1.Create the students table with relationships to both department and year

```
CREATE TABLE students ( s_id INT AUTO_INCREMENT PRIMARY KEY, s_name VARCHAR(50), d_name
VARCHAR(50),y_id INT,  FOREIGN KEY (d_name) REFERENCES department(d_name), FOREIGN KEY (y_id)
REFERENCES year(y_id));
```

```
Query OK,0 rows affected(0.01 sec)
```

2)student should contain relationship to both department and year

```
 CREATE TABLE department (d_name VARCHAR(50)  PRIMARY KEY,  d_id INT);
```

```
Query OK,0 rows affected(0.01 sec)
```

```
 CREATE TABLE year ( y_id INT AUTO_INCREMENT PRIMARY KEY,y_name VARCHAR(20));
```

```
Query OK,0 rows affected(0.01 sec)
```

3)store 5 students for each department:

```
INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (1, 'John Doe', 'cse', 1);

Query OK,1` rows affected(0.01 sec)

INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (2, 'Jane Smith', 'cse', 1);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (3, 'Michael Johnson', 'cse', 2);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (4, 'Emily Davis', 'cse', 2);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (5, 'David Brown', 'cse', 3);

Query OK,1` rows affected(0.01 sec)
```

```
INSERT INTO students (s_id, s_name, d_name, y_id) VALUES(6, 'Sarah Wilson', 'Ece', 1);
```

```
 Query OK,1` rows affected(0.01 sec)

INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (7, 'Daniel Martinez', 'Ece', 1);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id) VALUES  (8, 'Jessica Anderson', 'Ece', 2);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id) VALUES  (9, 'Christopher Taylor', 'Ece', 2);

Query OK,1` rows affected(0.01 sec)

INSERT INTO students (s_id, s_name, d_name, y_id) VALUES (10, 'Ashley Thomas', 'Ece', 3);

Query OK,1` rows affected(0.01 sec)


 select *from students;

INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES (11, 'Matthew Lee', 'Civil', 1);

 Query OK,1` rows affected(0.01 sec)

INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES (12, 'Amanda White', 'Civil', 1);

Query OK,1` rows affected(0.01 sec)

  INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES (13, 'Ryan Garcia', 'Civil', 2);

Query OK,1` rows affected(0.01 sec)

  INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES (14, 'Brittany Hall', "Civil", 2);

Query OK,1` rows affected(0.01 sec)

  INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES (15, 'Olivia Clark', 'Civil', 3);

Query OK,1` rows affected(0.01 sec)


 INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES  (16, 'Nicholas Perez', 'Mech', 1);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES  (17, 'Kayla Hernandez', 'Mech', 1);
```

```
Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES    (18, 'Justin Young', 'Mech', 2);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES  (19, 'Lauren King', 'Mech', 2);

Query OK,1` rows affected(0.01 sec)

 INSERT INTO students (s_id, s_name, d_name, y_id)  VALUES  (20, 'Brandon Wright', 'Mech', 3);

Query OK,1` rows affected(0.01 sec)
```

5)write a query to display students from CSE department

```
SELECT *  FROM students  WHERE d_name='cse';

Query OK,1` rows affected(0.01 sec)
```

```
+------+-----------------+--------+------+
| s_id | s_name          | d_name | y_id |
+------+-----------------+--------+------+
|    1 | John Doe        | CSE    |    1 |
|    2 | Jane Smith      | CSE    |    1 |
|    3 | Michael Johnson | CSE    |    2 |
|    4 | Emily Davis     | CSE    |    2 |
|    5 | David Brown     | CSE    |    3 |
+------+-----------------+--------+------+
```

6)write a query to display only deptname using student table

SELECT DISTINCT d.dept_name FROM students s JOIN department d ON s.dept_id = d.dept_id;

Query OK,1` rows affected(0.01 sec)

```
+--------+
| d_name |
+--------+
| CSE    |
| ECE    |
| Civil  |
| Mech   |
+--------+
```

