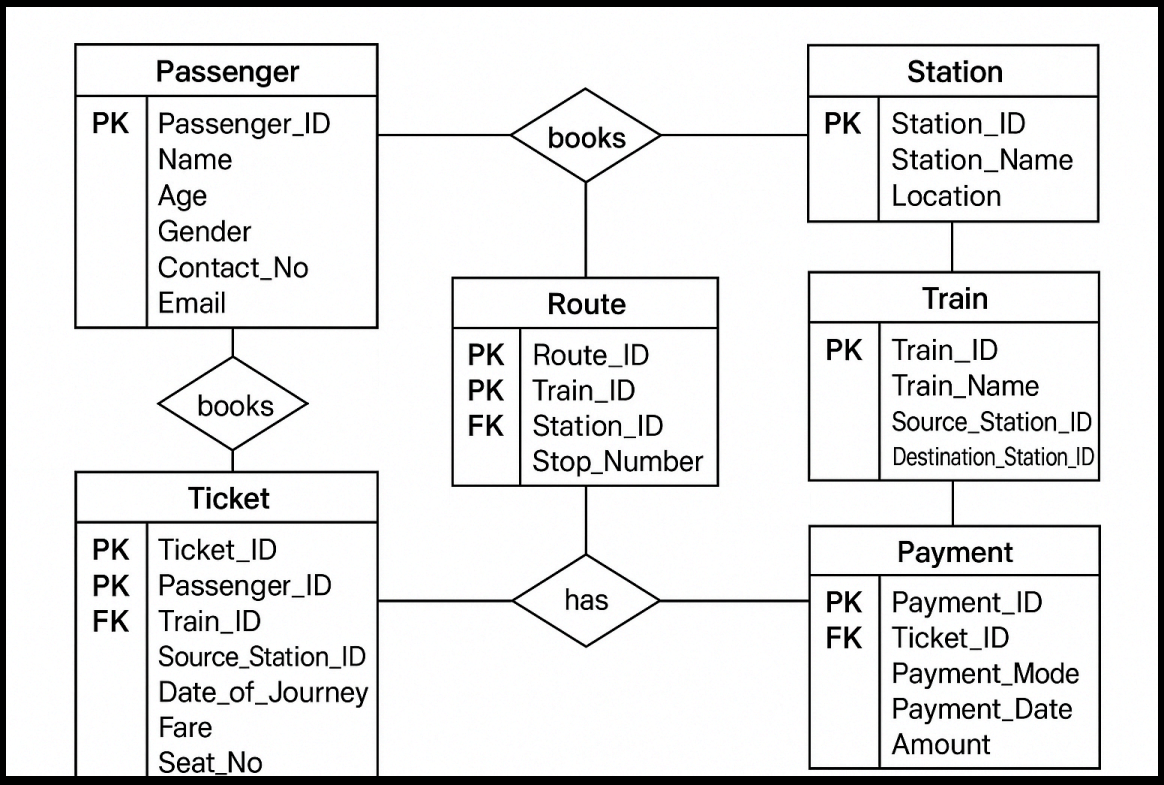
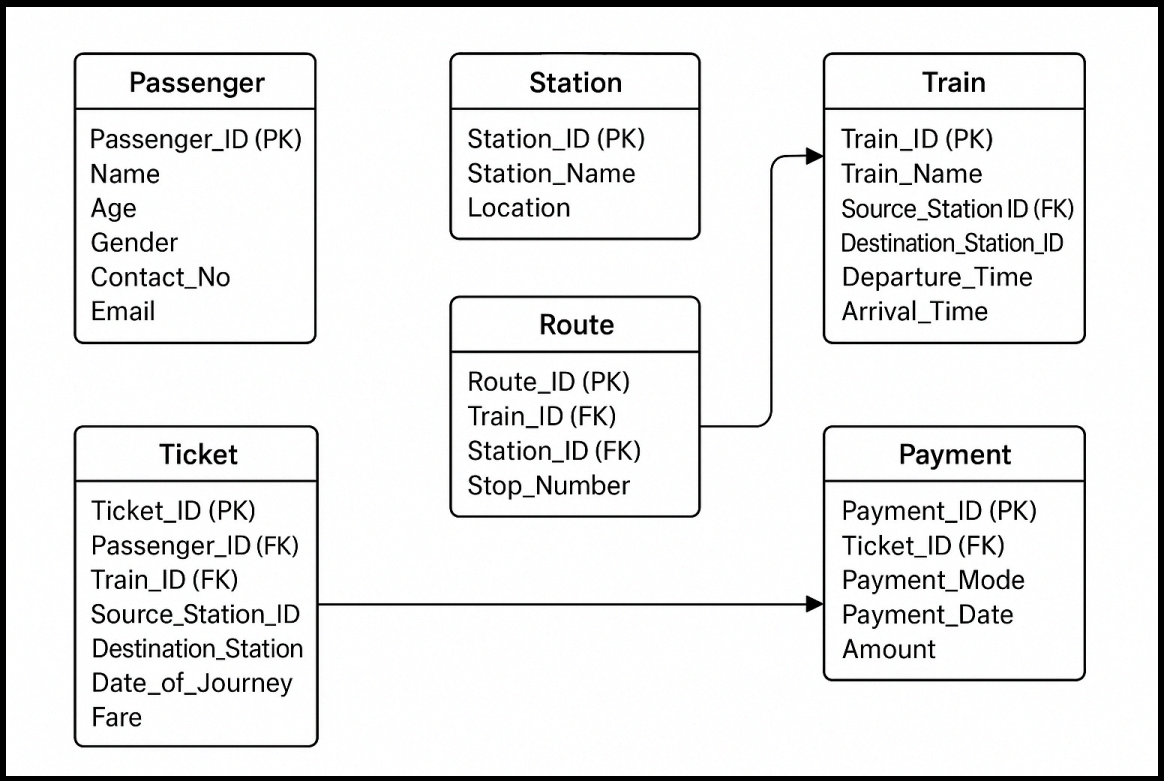
VTR UGE2021- (CBCS)

