

1) Create 3 tables named students, department, year

```
mysql> use nikhitha;
```

Database changed

#creating department table

```
mysql> -- Create the department table
```

```
mysql> CREATE TABLE department (
```

```
-> dept_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
-> dept_name VARCHAR(50) NOT NULL -> );
```

Query OK, 0 rows affected (0.01 sec)

#creating year table

```
mysql>
```

```
mysql> -- Create the year table
```

```
mysql> CREATE TABLE year (
```

```
-> year_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
-> year_name VARCHAR(20) NOT NULL
```

```
-> );
```

Query OK, 0 rows affected (0.01 sec)

2) student should contain relationship to both department and year

#creating student table

```
mysql>
```

```
mysql> -- Create the students table with foreign key relationships
```

```
mysql> CREATE TABLE students (
```

```
-> student_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
-> student_name VARCHAR(100) NOT NULL,
```

```
-> 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)

mysql> desc department;

Field	Type	Null	Key	Default	Extra
dept_id	int	NO	PRI	NULL	auto_increment
dept_name	varchar(50)	NO		NULL	

2 rows in set (0.00 sec)

mysql> desc year;

Field	Type	Null	Key	Default	Extra
year_id	int	NO	PRI	NULL	auto_increment
year_name	varchar(20)	NO		NULL	

2 rows in set (0.00 sec)

mysql> desc student;

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	auto_increment
student_name	varchar(100)	NO		NULL	
dept_id	int	YES	MUL	NULL	
year_id	int	YES	MUL	NULL	

4 rows in set (0.00 sec)

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

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

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('664e258575100bc00ad002ce')
insert : "dept"
documents : Array (4)
```

```
_id: ObjectId('664e261275100bc00ad1660d')
insert : "dept"
documents : Array (4)
```

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](#)

Filter [Filter](#) Type a query: { field: 'value' }

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('664e278a75100bc00ad500d5')
insert : "year"
documents : Array (2)
```

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: 1-2 OF 2

```

_id: ObjectId('664e23fe75100bc00acc407f')
insert : "student"
documents : Array (3)

```

```

_id: ObjectId('664e246075100bc00acd24f2')
insert : "student"
documents : Array (2)

```

#inserting values into student table

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

```
-> ('Student 1 - CSE', 1, 1), ('Student 2 - CSE', 1, 2), ('Student 3 - CSE', 1, 3), ('Student 4 - CSE', 1, 4),
```

```
-> ('Student 1 - ECE', 2, 1), ('Student 2 - ECE', 2, 2), ('Student 3 - ECE', 2, 3), ('Student 4 - ECE', 2, 4),
```

```
-> ('Student 1 - EE', 3, 1), ('Student 2 - EE', 3, 2), ('Student 3 - EE', 3, 3), ('Student 4 - EE', 3, 4),
```

```
-> ('Student 1 - ME', 4, 1), ('Student 2 - ME', 4, 2), ('Student 3 - ME', 4, 3), ('Student 4 - ME', 4, 4),
```

```
-> ('Student 1 - Civil', 5, 1), ('Student 2 - Civil', 5, 2), ('Student 3 - Civil', 5, 3), ('Student 4 - Civil', 5, 4);
```

Query OK, 20 rows affected (0.00 sec)

Records: 20 Duplicates: 0 Warnings:

#Displaying values of student table

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

```

+-----+-----+-----+-----+
| student_id | student_name | dept_id | year_id |
+-----+-----+-----+-----+
| 1 | Student 1 - CSE | 1 | 1 |
| 2 | Student 2 - CSE | 1 | 2 |

```

```
| 3 | Student 3 - CSE | 1 | 3 |
| 4 | Student 4 - CSE | 1 | 4 |
| 5 | Student 1 - ECE | 2 | 1 |
| 6 | Student 2 - ECE | 2 | 2 |
| 7 | Student 3 - ECE | 2 | 3 |
| 8 | Student 4 - ECE | 2 | 4 |
| 9 | Student 1 - EE | 3 | 1 |
| 10 | Student 2 - EE | 3 | 2 |
10 rows in set(0.00sec)
```

5)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 | Student 1 - CSE | CSE |
| 2 | Student 2 - CSE | CSE |
| 3 | Student 3 - CSE | CSE |
| 4 | Student 4 - CSE | CSE |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

6)write a query to display only deptname 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 |

| ME |

| Civil |

+-----+

5 rows in set (0.00 sec)

7)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 |

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

| 17 | Student 1 - Civil | Civil |

| 18 | Student 2 - Civil | Civil |

| 19 | Student 3 - Civil | Civil |

| 20 | Student 4 - Civil | Civil |

| 1 | Student 1 - CSE | CSE |

| 2 | Student 2 - CSE | CSE |

| 3 | Student 3 - CSE | CSE |

| 4 | Student 4 - CSE | CSE |

| 5 | Student 1 - ECE | ECE |

9 rows in set (0.01 sec)

