

# **Music Player Application**

**Milestone:** Implementation in NoSQL

- ETIKALA SAI MAHITHA

The screenshots have all the tables imported into MongoDB:

## Album

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'AlbumTable'. The table contains 50 documents. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a list of album documents with fields like \_id, \_idempotency\_token, title, artist, and length.

_id	_idempotency_token	title	artist	length
1	1	"Sweet 1913"	"Sweet 1913"	3
2	2	"Love Song"	"Love Song"	3
3	3	"Best of 80s"	"Best of 80s"	3
4	4	"Classic Rock Anthems"	"Classic Rock Anthems"	3
5	5	"Acoustic Favorites"	"Acoustic Favorites"	3

## Artist

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'ArtistTable'. The table contains 50 documents. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a list of artist documents with fields like \_id, \_idempotency\_token, name, and birth\_date.

_id	_idempotency_token	name	birth_date
1	1	"Artist 1"	1913-01-01
2	2	"Artist 2"	1913-01-01
3	3	"Artist 3"	1913-01-01
4	4	"Artist 4"	1913-01-01
5	5	"Artist 5"	1913-01-01

## Company

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'Company' table. The table contains 1 document. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a single company document with fields like \_id, \_idempotency\_token, name, address, and website.

_id	_idempotency_token	name	address	website
1	1	"Company Name"	"123 Main Street, City, State, Country"	"http://www.example.com"

## Creates

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'CreateTable'. The table contains 450 documents. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a list of create documents with fields like \_id, \_idempotency\_token, playlist\_id, and song\_id.

_id	_idempotency_token	playlist_id	song_id
1	1	1	1
2	2	1	1
3	3	1	1
4	4	1	1
5	5	1	1

## Developer

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'DeveloperTable'. The table contains 87 documents. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a list of developer documents with fields like \_id, \_idempotency\_token, name, birth\_date, and role.

_id	_idempotency_token	name	birth_date	role
1	1	"Developer 1"	"1913-01-01"	"Software Engineer"
2	2	"Developer 2"	"1913-01-01"	"Software Engineer"
3	3	"Developer 3"	"1913-01-01"	"Software Engineer"
4	4	"Developer 4"	"1913-01-01"	"Software Engineer"
5	5	"Developer 5"	"1913-01-01"	"Software Engineer"

## Develops

The screenshot shows the MongoDB Compass interface for the 'MusicPlayerSystem' database, specifically the 'DevelopsTable'. The table contains 87 documents. The left sidebar shows a list of tables including AlbumTable, ArtistTable, Company, CreateTable, DeveloperTable, DevelopTable, LikeTable, MusicPlayerSystem, PersonTable, PlaylistTable, ProducedSongs, SongTable, UserPlayer, UserTable, admin, config, and local. The main area displays a list of develops documents with fields like \_id, \_idempotency\_token, developer\_id, and album\_id.

_id	_idempotency_token	developer_id	album_id
1	1	1	1
2	2	1	1
3	3	1	1
4	4	1	1
5	5	1	1

## Playlist

## Person

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 500

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Playlist	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Playlist	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Playlist	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Playlist	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Playlist	5

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 500

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Person	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Person	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Person	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Person	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Person	5

## MusicPlayerSystem

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 3,000

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	MusicPlayerSystem	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	MusicPlayerSystem	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	MusicPlayerSystem	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	MusicPlayerSystem	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	MusicPlayerSystem	5

## Likes

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 84

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Likes	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Likes	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Likes	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Likes	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Likes	5

## Produces

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 250

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	ProducesSongs	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	ProducesSongs	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	ProducesSongs	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	ProducesSongs	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	ProducesSongs	5

## Songs

localhost:27017 Documents MusicPlayerSystem

My Queries Databases

MusicPlayerSystem

AlbumTable ArtistTable Company CreateTable DeveloperTable DeveloperTable LiveTable MusicPlayerSystem PlaylistTable ProducersTable SongTable UserPlayer UserTable

Filter: Type a query (Field: 'value') or **Generate Query**

1-20 of 250

_id	_source	_type	_value
1	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Songs	1
2	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Songs	2
3	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Songs	3
4	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Songs	4
5	{ "name": "The Beatles", "year": 1963, "genre": "Rock", "length": 2:30 }	Songs	5

## UserPlayer

## User

localhost:27017

Documents MusicPlayerSystem...