7)Display students sorted by department and first name:

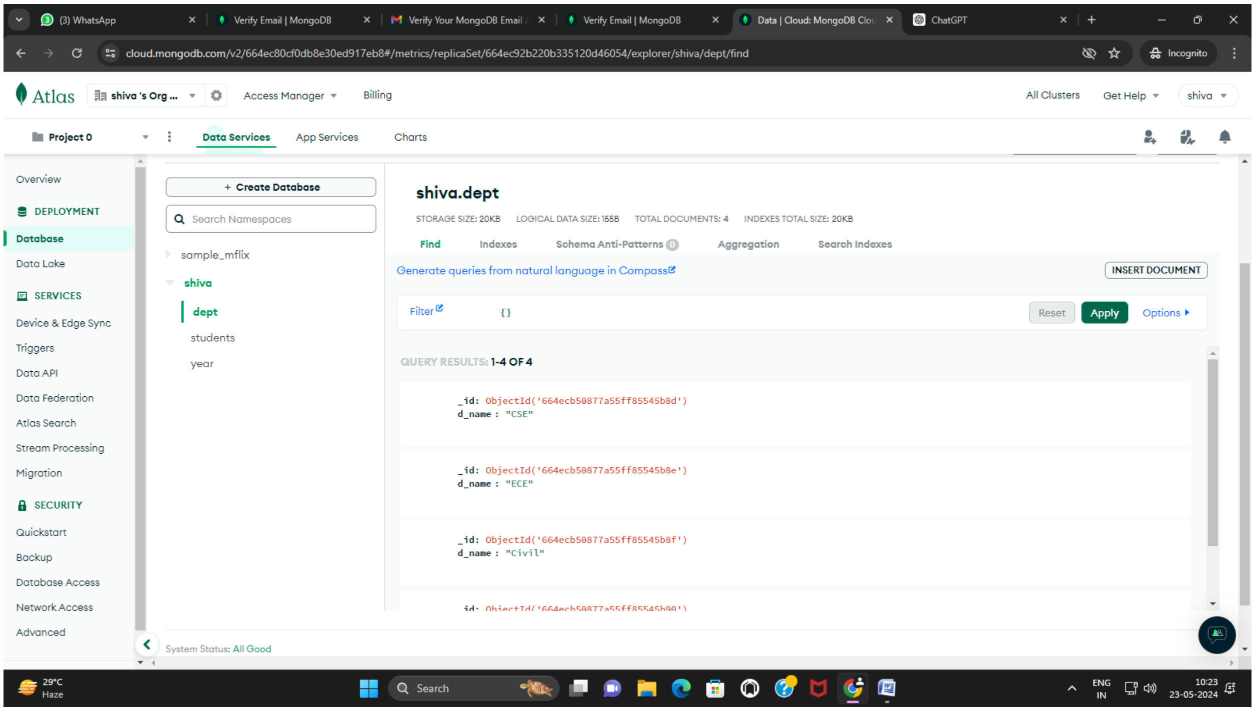
SELECT s.first_name, s.last_name, d.dept_name FROM students s JOIN department d ON s.dept_id = d.dept_id ORDER BY d.dept_name, s.first_name;

Query OK,1` rows affected(0.01 sec)

```
+-----------------+--------+
| s_name         | d_name |
+-----------------+--------+
| John Doe       | CSE    |
| Jane Smith      | CSE    |
| Michael Johnson | CSE    |
| Emily Davis     | CSE    |
| David Brown     | CSE    |
| Sarah Wilson    | ECE    |
| Daniel Martinez | ECE    |
| Jessica Anderson| ECE    |
| Christopher Taylor| ECE  |
| Ashley Thomas   | ECE    |
| Matthew Lee     | Civil  |
| Amanda White    | Civil  |
| Ryan Garcia     | Civil  |
| Brittany Hall   | Civil  |
| Olivia Clark    | Civil  |
| Nicholas Perez  | Mech   |
| Kayla Hernandez | Mech   |
| Justin Young    | Mech   |
| Lauren King     | Mech   |
| Brandon Wright  | Mech   |
+-----------------+--------+
```

