

1. ****Create Tables in MySQL****:

```
CREATE TABLE students (  
    student_id INT AUTO_INCREMENT PRIMARY KEY,  
    student_name VARCHAR(50),  
    department_id INT,  
    year_id INT  
);
```

```
CREATE TABLE department (  
    department_id INT AUTO_INCREMENT PRIMARY KEY,  
    department_name VARCHAR(50)  
);
```

```
CREATE TABLE year (  
    year_id INT AUTO_INCREMENT PRIMARY KEY,  
    year_name VARCHAR(20)  
);
```

2. ****Establish Relationships****:

In MySQL, you would establish foreign key relationships between the `students` table and both `department` and `year` tables. Assuming `department_id` and `year_id` are foreign keys referencing the respective primary keys in the `department` and `year` tables.

```
ALTER TABLE students
```

ADD FOREIGN KEY (department_id) REFERENCES department(department_id),

ADD FOREIGN KEY (year_id) REFERENCES year(year_id);

3. ****Converting to MongoDB****:

MongoDB is a NoSQL database, so it doesn't have the concept of tables, but rather collections. You can model the data in MongoDB using JSON-like documents.

For the `students` collection:

```
{  
  "student_id": 1,  
  "student_name": "John Doe",  
  "department_id": 1,  
  "year_id": 1;  
}
```

For the `department` collection:

```
{  
  "department_id": 1,  
  "department_name": "CSE";  
}
```

For the `year` collection:

```
{  
  "year_id": 1,  
  "year_name": "First Year"  
}
```

4. ****Insert Data****:

```
INSERT INTO students (student_name, department_id, year_id) VALUES  
  
('Alice', 1, 1),  
('Bob', 1, 1),  
('Charlie', 1, 1),  
('David', 1, 1),  
('Eve', 1, 1);
```

5. ****Query to Display Students from CSE Department****:

In MySQL:

```
SELECT * FROM students WHERE department_id = 1;
```

In MongoDB:

```
db.students.find({ department_id: 1 });
```

6. ****Query to Display Department Name Using Student Table****:

In MySQL:

```
SELECT department_name FROM department WHERE department_id IN (SELECT department_id FROM students);
```

In MongoDB, it depends on how you've structured your data. If you embedded department information within the student document, you could do:

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

Then, using the distinct department IDs, you could fetch the department names from the `department` collection.

7. ****Query to Display Students Sorted by Department and First Name****:

In MySQL:

```
SELECT s.student_name, d.department_name
FROM students s
JOIN department d ON s.department_id = d.department_id
ORDER BY d.department_name, s.student_name;
```

In MongoDB, you can achieve this using the aggregation framework:

```
db.students.aggregate([
  {
```

```
$lookup: {
  from: "department",
  localField: "department_id",
  foreignField: "department_id",
  as: "department"
},
{
  $unwind: "$department"
},
{
  $sort: {
    "department.department_name": 1,
    "student_name": 1
  }
},
{
  $project: {
    _id: 0,
    student_name: 1,
    department_name: "$department.department_name"
  }
}
});
```