

# **MODERN APPLICATION DEVELOPMENT**

**(JAVA SPRING BOOT)**

**ASSIGNMENT-02**

**NAME: SHARANKANTH.E**

**REGNO: 20MIS0076**

**COLLEGE: VIT VELLORE**

**BRANCH: INTEGRATED M TECH (SOFTWARE  
ENGINEERING)**

**GOOGLE DRIVE**

**LINK:**[https://docs.google.com/document/d/1EP8Te\\_IFSt\\_Gc5miJKQnhpGGDhX4pQGF/edit?usp=share\\_link&oid=112079020877035333650&rtpof=true&sd=true](https://docs.google.com/document/d/1EP8Te_IFSt_Gc5miJKQnhpGGDhX4pQGF/edit?usp=share_link&oid=112079020877035333650&rtpof=true&sd=true)

(i) Create , update, delete commands in MYSQL.

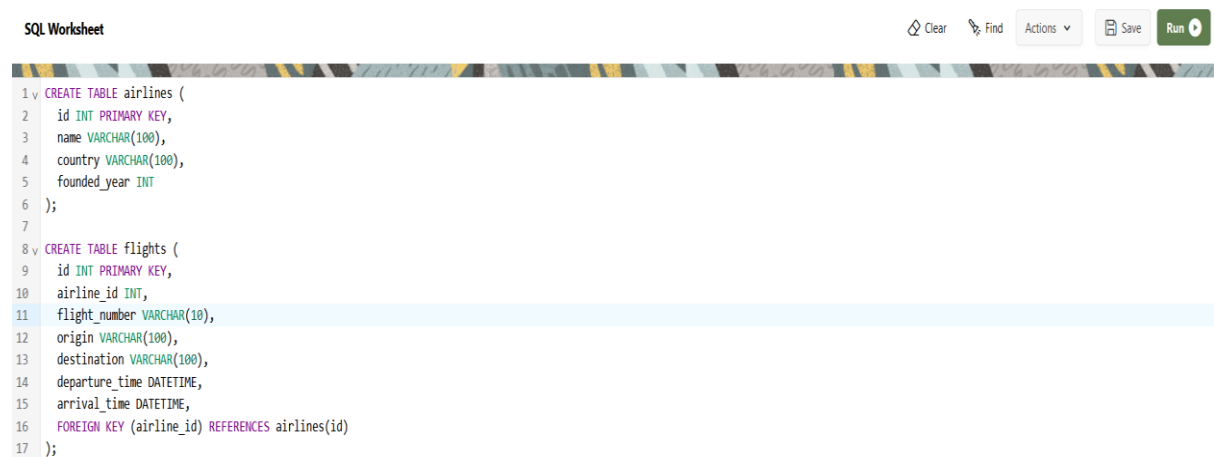
**Create:**

```
CREATE TABLE airlines (  
    id INT PRIMARY KEY,  
    name VARCHAR(100),  
    country VARCHAR(100),  
    founded_year INT  
);
```

```
CREATE TABLE flights (  
    id INT PRIMARY KEY,  
    airline_id INT,  
    flight_number VARCHAR(10),  
    origin VARCHAR(100),  
    destination VARCHAR(100),
```

```
departure_time DATETIME,  
arrival_time DATETIME,  
FOREIGN KEY (airline_id) REFERENCES airlines(id)  
);
```

Screenshots:



The screenshot shows an SQL Worksheet interface. At the top, there's a header bar with 'SQL Worksheet' on the left and 'Clear', 'Find', 'Actions', 'Save', and 'Run' buttons on the right. Below the header, the SQL code is displayed in a monospaced font with syntax highlighting. The code defines two tables: 'airlines' and 'flights'. The 'airlines' table has columns 'id' (INT, PRIMARY KEY), 'name' (VARCHAR(100)), 'country' (VARCHAR(100)), and 'founded\_year' (INT). The 'flights' table has columns 'id' (INT, PRIMARY KEY), 'airline\_id' (INT), 'flight\_number' (VARCHAR(10)), 'origin' (VARCHAR(100)), 'destination' (VARCHAR(100)), 'departure\_time' (DATETIME), 'arrival\_time' (DATETIME), and a foreign key constraint 'FOREIGN KEY (airline\_id) REFERENCES airlines(id)'. The code is numbered from 1 to 17 on the left margin.

```
1 v CREATE TABLE airlines (  
2   id INT PRIMARY KEY,  
3   name VARCHAR(100),  
4   country VARCHAR(100),  
5   founded_year INT  
6 );  
7  
8 v CREATE TABLE flights (  
9   id INT PRIMARY KEY,  
10  airline_id INT,  
11  flight_number VARCHAR(10),  
12  origin VARCHAR(100),  
13  destination VARCHAR(100),  
14  departure_time DATETIME,  
15  arrival_time DATETIME,  
16  FOREIGN KEY (airline_id) REFERENCES airlines(id)  
17 );
```

