```
CREATE TABLE department (
   dept_id INT PRIMARY KEY,
   dept_name VARCHAR(100)
);
```

Creating the 'year' table

```
CREATE TABLE year (
   year_id INT PRIMARY KEY,
   year_name VARCHAR(50)
);
```

Creating the 'students' table with relationships to 'department' and 'year'

```
CREATE TABLE students (
   student_id INT PRIMARY KEY,
   student_name VARCHAR(100),
   dept_id INT,
   year_id INT,
   FOREIGN KEY (dept_id) REFERENCES department(dept_id),
   FOREIGN KEY (year_id) REFERENCES year(year_id)
);
```

Inserting Data:

Inserting data into the 'department' table

```
INSERT INTO department (dept_id, dept_name) VALUES
(1, 'CSE'),
(2, 'ECE'),
(3, 'Mechanical'),
(4, 'Civil');
```

Inserting data into the 'year' table

```
INSERT INTO year (year_id, year_name) VALUES
```

```
(1, 'First Year'),
(2, 'Second Year'),
(3, 'Third Year'),
(4, 'Fourth Year');
```

Inserting students into the 'students' table

```
INSERT INTO students (student_id, student_name, dept_id, year_id) VALUES
(1, 'John Doe', 1, 2),   -- CSE
(2, 'Jane Smith', 1, 3), -- CSE
(3, 'Alice Johnson', 1, 1), -- CSE
(4, 'Bob Brown', 1, 2),  -- CSE
(5, 'Charlie Davis', 1, 4), -- CSE

(6, 'David Lee', 2, 2),  -- ECE
(7, 'Eva Martinez', 2, 1), -- ECE
(8, 'Frank Wilson', 2, 3), -- ECE
(9, 'Grace Taylor', 2, 4), -- ECE
(10, 'Harry Clark', 2, 3); -- ECE
```

Similarly insert for other departments

Query to Display Students from CSE Department:

```
SELECT *
FROM students
WHERE dept_id = 1; -- Assuming 'CSE' has dept_id = 1
```

This query retrieves all students from the 'CSE' department.

```
+------------+--------------+---------+---------+
| student_id | student_name | dept_id | year_id |
+------------+--------------+---------+---------+
|    1     |  John Doe  |   1   |   2   |
|    2     | Jane Smith |   1   |   3   |
|     3     | Alice Johnson|    1    |    1    |

|     4     |   Bob Brown  |    1    |    2    |

|     5     | Charlie Davis|    1    |    4    |

+------------+--------------+---------+---------+
```

Query to Display Only Department Names Using Student Table:

```
SELECT DISTINCT department.dept_name

FROM department

JOIN students ON department.dept_id = students.dept_id;
```

This query retrieves distinct department names using the students table and joining with the department table.

```
+-----------+

| dept_name |

+-----------+

|    CSE    |

|    ECE    |

|Mechanical |

|   Civil   |

+-----------+
```

Query to Display Students Sorted by Department and First Name:

```
SELECT students.student_id, students.student_name, department.dept_name

FROM students

JOIN department ON students.dept_id = department.dept_id

ORDER BY department.dept_name, students.student_name;

+------------+--------------+-----------+

| student_id | student_name | dept_name |
+------------+--------------+-----------+
|     1      |   John Doe   |    CSE    |
|     3      | Alice Johnson|    CSE    |
|     4      |   Bob Brown  |    CSE    |
|     5      | Charlie Davis|    CSE    |
|     2      |  Jane Smith  |    CSE    |
|     6      |   David Lee  |    ECE    |
|     7      |  Eva Martinez|    ECE    |
|     10     |  Harry Clark |    ECE    |
|     8      |  Frank Wilson|    ECE    |
|     9      |  Grace Taylor|    ECE    |
+------------+--------------+-----------+
```

Writing queries in mangodb

```
// Creating the 'department' collection
db.createCollection("department");

// Creating the 'year' collection
db.createCollection("year");

// Creating the 'students' collection
db.createCollection("students");
```

Inserting Data

You can insert data directly into MongoDB collections using the insertOne() or insertMany() methods.

```
// Inserting data into the 'department' collection
db.department.insertMany([
   { dept_id: 1, dept_name: "CSE" },
   { dept_id: 2, dept_name: "ECE" },
   { dept_id: 3, dept_name: "Mechanical" },
   { dept_id: 4, dept_name: "Civil" }
]);


// Inserting data into the 'year' collection
db.year.insertMany([
   { year_id: 1, year_name: "First Year" },
   { year_id: 2, year_name: "Second Year" },
   { year_id: 3, year_name: "Third Year" },
   { year_id: 4, year_name: "Fourth Year" }
]);


// Inserting data into the 'students' collection
db.students.insertMany([
   { student_id: 1, student_name: "John Doe", dept_id: 1, year_id: 2 },
   { student_id: 2, student_name: "Jane Smith", dept_id: 1, year_id: 3 },
   { student_id: 3, student_name: "Alice Johnson", dept_id: 1, year_id: 1 },
   { student_id: 4, student_name: "Bob Brown", dept_id: 1, year_id: 2 },
   { student_id: 5, student_name: "Charlie Davis", dept_id: 1, year_id: 4 },


   { student_id: 6, student_name: "David Lee", dept_id: 2, year_id: 2 },
   { student_id: 7, student_name: "Eva Martinez", dept_id: 2, year_id: 1 },
   { student_id: 8, student_name: "Frank Wilson", dept_id: 2, year_id: 3 },
   { student_id: 9, student_name: "Grace Taylor", dept_id: 2, year_id: 4 },
   { student_id: 10, student_name: "Harry Clark", dept_id: 2, year_id: 3 }
]);
```

Queries (Equivalent to SQL Queries):

MongoDB queries use a different syntax compared to SQL queries. Here's how you can perform similar operations in MongoDB:

Query to Display Students from CSE Department:

```
db.students.find({ dept_id: 1 }); // Assuming 'CSE' has dept_id = 1
```

Query to Display Only Department Names Using Student Collection:

```
db.students.distinct("dept_name");
```

Query to Display Students Sorted by Department and First Name:

```
db.students.aggregate([
   {
      $lookup: {
         from: "department",
         localField: "dept_id",
         foreignField: "dept_id",
         as: "department"
      }
   },
   { $unwind: "$department" },
   { $sort: { "department.dept_name": 1, "student_name": 1 } },
   { $project: { _id: 0, student_id: 1, student_name: 1, "department.dept_name": 1 } }
]);
```