DBMS TASKS REPORT


Passenger
PK
Passenger_ID
Name
Age
Gender
Contact_No
Email
books
Station
PK
Station_ID
Station_Name
Location
Route
PK
Route_ID
PK
Train_ID
FK
Station_ID
Stop_Number
Train
PK
Train_ID
Train_Name
Source_Station_ID
Destination_Station_ID
books
Ticket
PK
Ticket_ID
PK
Passenger_ID
FK
Train_ID
Source_Station_ID
Date_of_Journey
Fare
Seat_No
has
Payment
PK
Payment_ID
FK
Ticket_ID
Payment_Mode
Payment_Date
Amount


Passenger
Passenger_ID (PK)
Name
Age
Gender
Contact_No
Email
Station
Station_ID (PK)
Station_Name
Location
Train
Train_ID (PK)
Train_Name
Source_Station ID (FK)
Destination_Station_ID
Departure_Time
Arrival_Time
Route
Route_ID (PK)
Train_ID (FK)
Station_ID (FK)
Stop_Number
Ticket
Ticket_ID (PK)
Passenger_ID (FK)
Train_ID (FK)
Source_Station_ID
Destination_Station
Date_of_Journey
Fare
Payment
Payment_ID (PK)
Ticket_ID (FK)
Payment_Mode
Payment_Date
Amount