My Databases

MusicPlayerSystem.UseTable

363 1

DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Search

MusicPlayerSystem

AlbumTable

ArtistTable

Company

CreateTable

DeveloperTable

DesktopTable

UseTable

MusicPlayerSystem

PersonTable

PlaylistTable

ProducedSongs

SongTable

UserPlayer

UserTable

admin

config

local

Type a query: ( ? ) value: 'value' or generate\_query

Execute Reset Find + Options +

ADD DATA - EXPORT DATA

1 - 25 of 363

```

{
  "id": 1,
  "name": "MusicPlayerSystem",
  "description": "A music player system that allows users to play music from a database.",
  "version": "1.0.0",
  "author": "John Doe",
  "created": "2023-01-01",
  "updated": "2023-01-01",
  "status": "active"
}

```

```

{
  "id": 2,
  "name": "MusicPlayerSystem",
  "description": "A music player system that allows users to play music from a database.",
  "version": "1.0.0",
  "author": "John Doe",
  "created": "2023-01-01",
  "updated": "2023-01-01",
  "status": "active"
}

```

```

{
  "id": 3,
  "name": "MusicPlayerSystem",
  "description": "A music player system that allows users to play music from a database.",
  "version": "1.0.0",
  "author": "John Doe",
  "created": "2023-01-01",
  "updated": "2023-01-01",
  "status": "active"
}

```

```

{
  "id": 4,
  "name": "MusicPlayerSystem",
  "description": "A music player system that allows users to play music from a database.",
  "version": "1.0.0",
  "author": "John Doe",
  "created": "2023-01-01",
  "updated": "2023-01-01",
  "status": "active"
}

```

```

{
  "id": 5,
  "name": "MusicPlayerSystem",
  "description": "A music player system that allows users to play music from a database.",
  "version": "1.0.0",
  "author": "John Doe",
  "created": "2023-01-01",
  "updated": "2023-01-01",
  "status": "active"
}

```

## NoSQL Queries

1) This query filtered the Album details with the AlbumName "Indie Showcase"

```
db.AlbumTable.find({  
  AlbumName: 'Indie Showcase'  
})
```

```
> db.AlbumTable.find({  
  AlbumName: 'Indie Showcase'  
})  
< {  
  _id: ObjectId("656d4755a5ce939e3d699401"),  
  AlbumID: 8,  
  AlbumName: 'Indie Showcase',  
  SongID: 33  
}  
MusicPlayerSystem> |
```

2) This query filters out all the males whose age is between 23 and 35

```
db.PersonTable.find({  
  Gender: 'Male',  
  Age: {  
    $gte: 23,  
    $lte: 35  
  }  
});
```

```

>_MONGOSH
>
> db.PersonTable.find({
  Gender: 'Male',
  Age: {
    $gte: 23,
    $lte: 35
  }
});
< {
  _id: ObjectId("656d4c24a5ce939e3d699732"),
  id: 3,
  PID: 3,
  Name: 'Blossom Evans',
  Gender: 'Male',
  Age: 23,
  phone: '1-995-746-8709',
  address: 'Ap #263-9754 Dui. Ave'
}
{
  _id: ObjectId("656d4c24a5ce939e3d69973c"),
  id: 13,
  PID: 13,
  Name: 'Aike Norman',
  Gender: 'Male',
  Age: 26,
  phone: '1-141-670-3418',
  address: 'Ap #134-463 Sem, Road'
}
{
  _id: ObjectId("656d4c24a5ce939e3d69973d"),
  id: 14,
  PID: 14,
  Name: 'Cedric Powell',
  Gender: 'Male',
  Age: 30,
  phone: '1-642-783-0153',
}

```

3) This query gives the song Title and number of likes it got

```

db.LikesTable.aggregate([
  {
    $lookup: { from:
      "SongsTable",
      localField:
        "SongID",
      foreignField: "ID",
      as: "songData"
    }
  },
  {
    $group: { _id:
      "$SongID", count:
        { $sum: 1 },
      songDetails: { $first: "$songData" }
    }
  },
  {
    $project: {
      _id: 0,

```

SongName: "\$songDetails.Title", // Replace with the actual field name in SongsTable  
Count: "\$count"

```
}  
}  
]);
```

```
>.MONGOOSH  
> db.LikesTable.aggregate([  
  {  
    $lookup: {  
      from: "SongsTable",  
      localField: "SongID",  
      foreignField: "ID",  
      as: "songData"  
    }  
  },  
  {  
    $group: {  
      _id: "$SongID",  
      count: { $sum: 1 },  
      songDetails: { $first: "$songData" }  
    }  
  },  
  {  
    $project: {  
      _id: 0,  
      SongName: "$songDetails.Title", // Replace with the actual field name in SongsTable  
      Count: "$count"  
    }  
  }  
]);  
< {  
  SongName: [  
    'Unchained Melody'  
  ],  
  Count: 1  
}  
{  
  SongName: [  
    'My Way'  
  ],  
  Count: 1  
}
```

#### 4) This query combines CreatesTable with PlaylistTable and UserTable

```
db.CreatesTable.aggregate([  
  {  
    $lookup: { from:  
      "PlaylistTable",  
      localField: "PlaylistID",  
      foreignField:  
      "PlaylistID", as:  
      "playlistData"  
    }  
  },  
  {  
    $lookup: { from:  
      "UserTable",  
      localField:  
      "UserID",
```

```

        foreignField: "UID",
        as: "userData"
    }
}
]);

