

2)Create the left join between student and school fees

```
SELECT
   s.STUDENT_ID,
   s.STUDENT_NAME,
   s.CLASS,
   s.PARENT_NAME,
   s.PARENT_MOBILE,
   f.FEES_AMOUNT,
   f.FEES_PAY,
   f.DUE_DATE
FROM
   student s
 left JOIN
   student_fees f ON s.STUDENT_ID = f.STUDENT_ID;
```

3)Create the right join between student ,employee school bus

```
SELECT*
FROM
   school_bus b
RIGHT JOIN
   student s ON b.STUDENT_ID = s.STUDENT_ID
LEFT JOIN
   employees e ON b.EMPLOYEE_ID = e.EMPLOYEE_ID;
```

1.9 CONCLUSION

- The School Management System exemplifies the effective application of SQL in handling diverse administrative functions within an educational institution. By utilizing SQL's advanced features, such as queries, joins, subqueries, views, and stored procedures, the system adeptly manages essential components including student records, employee details, salary information, and school fees.

- This project emphasizes the critical role of structured data management in ensuring smooth and efficient school operations. Centralizing data and automating routine processes not only enhances operational efficiency but also improves accuracy and scalability. The system supports better decision-making and ensures transparency in financial and administrative matters, ultimately benefiting students, staff, and school administrators.

- Overall, the School Management System demonstrates how SQL can be leveraged to create a robust, reliable, and user-friendly database solution. This project highlights SQL's capability to streamline complex data management tasks and provides significant advantages in managing educational institutions effectively.