**Database Queries**

**Create Table Scripts**

1. **Table: User**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[User](

[ID] [int] IDENTITY(1,1) NOT NULL,

[FName] [nvarchar](max) NULL,

[LName] [nvarchar](max) NULL,

[email] [nvarchar](max) NULL,

[Gender] [nvarchar](max) NULL,

[City] [nvarchar](max) NULL,

[Country] [nvarchar](max) NULL,

[Password] [nvarchar](max) NOT NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

1. **Table: Artist**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ARTIST](

[ArtistID] [int] IDENTITY(1,1) NOT NULL,

[ArtistName] [varchar](50) NOT NULL,

[Gender] [varchar](6) NOT NULL,

PRIMARY KEY CLUSTERED

(

[ArtistID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

1. **Table: Track**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TRACKS](

[TrackID] [int] NOT NULL,

[Track] [varchar](max) NOT NULL,

[ArtistID] [int] NOT NULL,

[GenreID] [int] NOT NULL,

PRIMARY KEY CLUSTERED

(

[TrackID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[TRACKS] WITH CHECK ADD FOREIGN KEY([ArtistID])

REFERENCES [dbo].[ARTIST] ([ArtistID])

GO

ALTER TABLE [dbo].[TRACKS] WITH CHECK ADD FOREIGN KEY([GenreID])

REFERENCES [dbo].[GENRE] ([GENREID])

GO

1. **Table: Genre**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[GENRE](

[GENREID] [int] IDENTITY(1,1) NOT NULL,

[GENRE] [nvarchar](20) NULL,

PRIMARY KEY CLUSTERED

(

[GENREID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

1. **Table: Country**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[COUNTRY](

[COUNTRYID] [int] IDENTITY(1,1) NOT NULL,

[Country\_Name] [nvarchar](30) NULL,

[CountryCode] [nvarchar](30) NULL,

PRIMARY KEY CLUSTERED

(

[COUNTRYID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

1. **Table: COUNTRY\_TRACK\_MAP**

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[COUNTRY\_TRACK\_MAP](

[ID] [int] IDENTITY(1,1) NOT NULL,

[CountryID] [int] NOT NULL,

[TrackID] [int] NOT NULL,

PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[COUNTRY\_TRACK\_MAP] WITH CHECK ADD FOREIGN KEY([CountryID])

REFERENCES [dbo].[COUNTRY] ([COUNTRYID])

GO

ALTER TABLE [dbo].[COUNTRY\_TRACK\_MAP] WITH CHECK ADD FOREIGN KEY([TrackID])

REFERENCES [dbo].[TRACKS] ([TrackID])

GO

**Stored Procedures Scripts**

1. **Stored procedure to verify the login of a user**

CREATE PROCEDURE SP\_VERIFY\_USER

(

@Name nvarchar(max),

@Pass nvarchar(max)

)

AS BEGIN

IF EXISTS(SELECT Fname, [password] FROM User WHERE Fname=@Name AND [password]=@Pass)

BEGIN

SELECT TOP 1 \* FROM User WHERE Fname=@Name AND [password]=@Pass

END

ELSE

BEGIN

PRINT 'No User Found'

END

END

1. **Stored Procedure to create a new user, and insert the details in the Database**

ALTER PROCEDURE SP\_CREATE\_FORM

(

@FName nvarchar(max),

@LName nvarchar(max),

@email nvarchar(max),

@Gender nvarchar(max),

@City nvarchar(max),

@Country nvarchar(max),

@Password nvarchar(max)

)

AS BEGIN

IF EXISTS(SELECT Fname, [password],email FROM T\_NEW\_USER\_MASTER WHERE Fname=@FName AND [password]=@Password)

BEGIN

INSERT INTO T\_NEW\_USER\_MASTER VALUES(@FName,@LName,@email,@Gender,@City,@Country,@Password)

END

ELSE

BEGIN

PRINT 'User Already Exists!'

END

END

1. **Stored procedure to display top 200 global songs**

CREATE PROCEDURE fetch\_artist

AS BEGIN

select ROW\_NUMBER() OVER (

ORDER BY Country\_Name) ID,

ARTISTNAME, Track as 'SONG',Country\_Name as 'COUNTRY'

from ARTIST A inner join TRACKS T on t.ArtistID=a.ArtistID

inner join COUNTRY\_TRACK\_MAP CM on t.TrackID=cm.TrackID

inner join COUNTRY C on c.COUNTRYID=cm.CountryID where c.COUNTRYID=1

end

1. **Stored procedure to display top 200 songs based on country selection**

ALTER PROCEDURE fetch\_artist\_by\_country

(

@country nvarchar(max)

)

as begin

declare @country\_id int

IF EXISTS(select top 1 COUNTRYID from country where Country\_Name= @country)

begin

set @country\_id= (select top 1 COUNTRYID from country where Country\_Name= @country)

select ROW\_NUMBER() OVER (

ORDER BY Country\_Name) ID,

ARTISTNAME, Track as 'SONG',Country\_Name as 'COUNTRY'

from ARTIST A inner join TRACKS T on t.ArtistID=a.ArtistID

inner join COUNTRY\_TRACK\_MAP CM on t.TrackID=cm.TrackID

inner join COUNTRY C on c.COUNTRYID=cm.CountryID where c.COUNTRYID=@country\_id

end

else

begin

print 'Country Not Listed'

end

end

