

Database Management Systems

CSC 220

Assignment 3/5 Report

Submitted to: **Sir Hassan Tanvir and Sir Sikandar Hayat**

Submitted by: **Mubeen Ahmed Jawad (01-134182-081)**

**Umair Ilyas (01-134182-108)**

Class: **BSCS 4-A**

Submission Date: **7th May, 2020**

Department of Computer Science

Bahria University Islamabad Campus

**Table of Contents**

Question 1………………………………………………………………………………………………………….. Page 3

Question 2………………………………………………………………………………………………………….. Page 4

Question 3………………………………………………………………………………………………………….. Page 4

Question 4………………………………………………………………………………………………………….. Page 5

Question 5………………………………………………………………………………………………………….. Page 6

Question 6………………………………………………………………………………………………………….. Page 7

Question 7………………………………………………………………………………………………………….. Page 7

Question 8………………………………………………………………………………………………………….. Page 8

Question 9………………………………………………………………………………………………………….. Page 8

Question 10………………………………………………………………………………………………………… Page 9

Database Code……………………………………………………………………………………………………. Page 10

Q.1 Display the list of artists who have performed in each season of Nescafe Basement along with relevant details, e.g. date performed, song title and author, etc.? You should display all artists along with the songs for season 1, then season 2 and so on.

**Query:**

SELECT\*

FROM Song

INNER JOIN Musician

ON Song.MusicianID = Musician.MusicianID

INNER JOIN Genre

ON Genre.SongID = Song.SongID

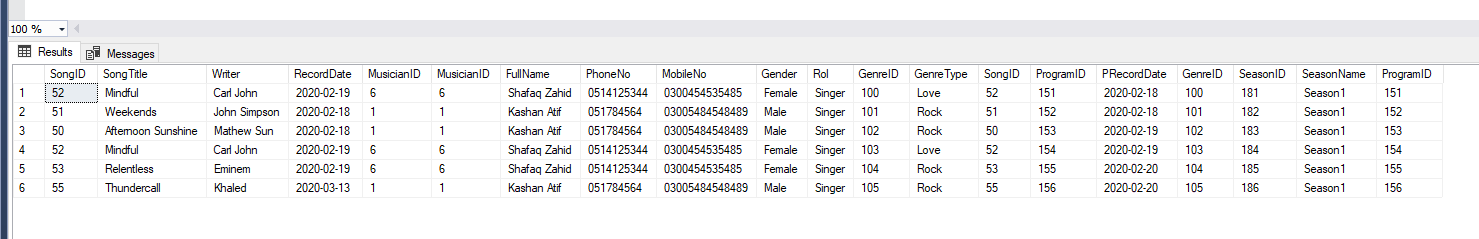
INNER JOIN Program

ON Program.GenreID = Genre.GenreID

INNER JOIN Season

ON Season.ProgramID = Program.ProgramID

Order by SeasonID asc;



Q.2 List all the musicians that have performed in the Nescafe Basement ever; along with the number of appearances and the type of music they do, e.g. rock, pop, etc.?

**Query:**

SELECT \*

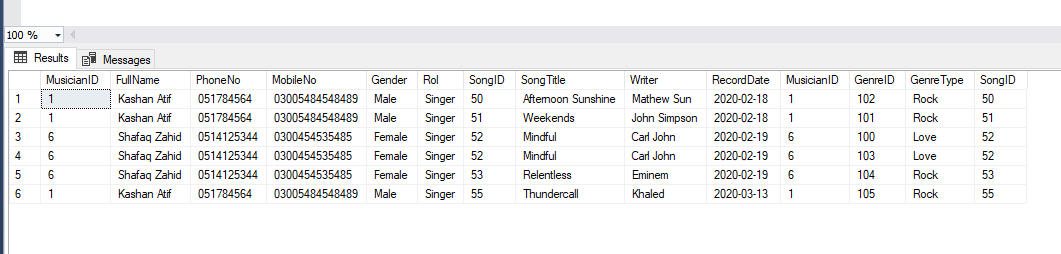
FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Genre

On Genre.SongID = Song.SongID;



Q.3 Display the musicians that can play a specific instrument, say e.g. flute, along with their contact details (both landline and mobile number)?

**Query:**

SELECT Musician.FullName, Musician.PhoneNo, Musician.MobileNo, Instrument.InstrumentName

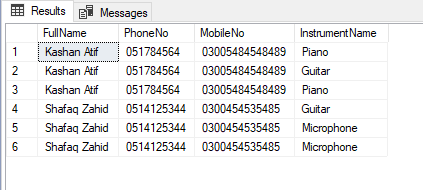
FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID;



Q.4 Show in Descending order the number of musicians that can play each instrument along with the instrument name?