MongoDB:

Department:-



cloud.mongodb.com/v2/664ec80cf0db8e30ed917eb8#/metrics/replicaSet/664ec92b220b335120d46054/explorer/shiva/dept/find

shiva.dept

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 155B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 20KB

Filter {}

QUERY RESULTS: 1-4 OF 4

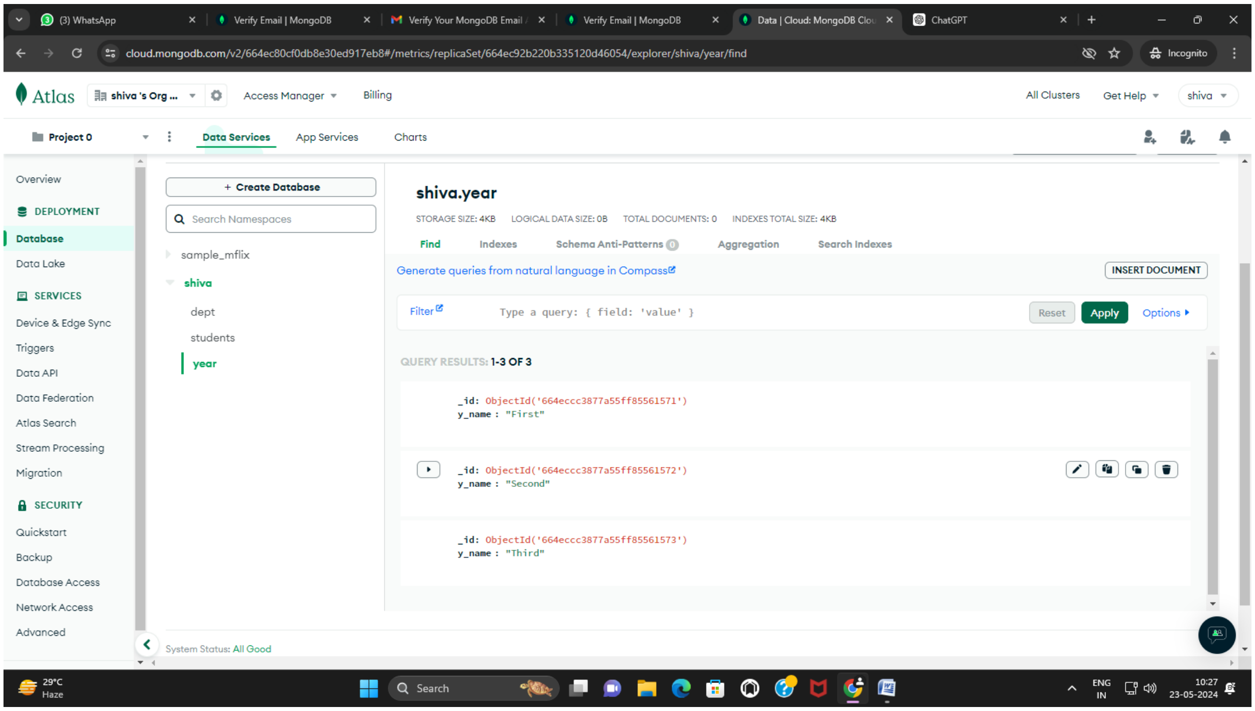
```
_id: ObjectId('664ecb50877a55ff85545b8d')
d_name : "CSE"

_id: ObjectId('664ecb50877a55ff85545b8e')
d_name : "ECE"

_id: ObjectId('664ecb50877a55ff85545b8f')
d_name : "Civil"

_id: ObjectId('664ecb50877a55ff85545b90')
```