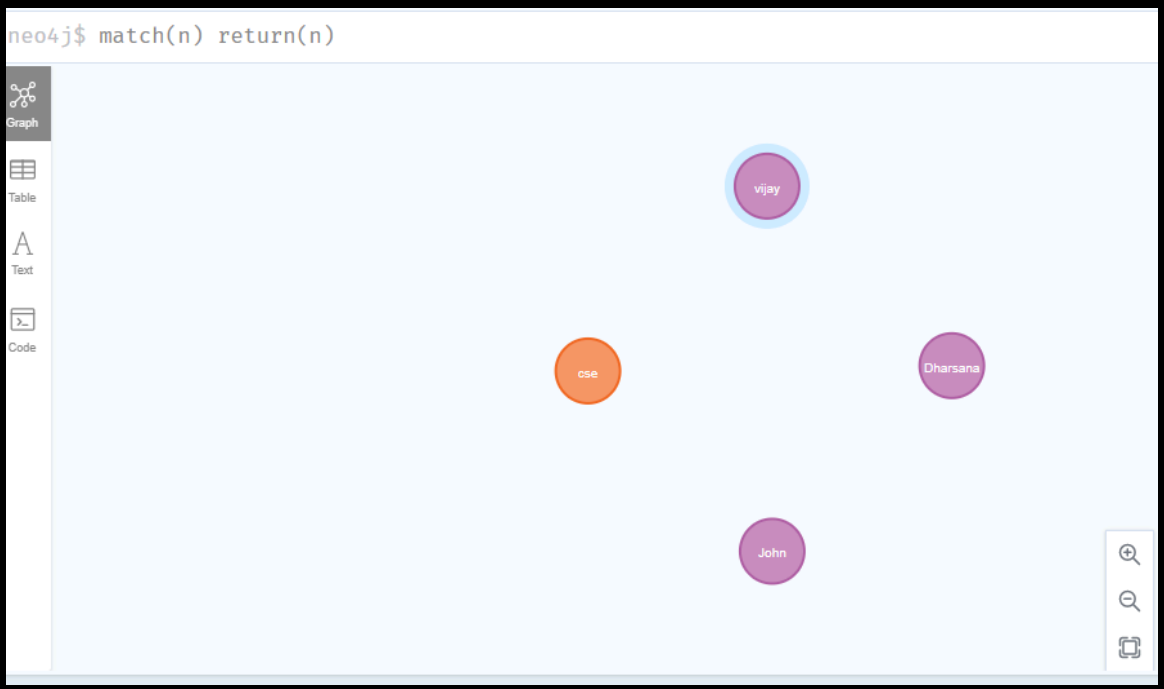
Avg Salary	Max Salary	Min Salary

Department	Avg Salary	Max Salary	Min Salary

Emp Name	Hire Date