```

```

>_MONGOSH
> db.CreatesTable.aggregate([
  {
    $lookup: {
      from: "PlaylistTable",
      localField: "PlaylistID",
      foreignField: "PlaylistID",
      as: "playlistData"
    }
  },
  {
    $lookup: {
      from: "UserTable",
      localField: "UserID",
      foreignField: "UID",
      as: "userData"
    }
  }
]);
< {
  _id: ObjectId("656d48eca5ce939e3d699466"),
  PlaylistID: 39,
  UserID: 11,
  playlistData: [
    {
      _id: ObjectId("656d4c3fa5ce939e3d69994b"),
      PlaylistID: 39,
      Name: 'Pop Divas Anthems',
      Duration: '03:10:00',
      CreatorID: 237
    }
  ],
  userData: [
    {
      _id: ObjectId("656d4caca5ce939e3d699e86"),
      Name: 'Pop Divas Anthems',
      Duration: '03:10:00',
      CreatorID: 237
    }
  ]
}

```



```

>_MONGOSH
],
  userData: [
    {
      _id: ObjectId("656d4caca5ce939e3d699e86"),
      id: 11,
      UID: 11,
      email_ID: 'fermentum.vel@outlook.ca',
      PID: 98
    }
  ]
}
{
  _id: ObjectId("656d48eca5ce939e3d699467"),
  PlaylistID: 89,
  UserID: 11,
  playlistData: [
    {
      _id: ObjectId("656d4c3fa5ce939e3d69997d"),
      PlaylistID: 89,
      Name: 'Funky Disco Grooves',
      Duration: '02:40:00',
      CreatorID: 166
    }
  ]
},
  userData: [
    {
      _id: ObjectId("656d4caca5ce939e3d699e86"),
      id: 11,
      UID: 11,
      email_ID: 'fermentum.vel@outlook.ca',
      PID: 98
    }
  ]
}
}
{
  _id: ObjectId("656d48eca5ce939e3d699468"),

```

5) This query gives us Name, Gender, Age, Phone, Address of users from age 21 and 35 in ascending order and groups people of same age

```

db.PersonTable.aggregate([
  {
    $match: {
      Age: { $gte: 21, $lte: 35 }
    }
  },
  {
    $lookup: { from:
      "UserTable",
      localField: "PID",
      foreignField:
        "PID", as:
        "userData"
    }
  },
  {
    $group: { _id: "$Age", persons: {
      $push: "$$ROOT" }, userData:
      { $first: "$userData" }
    }
  }
])

```

```
},
{
  $project: {
    _id: 0,
    "persons.Name": 1,
    "persons.Gender": 1,
    "persons.Age": 1,
    "persons.phone": 1,
    "persons.address": 1
  }
},
{
  $sort: {
    "_id": 1
  }
}
]);
```

```

>_MONGOSH
> db.PersonTable.aggregate([
  {
    $match: {
      Age: { $gte: 21, $lte: 35 }
    }
  },
  {
    $lookup: {
      from: "UserTable",
      localField: "PID",
      foreignField: "PID",
      as: "userData"
    }
  },
  {
    $group: {
      _id: "$Age",
      persons: { $push: "$$ROOT" },
      userData: { $first: "$userData" }
    }
  },
  {
    $project: {
      _id: 0,
      "persons.Name": 1,
      "persons.Gender": 1,
      "persons.Age": 1,
      "persons.phone": 1,
      "persons.address": 1
    }
  },
  {
    $sort: {
      "_id": 1
    }
  }
]);
< {
  persons: [
    {
      Name: 'Blossom Evans',
      Gender: 'Male',
      Age: 23,

```

```

      Age: 23,
      phone: '1-995-746-8769',
      address: 'Ap #263-9754 Dui. Ave'
    },
    {
      Name: 'Valentine Henderson',
      Gender: 'Female',
      Age: 23,
      phone: '1-316-695-5560',
      address: '5702 In Road'
    },
    {
      Name: 'Tamara Holland',
      Gender: 'Male',
      Age: 23,
      phone: '1-206-184-4431',
      address: 'P.O. Box 620, 1990 Orci Avenue'
    },
    {
      Name: 'Astra Murray',
      Gender: 'Male',
      Age: 23,
      phone: '1-561-635-6589',
      address: '290-2880 Aliquam St.'
    },
    {
      Name: 'Kyle Simpson',
      Gender: 'Female',
      Age: 23,
      phone: '1-412-931-2767',
      address: '214-2298 Quisque Avenue'
    },
    {
      Name: 'Brent Cruz',
      Gender: 'Female',
      Age: 23,
      phone: '1-627-338-3513',
      address: '5850 Non St.'
    }
  ]
}
{
  persons: [
    /

```