Year:-



cloud.mongodb.com/v2/664ec80cf0db8e30ed917eb8#/metrics/replicaSet/664ec92b220b335120d46054/explorer/shiva/year/find

shiva.year

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

Filter Type a query: { field: 'value' }

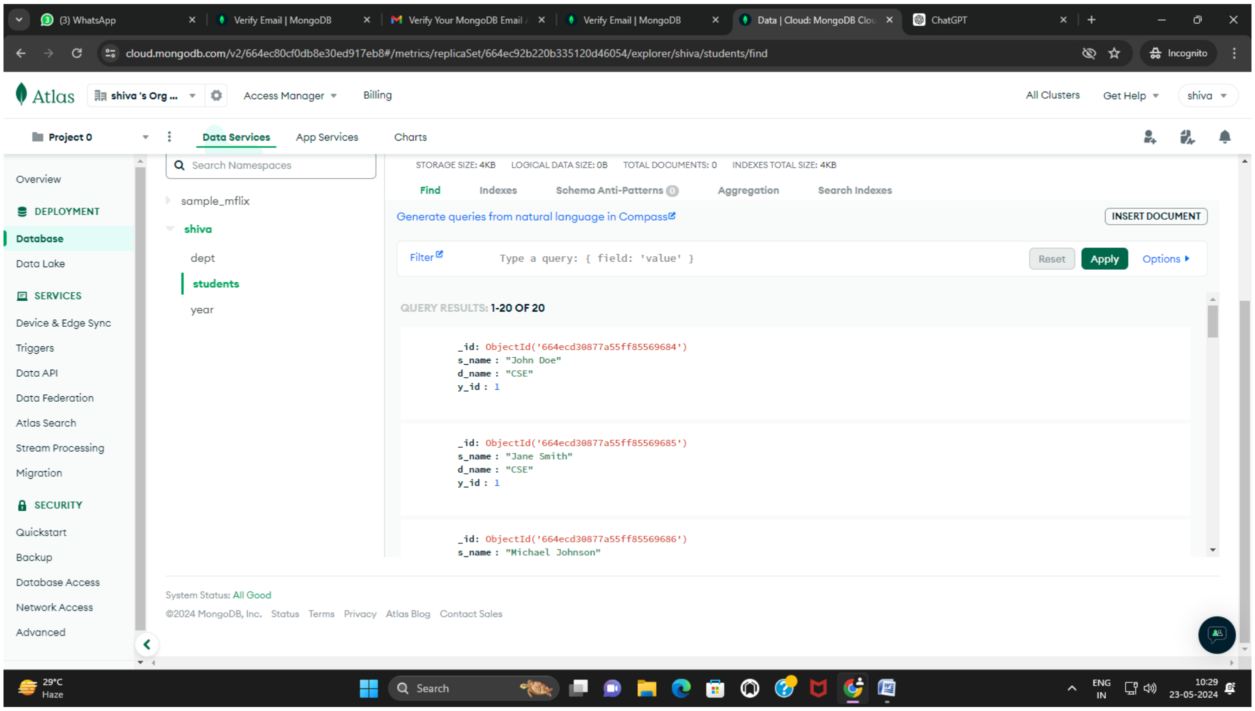
QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('664eccc3877a55ff85561571')
y_name : "First"

_id: ObjectId('664eccc3877a55ff85561572')
y_name : "Second"

_id: ObjectId('664eccc3877a55ff85561573')
y_name : "Third"
```

Student:-



cloud.mongodb.com/v2/664ec80cf0db8e30ed917eb8#/metrics/replicaSet/664ec92b220b335120d46054/explorer/shiva/students/find
Atlas
Data Services
sample_mflix
shiva
dept
students
year
STORAGE SIZE: 4KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
Reset
Apply
Options

Filter `Type a query: { field: 'value' }`

QUERY RESULTS: 1-20 OF 20

```
_id: ObjectId('664ecd30877a55ff85569684')
s_name : "John Doe"
d_name : "CSE"
y_id : 1

_id: ObjectId('664ecd30877a55ff85569685')
s_name : "Jane Smith"
d_name : "CSE"
y_id : 1

_id: ObjectId('664ecd30877a55ff85569686')
s_name : "Michael Johnson"
```