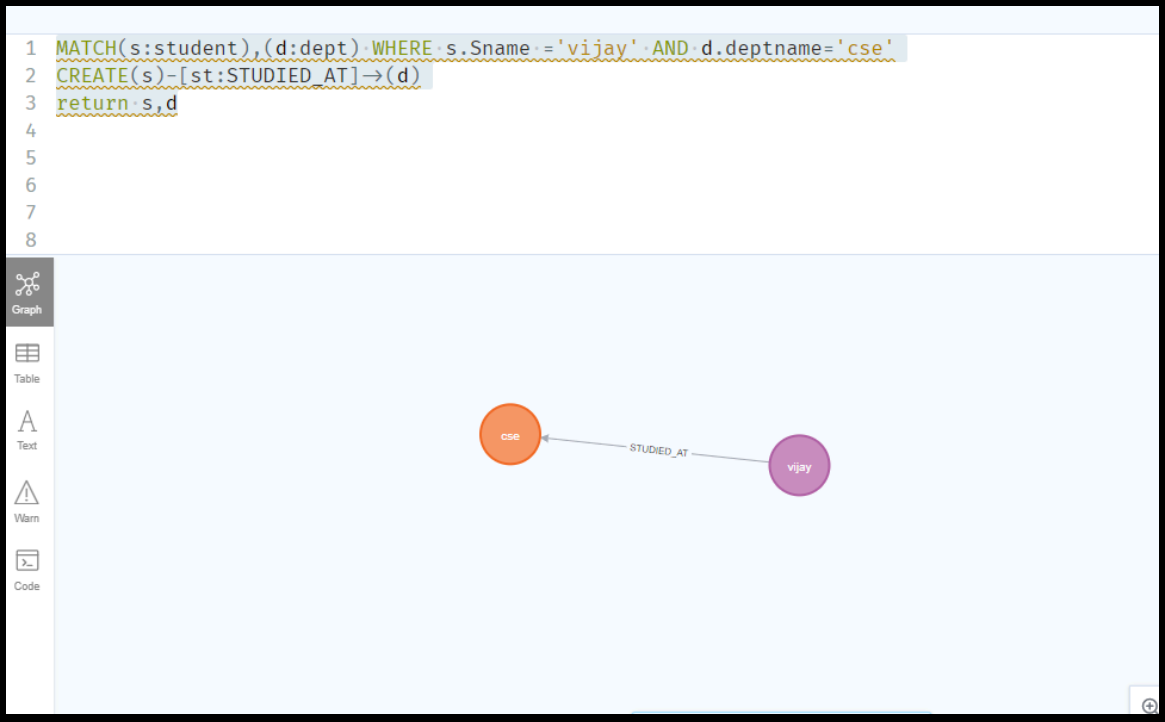
Name	Course	Credits
Ajay	CSE	3

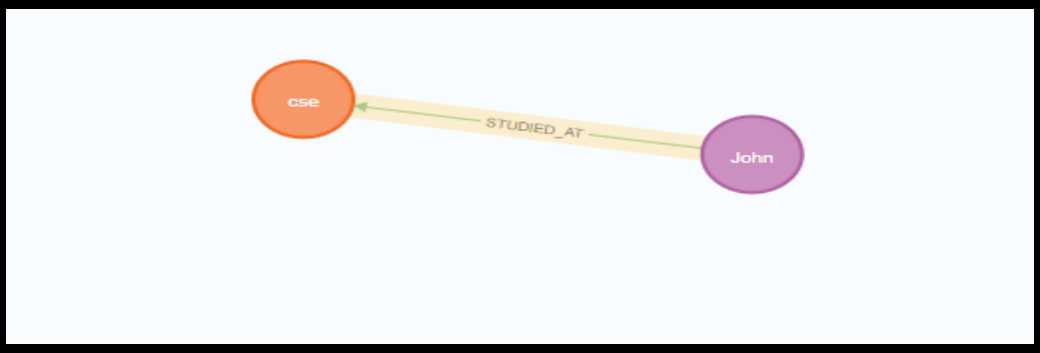
Manoj	ECE	4
Shashi	CSE	3
Pandhu	CSE	3
Sai	CSE	3