### Insert command:

```
INSERT INTO airlines (id, name, country, founded_year)  
VALUES (1, 'Airline A', 'Country A', 1990);
```

```
INSERT INTO airlines (id, name, country, founded_year)  
VALUES (2, 'Airline B', 'Country B', 1985);
```

```
INSERT INTO flights (id, airline_id, flight_number,  
origin, destination, departure_time, arrival_time)
```

```
VALUES (1, 1, 'A100', 'City X', 'City Y', '2023-05-28  
10:00:00', '2023-05-28 12:00:00');
```

```
INSERT INTO flights (id, airline_id, flight_number,  
origin, destination, departure_time, arrival_time)
```

```
VALUES (2, 2, 'B200', 'City Y', 'City Z', '2023-05-28  
14:00:00', '2023-05-28 16:00:00');
```

Screenshots:

ID	NAME	COUNTRY	FOUNDED_YEAR
1	Airline A	New Country	1990

Download CSV

### **Update command:**

```
UPDATE airlines
```

```
SET country = 'New Country'
```

```
WHERE id = 1;
```

```
UPDATE flights
```

```
SET flight_number = 'A101'
```

```
WHERE id = 1;
```

## Screenshots:

```
UPDATE airlines
SET country = 'New Country'
WHERE id = 1;

UPDATE flights
SET flight_number = 'A101'
WHERE id = 1;
```

## Delete command:

DELETE FROM airlines

WHERE id = 2;

DELETE FROM flights

WHERE id = 2;