**Query:**

SELECT Musician.FullName, Musician.PhoneNo, Musician.MobileNo, Instrument.InstrumentName

FROM Musician

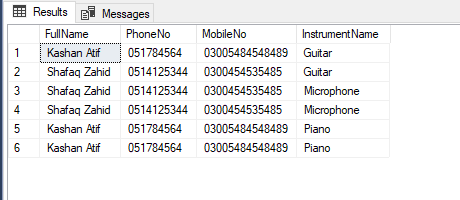
INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID

order by InstrumentName;



Q.5 List the records which have been recorded by more than one artist? You should display the record title, author, and the artists who have performed the record.

**Query:**

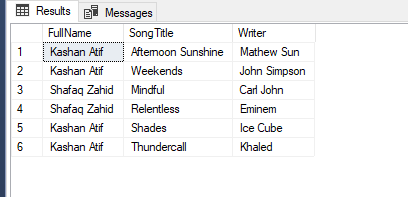
SELECT Musician.FullName, Song.SongTitle, Song.Writer

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

where SongTitle in (select SongTitle from Song group by SongTitle having count (\*) >= 1);



Q.6 For a specific record, list all the instruments and the key notes that have been used to produce the record?

**Query:**

SELECT Song.SongTitle, Instrument.InstrumentName, MusicalKey.MusicKeyType

FROM Musician

INNER JOIN Song

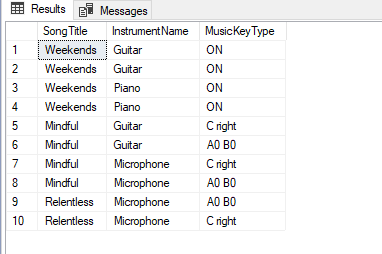
ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID

Inner Join MusicalKey

On MusicalKey.SongID = Song.SongID;



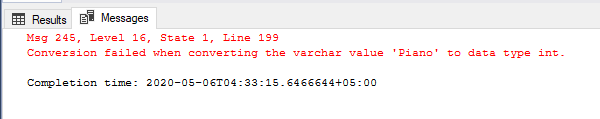
Q. 7 List all the records that use more than 3 instruments along with the number of instruments they use?

**Query:**

select Song.SongTitle, Instrument.InstrumentName,(LEN(Instrument.InstrumentName)-LEN(replace(Instrument.InstrumentName,',',' ')) + 1)

from Song inner join Instrument on Instrument.InstrumentName = Song.MusicianID where

(LEN(Instrument.InstrumentName)-LEN(replace(Instrument.InstrumentName,',',' ')) + 1) >= 1;



Q.8 List the band members (along with their contact information) of the band that has performed most songs ever among all bands in the Nescafe Basement?

**Query:**

SELECT Band.BandName, Musician.PhoneNo, Musician.MobileNo

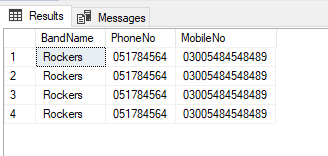
FROM Musician

INNER JOIN Band

ON Band.MusicianID = Musician.MusicianID

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID;



Q.9 List the top 3 most popular musical keys and also the song titles which use those keys?

**Query:**

select top 3 MusicalKey.MusicKeyType, Song.SongTitle

from Musician

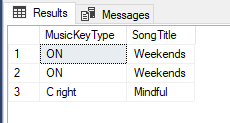
INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

INNER JOIN MusicalKey

ON MusicalKey.SongID = Song.SongID

where Song.MusicianID = Musician.MusicianID;



Q.10 Show the list of artists (along with the music type) that performed in the most recent season?

**Query:**

SELECT Musician.FullName, Season.SeasonName

FROM Song

INNER JOIN Musician

ON Song.MusicianID = Musician.MusicianID

INNER JOIN Genre

ON Genre.SongID = Song.SongID

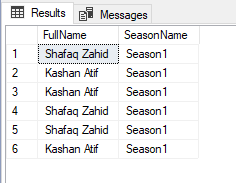
INNER JOIN Program

ON Program.GenreID = Genre.GenreID

INNER JOIN Season

ON Season.ProgramID = Program.ProgramID

order by SeasonID asc;



**Database Code:**

create Table Musician

(

MusicianID int Primary Key NOT NULL,

FullName Varchar(30),

PhoneNo Varchar(15),

MobileNo Varchar(15),

Gender Varchar(10),

Rol Varchar(15)