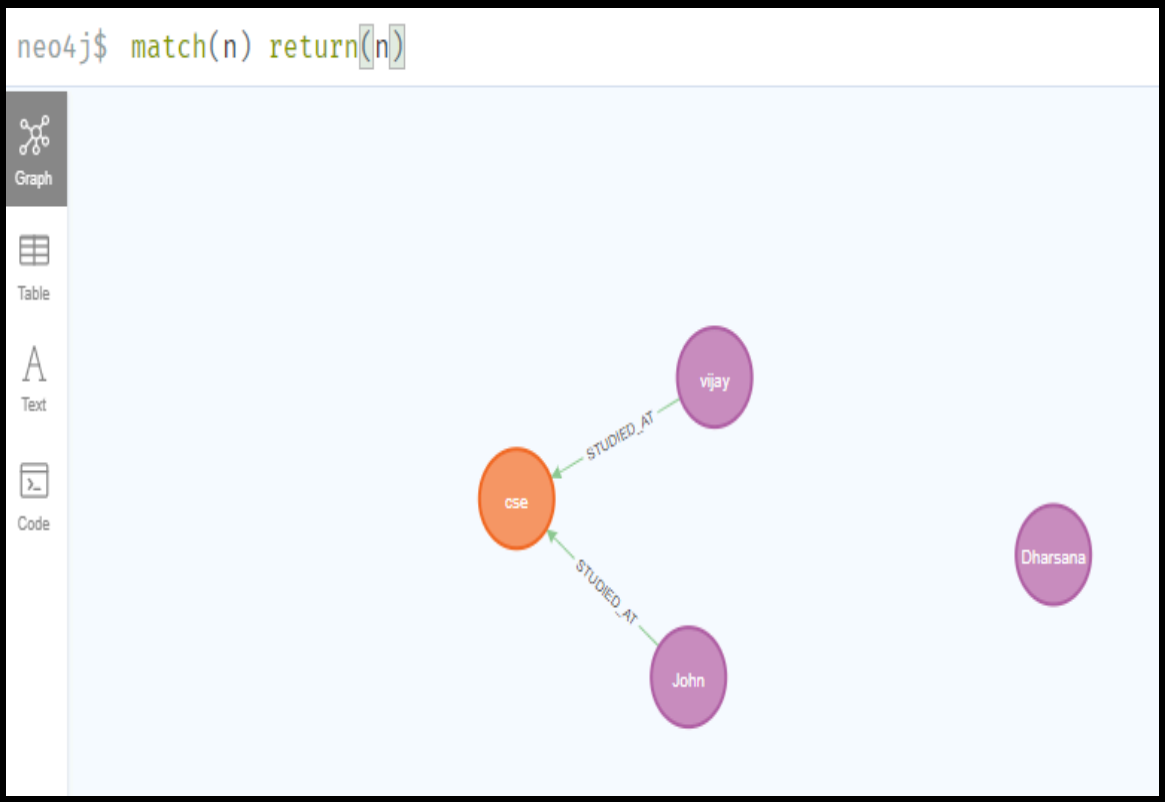

neo4j$ match(n) return(n)
Graph
Table
Text
Code
vijay
cse
Dharsana
John

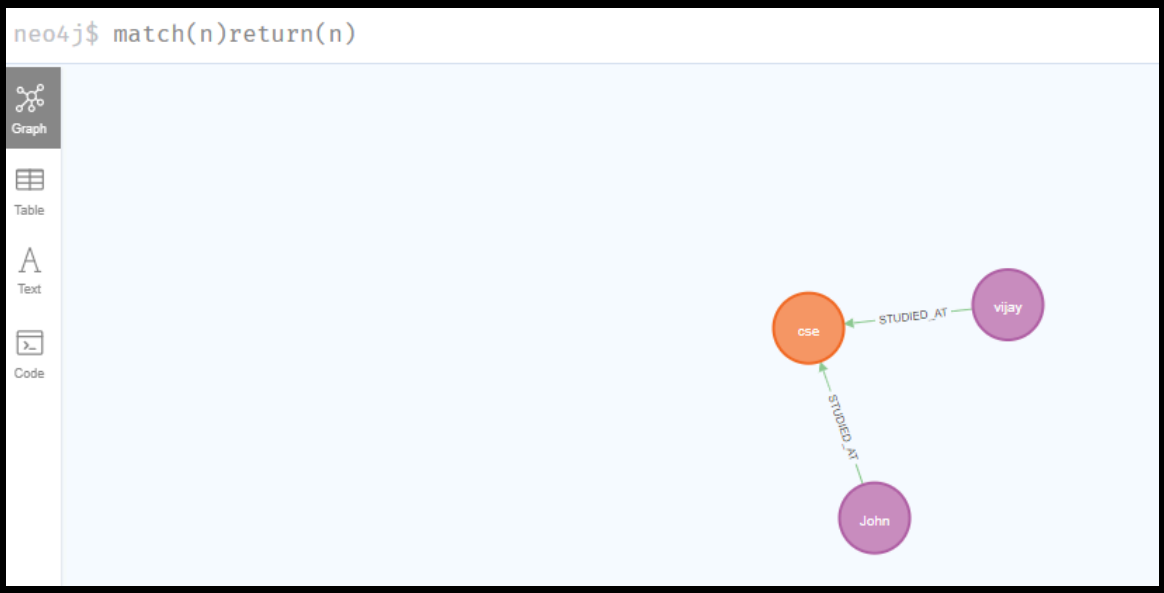

neo4j$ match(n:student) return(n)
Graph
Table
Text
Code
vijay
Dharsana
John

```
MATCH(s:student),(d:dept) WHERE s.Sname ='vijay' AND d.deptname='cse'
CREATE(s)-[st:STUDIED_AT]→(d)
return s,d
```