## Screenshots:

```
DELETE FROM airlines
WHERE id = 2;
```

```
DELETE FROM flights
WHERE id = 2;
```

**Q3. Create, update, delete commands in mongoDB.**

**Commands:** here we taken an example for flight/airline database.

**1.insert command:**

**db.airlines.insertOne({**

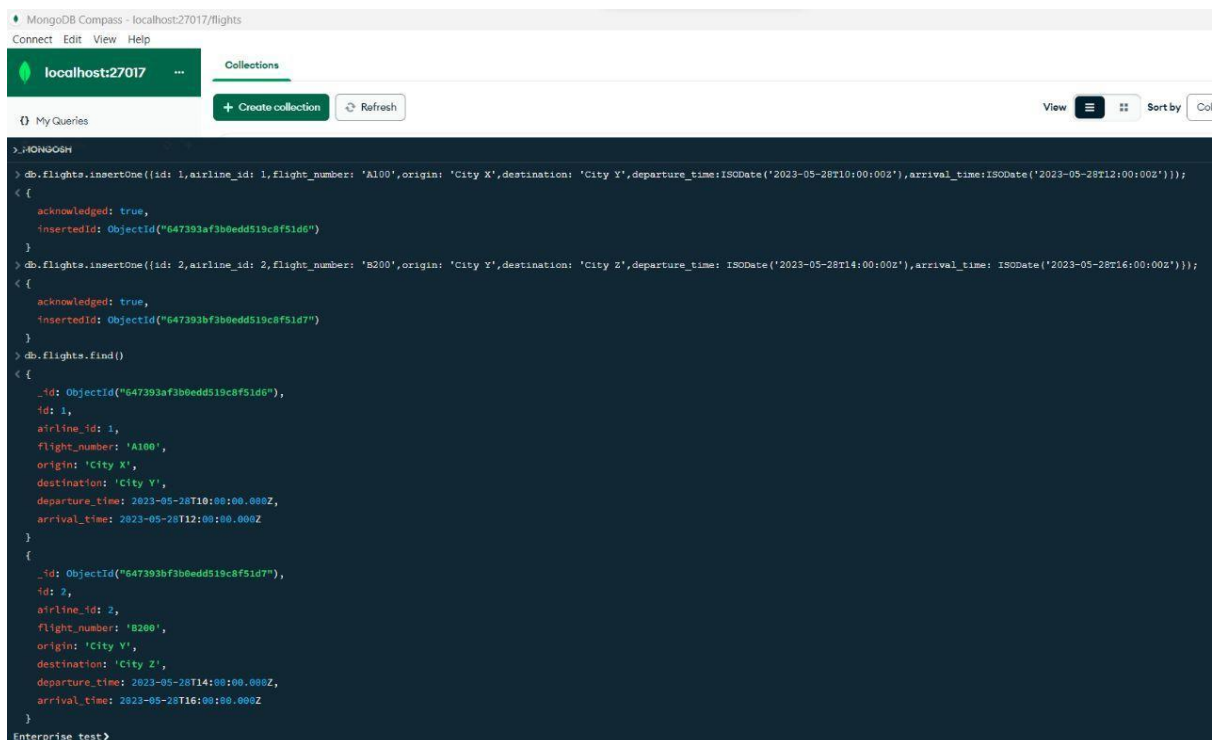
```
id: 1,  
  
name: 'Airline A',  
  
country: 'Country A',  
  
founded_year: 1990  
});
```

```
db.airlines.insertOne({  
  
id: 2,  
  
name: 'Airline B',  
  
country: 'Country B',  
  
founded_year: 1985  
});
```

```
db.flights.insertOne({  
  
id: 1,  
  
airline_id: 1,  
  
flight_number: 'A100',  
  
origin: 'City X',  
  
destination: 'City Y',  
  
departure_time: ISODate('2023-05-28T10:00:00Z'),  
  
arrival_time: ISODate('2023-05-28T12:00:00Z')  
});
```

```
db.flights.insertOne({  
  
  id: 2,  
  
  airline_id: 2,  
  
  flight_number: 'B200',  
  
  origin: 'City Y',  
  
  destination: 'City Z',  
  
  departure_time: ISODate('2023-05-28T14:00:00Z'),  
  
  arrival_time: ISODate('2023-05-28T16:00:00Z')  
  
});
```

## Screenshot:



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the 'flights' collection. The 'My Queries' tab is active, displaying a JavaScript snippet that performs two insertOne operations. The first operation inserts a document with id: 1, airline\_id: 1, flight\_number: 'A100', origin: 'City X', destination: 'City Y', departure\_time: ISODate('2023-05-28T10:00:00Z'), and arrival\_time: ISODate('2023-05-28T12:00:00Z'). The second operation inserts a document with id: 2, airline\_id: 2, flight\_number: 'B200', origin: 'City Y', destination: 'City Z', departure\_time: ISODate('2023-05-28T14:00:00Z'), and arrival\_time: ISODate('2023-05-28T16:00:00Z'). Below the code, the output of the find() operation is shown, displaying two documents in a JSON-like format. The first document corresponds to the first insert operation, and the second document corresponds to the second insert operation. The interface also includes a 'Create collection' button and a 'Refresh' button.

```
> use MONGODB  
> db.flights.insertOne({id: 1,airline_id: 1,flight_number: 'A100',origin: 'City X',destination: 'City Y',departure_time:ISODate('2023-05-28T10:00:00Z'),arrival_time:ISODate('2023-05-28T12:00:00Z')});  
< {  
  acknowledged: true,  
  insertedId: ObjectId("647393af3b0edd519c8f51d6")  
}  
> db.flights.insertOne({id: 2,airline_id: 2,flight_number: 'B200',origin: 'City Y',destination: 'City Z',departure_time: ISODate('2023-05-28T14:00:00Z'),arrival_time: ISODate('2023-05-28T16:00:00Z')});  
< {  
  acknowledged: true,  
  insertedId: ObjectId("647393bf3b0edd519c8f51d7")  
}  
> db.flights.find()  
< {  
  _id: ObjectId("647393af3b0edd519c8f51d6"),  
  id: 1,  
  airline_id: 1,  
  flight_number: 'A100',  
  origin: 'City X',  
  destination: 'City Y',  
  departure_time: 2023-05-28T10:00:00.000Z,  
  arrival_time: 2023-05-28T12:00:00.000Z  
}  
  {  
    _id: ObjectId("647393bf3b0edd519c8f51d7"),  
    id: 2,  
    airline_id: 2,  
    flight_number: 'B200',  
    origin: 'City Y',  
    destination: 'City Z',  
    departure_time: 2023-05-28T14:00:00.000Z,  
    arrival_time: 2023-05-28T16:00:00.000Z  
  }  
}
```