);

create Table Solo

(

SoloID int Primary Key NOT NULL,

MusicianID int

);

create Table Band

(

BandID int Primary Key NOT NULL,

BandName Varchar(20),

MusicianID int

);

create Table Song

(

SongID int Primary Key NOT NULL,

SongTitle Varchar(30),

Writer Varchar(30),

RecordDate DATE,

MusicianID int

);

create Table Instrument

(

InstrumentID int Primary KEY NOT NULL,

InstrumentName Varchar(20),

SongID int

);

create Table MusicalKey

(

MusicKeyID int Primary Key NOT NULL,

MusicKeyType Varchar(20),

SongID int

);

create Table Genre

(

GenreID int Primary Key NOT NULL,

GenreType Varchar(30),

SongID int

);

create Table Program

(

ProgramID int Primary Key NOT NULL,

PRecordDate DATE,

GenreID int

);

create Table Season

(

SeasonID int Primary Key NOT NULL,

SeasonName Varchar(20),

ProgramID int

);

create Table Studio

(

StudioID int Primary Key NOT NULL,

StudioName Varchar(20),

StudioAddress Varchar(30),

SeasonID int

);

Alter Table Solo ADD Foreign Key(MusicianID) References Musician(MusicianID);

Alter Table Band ADD Foreign Key(MusicianID) References Musician(MusicianID);

Alter Table Song ADD Foreign Key(MusicianID) References Musician(MusicianID);

Alter Table Instrument ADD Foreign Key(SongID) References Song(SongID);

Alter Table MusicalKey ADD Foreign Key(SongID) References Song(SongID);

Alter Table Genre ADD Foreign Key(SongID) References Song(SongID);

Alter Table Program ADD Foreign Key(GenreID) References Genre(GenreID);

Alter Table Season ADD Foreign Key(ProgramID) References Program(ProgramID);

Alter Table Studio ADD Foreign Key(SeasonID) References Season(SeasonID);

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(1, 'Kashan Atif', '051784564', '03005484548489', 'Male', 'Singer');

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(2, 'Zafar Malik', '051745454', '03005495648189', 'Male', 'Pianist');

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(3, 'Maria Ali', '051454544', '03005484265485', 'Female', 'Guitarist');

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(4, 'Kashif Azhar', '051422666', '0300541515485', 'Male', 'Drummer');

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(5, 'Saad Naeem', '0514151544', '03005151565485', 'Male', 'Guitarist');

insert into Musician(MusicianID, FullName, PhoneNo, MobileNo, Gender, Rol) Values(6, 'Shafaq Zahid', '0514125344', '0300454535485', 'Female', 'Singer');

insert into Band(BandID, BandName, MusicianID) Values(10, 'Rockers', 1);

insert into Band(BandID, BandName, MusicianID) Values(11, 'Rockers', 3);

insert into Band(BandID, BandName, MusicianID) Values(12, 'Zameen', 2);

insert into Band(BandID, BandName, MusicianID) Values(13, 'Zameen', 4);

insert into Band(BandID, BandName, MusicianID) Values(14, 'Zameen', 5);

insert into Solo(SoloID, MusicianID) Values(20, 6);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(50, 'Afternoon Sunshine', 'Mathew Sun', '18-Feb-2020', 1);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(51, 'Weekends', 'John Simpson', '18-Feb-2020', 1);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(52, 'Mindful', 'Carl John', '19-Feb-2020', 6);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(53, 'Relentless', 'Eminem', '19-Feb-2020', 6);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(54, 'Shades', 'Ice Cube', '13-Mar-2020', 1);

insert into Song(SongID, SongTitle, Writer, RecordDate, MusicianID) Values(55, 'Thundercall', 'Khaled', '13-Mar-2020', 1);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(70, 'Piano', 50);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(71, 'Guitar', 51);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(72, 'Microphone', 53);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(73, 'Piano', 51);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(74, 'Guitar', 52);

insert into Instrument(InstrumentID, InstrumentName, SongID) Values(75, 'Microphone', 52);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(81, 'A0 B0', 53);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(82, 'C right', 52);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(83, 'ON', 51);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(84, 'A0 B0', 52);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(85, 'C right', 53);

insert into MusicalKey(MusicKeyID, MusicKeyType, SongID) Values(86, 'ON', 51);

insert into Genre(GenreID, GenreType, SongID) Values(100, 'Love', 52);

insert into Genre(GenreID, GenreType, SongID) Values(101, 'Rock', 51);

insert into Genre(GenreID, GenreType, SongID) Values(102, 'Rock', 50);

