

creating department table

```
mysql> CREATE TABLE department (  
-> dept_id INT PRIMARY KEY AUTO_INCREMENT,  
-> dept_name VARCHAR(50) );  
Query OK, 0 rows affected (0.01 sec)
```

creating year table

```
mysql> CREATE TABLE year (  
-> year_id INT PRIMARY KEY AUTO_INCREMENT,  
-> year_name VARCHAR(20) );  
Query OK, 0 rows affected (0.01 sec)
```

Student should contain relationship to both department and year

```
mysql> CREATE TABLE students (  
-> student_id INT PRIMARY KEY AUTO_INCREMENT,  
-> student_name VARCHAR(50) ,  
-> dept_id INT,  
-> year_id INT,  
-> FOREIGN KEY (dept_id) REFERENCES department(dept_id),  
-> FOREIGN KEY (year_id) REFERENCES year(year_id));  
Query OK, 0 rows affected (0.03 sec)
```

inserting values into student table

```
mysql> INSERT INTO students (student_name, dept_id, year_id) VALUES
```

```
-> ('Varun - CSE', 1, 1), ('Teja - CSE', 1, 2), ('Sai - CSE', 1, 3), ('Hari - CSE', 1, 4),
```

```
-> ('Pavan - ECE', 2, 1), ('Ram - ECE', 2, 2), ('Krishna - ECE', 2, 3), ('Tarun - ECE', 2, 4),
```

```
-> ('mahi - EE', 3, 1), ('virat - EE', 3, 2), ('kohli - EE', 3, 3), ('Sai - EE', 3, 4),
```

Query OK, 12 rows affected (0.00 sec)

Records: 12 Duplicates: 0 Warnings:

Displaying values of student table

```
mysql> select * from students;
```

```
+-----+-----+-----+-----+
| student_id | student_name | dept_id | year_id |
+-----+-----+-----+-----+
| 1 | Varun - CSE | 1 | 1 |
| 2 | Teja - CSE | 1 | 2 |
| 3 | Sai - CSE | 1 | 3 |
| 4 | Hari - CSE | 1 | 4 |
| 5 | Pavan - ECE | 2 | 1 |
| 6 | Ram - ECE | 2 | 2 |
| 7 | Krishna - ECE | 2 | 3 |
| 8 | Tarun - ECE | 2 | 4 |
| 9 | mahi - EE | 3 | 1 |
| 10 | virat- EE | 3 | 2 |
| 11 | kohli - EE | 3 | 3 |
| 12 | Sai- EE | 3 | 4 |
```

12 rows in set(0.00sec)

write a query to display students from CSE department

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

```
-> FROM students
```

```
-> JOIN department ON students.dept_id = department.dept_id
```

```
-> WHERE department.dept_name = 'CSE';
```

```
+-----+-----+-----+
| student_id | student_name | dept_name |
+-----+-----+-----+
| 1 | Varun - CSE | CSE |
| 2 | Teja - CSE | CSE |
| 3 | Sai - CSE | CSE |
| 4 | Hari - CSE | CSE |
+-----+-----+-----+
```

4 rows in set (0.01 sec)

write a query to display students sorted by dept and firstname

```
mysql> 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 |
+-----+-----+-----+
| 4 | Hari - CSE | CSE |
| 3 | Sai - CSE | CSE |
```

```
| 2 | Teja - CSE | CSE |
| 1 | Varun - CSE | CSE |
| 7 | Krishna - ECE | ECE |
| 5 | Pavan - ECE | ECE |
| 6 | Ram - ECE | ECE |
| 8 | Tarun - ECE | ECE |
| 11 | kohli - EE | EE |
| 9 | mahi - EE | EE |
| 12 | Sai- EE | EE |
| 10 | virat- EE | EE |
```

12 rows in set (0.01 sec)

write a query to display only department name using student table

```
mysql> SELECT DISTINCT department.dept_name
```

```
-> FROM students
```

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

```
+-----+
| dept_name |
+-----+
| CSE |
| ECE |
| EE |
+-----+
```

3 rows in set (0.00 sec)

Use chatgpt and ask like "this is my table in mysql how can i create same in mongodb

Gen-AI.department

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

[Find](#) [Indexes](#) [Schema Anti-Patterns 0](#) [Aggregation](#) [Search Indexes](#)

[Generate queries from natural language in Compass](#) [INSERT DOCUMENT](#)

[Filter](#) Type a query: { field: 'value' } [Reset](#) [Apply](#) [Options ▶](#)

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": ObjectId('664ecb29faf6f51eb983235d')
  "department": {
    "dept_id": {
      "$type": "objectId"
    },
    "dept_name": {
      "$type": "string"
    }
  }
}
```

Gen-AI.year

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

[Find](#) [Indexes](#) [Schema Anti-Patterns 0](#) [Aggregation](#) [Search Indexes](#)

[Generate queries from natural language in Compass](#) [INSERT DOCUMENT](#)

[Filter](#) Type a query: { field: 'value' } [Reset](#) [Apply](#) [Options ▶](#)

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": ObjectId('664ecc6bfaf6f51eb986caeb')
  "student": {
    "bsonType": "object",
    "required": [
      "student_id",
      "student_name"
    ],
    "properties": {
      "student_id": {
        "$type": "objectId"
      },
      "student_name": {
        "$type": "string"
      },
      "dept_id": {
        "$type": "objectId"
      },
      "year_id": {
        "$type": "objectId"
      }
    }
  }
}
```

Gen-AI.students

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

Find

Indexes

Schema Anti-Patterns (0)

Aggregation

Search Indexes

[Generate queries from natural language in Compass](#)

INSERT DOCUMENT

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('664ecbfbfaf6f51eb985891d')
year: Object
  bsonType: "object"
  required: Array (2)
    0: "year_id"
    1: "year_name"
  properties: Object
    year_id: Object
    year_name: Object
```