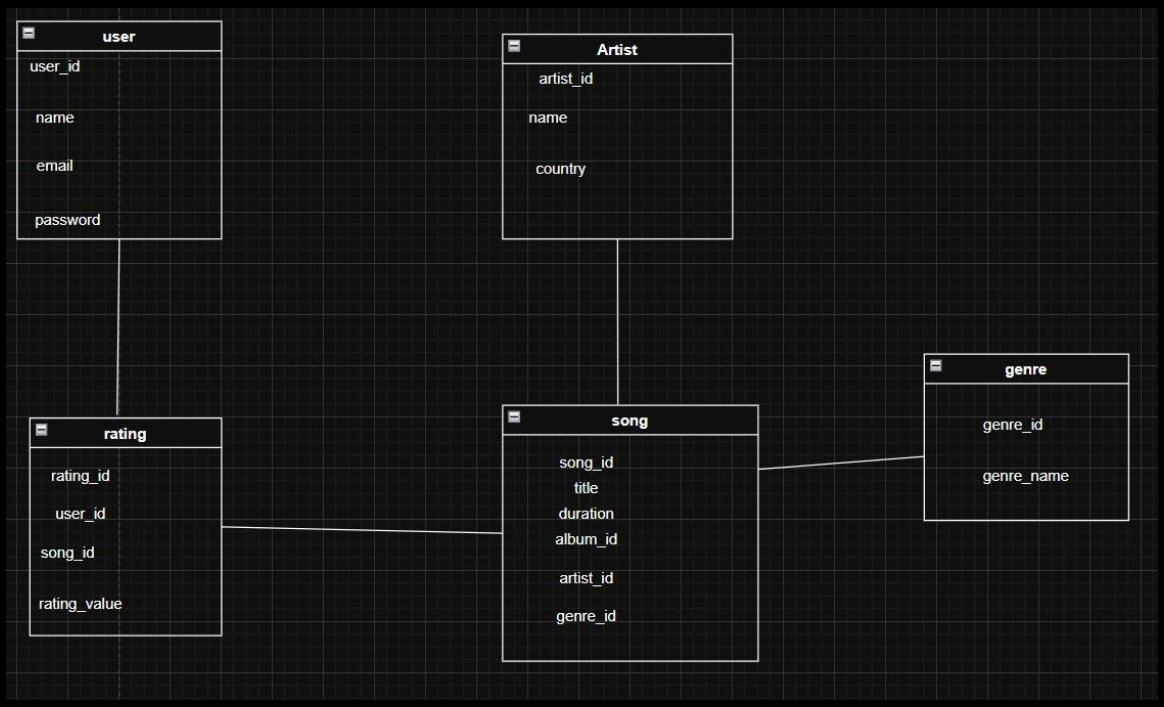


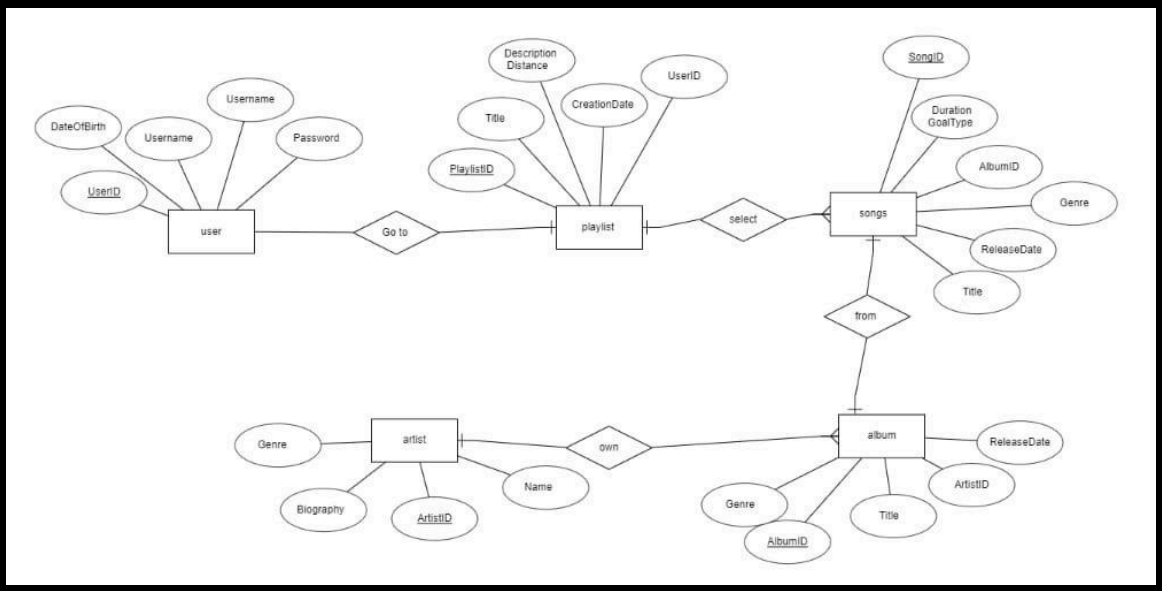



neo4j$ match(n) return(n)
Graph
Table
Text
Code
vijay
STUDIED_AT
cse
STUDIED_AT
John
Dharsana


neo4j$ match(n)return(n)
Graph
Table
Text
Code
cse
STUDIED_AT
vijay
STUDIED_AT
John

VTR UGE2021- (CBCS)


user
user_id
name
email
password
Artist
artist_id
name
country
genre
genre_id
genre_name
rating
rating_id
user_id
song_id
rating_value
song
song_id
title
duration
album_id
artist_id
genre_id