insert into Genre(GenreID, GenreType, SongID) Values(103, 'Love', 52);

insert into Genre(GenreID, GenreType, SongID) Values(104, 'Rock', 53);

insert into Genre(GenreID, GenreType, SongID) Values(105, 'Rock', 55);

insert into Program(ProgramID, PRecordDate, GenreID) Values(151, '18-Feb-2020', 100);

insert into Program(ProgramID, PRecordDate, GenreID) Values(152, '18-Feb-2020', 101);

insert into Program(ProgramID, PRecordDate, GenreID) Values(153, '19-Feb-2020', 102);

insert into Program(ProgramID, PRecordDate, GenreID) Values(154, '19-Feb-2020', 103);

insert into Program(ProgramID, PRecordDate, GenreID) Values(155, '20-Feb-2020', 104);

insert into Program(ProgramID, PRecordDate, GenreID) Values(156, '20-Feb-2020', 105);

insert into Season(SeasonID, SeasonName, ProgramID) Values(181, 'Season1', 151);

insert into Season(SeasonID, SeasonName, ProgramID) Values(182, 'Season1', 152);

insert into Season(SeasonID, SeasonName, ProgramID) Values(183, 'Season1', 153);

insert into Season(SeasonID, SeasonName, ProgramID) Values(184, 'Season1', 154);

insert into Season(SeasonID, SeasonName, ProgramID) Values(185, 'Season1', 155);

insert into Season(SeasonID, SeasonName, ProgramID) Values(186, 'Season1', 156);

insert into Studio(StudioID, StudioName, StudioAddress, SeasonID) Values(201, 'Max Studio', 'Street 5, ShariqRoad, Lahore', 181);

insert into Studio(StudioID, StudioName, StudioAddress, SeasonID) Values(202, 'Raven Studio', 'Street 34, GaziRoad, Islamabad', 182);

insert into Studio(StudioID, StudioName, StudioAddress, SeasonID) Values(203, 'Massive Studio', 'Street 65, AzeemRoad, Pindi', 183);

insert into Studio(StudioID, StudioName, StudioAddress, SeasonID) Values(204, 'Rocket Studio', 'Street 78, MukaRoad, Karachi', 184);

SELECT\*

FROM Song

INNER JOIN Musician

ON Song.MusicianID = Musician.MusicianID

INNER JOIN Genre

ON Genre.SongID = Song.SongID

INNER JOIN Program

ON Program.GenreID = Genre.GenreID

INNER JOIN Season

ON Season.ProgramID = Program.ProgramID

order by SeasonID asc;

SELECT \*

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Genre

On Genre.SongID = Song.SongID;

SELECT Musician.FullName, Musician.PhoneNo, Musician.MobileNo, Instrument.InstrumentName

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID;

SELECT Musician.FullName, Musician.PhoneNo, Musician.MobileNo, Instrument.InstrumentName

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID

order by InstrumentName;

SELECT Musician.FullName, Song.SongTitle, Song.Writer

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

where SongTitle in (select SongTitle from Song group by SongTitle having count (\*) >= 1);

SELECT Song.SongTitle, Instrument.InstrumentName, MusicalKey.MusicKeyType

FROM Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

Inner JOIN Instrument

On Instrument.SongID = Song.SongID

Inner Join MusicalKey

On MusicalKey.SongID = Song.SongID;

select Song.SongTitle, Instrument.InstrumentName,(LEN(Instrument.InstrumentName)-LEN(replace(Instrument.InstrumentName,',',' ')) + 1)

from Song inner join Instrument on Instrument.InstrumentName = Song.MusicianID where

(LEN(Instrument.InstrumentName)-LEN(replace(Instrument.InstrumentName,',',' ')) + 1) >= 1;

SELECT Band.BandName, Musician.PhoneNo, Musician.MobileNo

FROM Musician

INNER JOIN Band

ON Band.MusicianID = Musician.MusicianID

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID;

select top 3 MusicalKey.MusicKeyType, Song.SongTitle

from Musician

INNER JOIN Song

ON Song.MusicianID = Musician.MusicianID

INNER JOIN MusicalKey

ON MusicalKey.SongID = Song.SongID

where Song.MusicianID = Musician.MusicianID;

SELECT Musician.FullName, Season.SeasonName

FROM Song

INNER JOIN Musician

ON Song.MusicianID = Musician.MusicianID

INNER JOIN Genre

ON Genre.SongID = Song.SongID

INNER JOIN Program

ON Program.GenreID = Genre.GenreID

INNER JOIN Season

ON Season.ProgramID = Program.ProgramID

order by SeasonID asc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*