MongoDB Compass - localhost:27017/airlines.airlines

Connect Edit View Collection Help

localhost:27017 Documents airlines.airlines +

My Queries Databases

### airlines.airlines

Documents Associations Schema Explain Plan Indexes

```
>_MONGOSH
> db.airlines.insertOne({id: 1,name: 'Airline A',country: 'Country A',founded_year: 1990});
< {
  acknowledged: true,
  insertedId: ObjectId("6473928fe2584628ff44b7bf")
}
> db.airlines.insertOne({id: 2,name: 'Airline B',country: 'Country B',founded_year: 1985});
< {
  acknowledged: true,
  insertedId: ObjectId("6473929ce2584628ff44b7c0")
}
> db.airlines.find()
< {
  _id: ObjectId("6473928fe2584628ff44b7bf"),
  id: 1,
  name: 'Airline A',
  country: 'Country A',
  founded_year: 1990
}
{
  _id: ObjectId("6473929ce2584628ff44b7c0"),
  id: 2,
  name: 'Airline B',
  country: 'Country B',
  founded_year: 1985
}
Enterprise test>
```



## Update command:

MongoDB Compass - localhost:27017/flights

Connect Edit View Help

localhost:27017 ...

My Queries

Databases

Search

JAVASPRINGBOOT

admin

Collections

+ Create collection Refresh

flights

Storage size: 4.10 kB Documents: 0

```
> _MONGOSH
}
> db.flights.updateOne( { id: 1 }, { $set: { flight_number: 'A101' } });
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.airlines.find()
< {
  _id: ObjectId("6473928fe2584628ff44b7bf"),
  id: 1,
  name: 'Airline A',
  country: 'New Country',
  founded_year: 1990
}
Enterprise test>
```

MongoDB Compass - localhost:27017/airlines.airlines

Connect Edit View Collection Help

localhost:27017 Documents airlines.airlines +

My Queries

Databases

Search

JAVASPRINGBOOT

admin

airlines

airlines.airlines

Documents Aggregations Schema Explain Plan Indexes Validation

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

```
>_MONGOSH
}
> db.airlines.updateOne({ id: 1 }, { $set: { country: 'New Country' } });
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.airlines.find()
< {
  _id: ObjectId("6473928fe2584628ff44b7bf"),
  id: 1,
  name: 'Airline A',
  country: 'New Country',
  founded_year: 1990
}
{
  _id: ObjectId("6473929ce2584628ff44b7c0"),
  id: 2,
  name: 'Airline B',
  country: 'Country B',
  founded_year: 1985
}
Enterprise test>
```

## Delete command:

MongoDB Compass - localhost:27017/flights

Connect Edit View Help

localhost:27017 ...

My Queries

Databases

Search

JAVASPRINGBOOT

admin

Collections

+ Create collection Refresh

flights

Storage size: 4.10 kB Documents: 0

```
>_MONGOSH
}
> db.flights.deleteOne({ id: 2 });
< {
  acknowledged: true,
  deletedCount: 1
}
> db.airlines.find()
< {
  _id: ObjectId("6473928fe2584628ff44b7bf"),
  id: 1,
  name: 'Airline A',
  country: 'New Country',
  founded_year: 1990
}
Enterprise test>
```

MongoDB Compass - localhost:27017/airlines.airlines

Connect Edit View Collection Help

localhost:27017 Documents airlines.airlines +

My Queries

Databases

Search

- JAVASPRINGBOOT
- admin
- airlines
  - airlines
  - config
  - local

airlines.airlines

Documents Aggregations Schema Explain Plan Indexes Validation

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

+ ADD DATA EXPORT DATA

```
>_MONGOSH
}
> db.airlines.deleteOne({ id: 2 });
< {
  acknowledged: true,
  deletedCount: 1
}
> db.airlines.find()
< {
  _id: ObjectId("6473928fe2584628ff44b7bf"),
  id: 1,
  name: 'Airline A',
  country: 'New Country',
  founded_year: 1990
}
Enterprise test>
```