Description
Distance
UserID
Username
DateOfBirth
Username
Password
Title
CreationDate
SongID
Duration
GoalType
PlaylistID
AlbumID
UserID
Genre
user
Go to
playlist
select
songs
ReleaseDate
Title
from
Genre
artist
own
album
ReleaseDate
Name
ArtistID
Biography
ArtistID
Genre
Title
AlbumID

```
SQL> CREATE TABLE Users0 (
  2      UserID NUMBER PRIMARY KEY,
  3      UserName VARCHAR2(50) NOT NULL,
  4      Email VARCHAR2(100) UNIQUE NOT NULL,
  5      JoinDate DATE DEFAULT SYSDATE
  6  );

Table created.
```

```
SQL>
SQL> CREATE TABLE Artists (
  2      ArtistID NUMBER PRIMARY KEY,
  3      ArtistName VARCHAR2(100) NOT NULL
  4  );

Table created.
```

```
SQL> CREATE TABLE Genres (
  2      GenreID NUMBER PRIMARY KEY,
  3      GenreName VARCHAR2(50) NOT NULL
  4  );

Table created.
```

```
SQL> CREATE TABLE Albums0 (
  2       Album0ID NUMBER PRIMARY KEY,
  3       Album0Name VARCHAR2(100) NOT NULL,
  4       ArtistID NUMBER,
  5       ReleaseDate DATE
  6   );

Table created.
```

```
SQL> CREATE TABLE Songs0 (
  2      SongID NUMBER PRIMARY KEY,
  3      SongName VARCHAR2(100) NOT NULL,
  4      AlbumID NUMBER,
  5      GenreID NUMBER,
  6      Duration NUMBER(5,2)
  7  );

Table created.
```

```
SQL> CREATE TABLE Ratings (
  2      RatingID NUMBER PRIMARY KEY,
  3      UserID NUMBER,
  4      SongID NUMBER,
  5      Rating NUMBER(2,1) CHECK (Rating BETWEEN 1 AND 5),
  6      RatingDate DATE DEFAULT SYSDATE
  7  );

Table created.
```

```
SQL> INSERT INTO Users0 (UserID, UserName, Email, JoinDate)
  2  VALUES (1, 'Alice', 'alice@email.com', SYSDATE);

1 row created.
```

```
1 row created.

SQL> INSERT INTO Users0 (UserID, UserName, Email, JoinDate)
  2  VALUES (2, 'Bob', 'bob@email.com', SYSDATE);

1 row created.
```

```
SQL> INSERT INTO Artists (ArtistID, ArtistName) VALUES (1, 'Taylor Swift');

1 row created.
```

```
SQL> INSERT INTO Artists (ArtistID, ArtistName) VALUES (2, 'Ed Sheeran');

1 row created.
```

```
1 row created.

SQL> INSERT INTO Genres (GenreID, GenreName) VALUES (1, 'Pop');
```

```
SQL> INSERT INTO Genres (GenreID, GenreName) VALUES (2, 'Rock');

1 row created.
```

```
1 row created.

SQL> INSERT INTO Albums0 (Album0ID, Album0Name, ArtistID, ReleaseDate)
  2  VALUES (1, '1989', 1, TO_DATE('2014-10-27','YYYY-MM-DD'));

1 row created.
```

```
SQL> INSERT INTO Albums0 (Album0ID, Album0Name, ArtistID, ReleaseDate)
  2  VALUES (2, 'Divide', 2, TO_DATE('2017-03-03','YYYY-MM-DD'));

1 row created.
```

```
SQL> INSERT INTO Songs0 (SongID, SongName, AlbumID, GenreID, Duration)
  2  VALUES (1, 'Blank Space', 1, 1, 3.51);

1 row created.
```

```
SQL>
SQL> INSERT INTO Songs0 (SongID, SongName, AlbumID, GenreID, Duration)
  2  VALUES (2, 'Shape of You', 2, 1, 4.24);

1 row created.
```

```
SQL> INSERT INTO Ratings (RatingID, UserID, SongID, Rating, RatingDate)
  2  VALUES (2, 1, 2, 4, SYSDATE);

1 row created.
```

```
SQL> INSERT INTO Ratings (RatingID, UserID, SongID, Rating, RatingDate)
  2  VALUES (3, 2, 2, 5, SYSDATE);

1 row created.
```

```
  RATINGID USERNAME
---------- --------------------------------------------------
SONGNAME
--------------------------------------------------------------------------------
    RATING RATINGDAT
---------- ---------

         3 Bob
Shape of You
         5 24-OCT-25

         4 Bob
Hello
```

```
  RATINGID USERNAME
---------- --------------------------------------------------
SONGNAME
--------------------------------------------------------------------------------
    RATING RATINGDAT
---------- ---------
         3 24-OCT-25

         5 Charlie
Blank Space
         4 24-OCT-25

         6 Charlie
```