CPSC 304 Project Cover Page

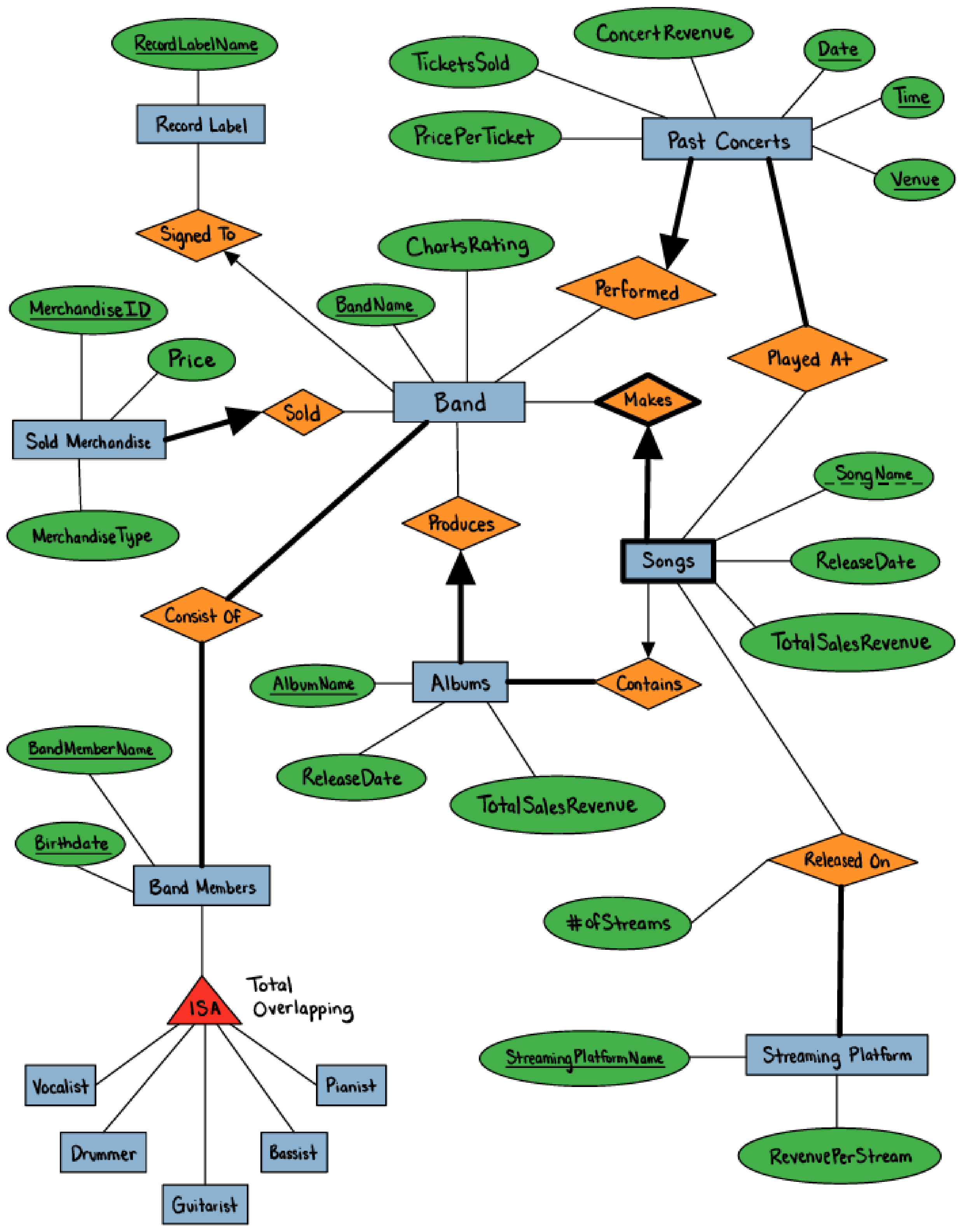
Milestone #: 2

Date: Oct 17, 2022

Group Number: 53

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Aashish Mehra	84289263	t7g1i	aashishkicks@gmail.com
Trixie Cadlaon	95858486	l3f2b	trixiecadlaon@gmail.com
Ryan Wall	12243820	n1a3b	rjwall2@shaw.ca

2.) ER Diagram


RecordLabelName
Record Label
Signed To
TicketsSold
ConcertRevenue
Date
Time
PricePerTicket
Past Concerts
Venue
ChartsRating
Performed
BandName
MerchandiseID
Price
Sold Merchandise
Sold
Band
Makes
Played At
MerchandiseType
Produces
Song Name
Songs
ReleaseDate
Consist Of
TotalSalesRevenue
AlbumName
Albums
Contains
BandMemberName
ReleaseDate
TotalSalesRevenue
Birthdate
Band Members
Released On
#ofStreams
ISA
Total
Overlapping
StreamingPlatformName
Streaming Platform
Vocalist
Pianist
Drummer
Bassist
Guitarist
RevenuePerStream

Notes:

- added attributes MerchandiseType, TicketsSold, and PricePerTicket to create meaningful functional dependencies
- changed names of some relationships for clarity

3.) Relational Schema

Attribute = Primary key
Attribute = Foreign key
No candidate keys present

Band (**BandName**: CHAR(20), ChartsRating: INTEGER, RecordLabel: CHAR(20))

BandName -> ChartsRating, RecordLabel

Record_Label (**RecordLabelName**: CHAR(20))

No non-trivial functional dependencies

Sold_Merchandise (**MerchandiseID**: INTEGER, Price: INTEGER, MerchandiseType: CHAR(10), Band: CHAR(20))

MerchandiseID -> Price, MerchandiseType, Band
MerchandiseType -> Price

Past_Concerts (**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), TicketsSold: INTEGER, PricePerTicket: INTEGER, ConcertRevenue: INTEGER, BandPlayed: CHAR(20))

Date,Time,Venue -> TicketsSold, PricePerTicket, ConcertRevenue, BandPlayed
TicketsSold, PricePerTicket -> ConcertRevenue

Albums (**AlbumName**: CHAR(20), ReleaseDate: INTEGER, TotalSalesRevenue: INTEGER, Band : CHAR(20))

AlbumName -> ReleaseDate, TotalSalesRevenue, Band

Songs (**SongName**: CHAR(30), ReleaseDate: INTEGER, TotalSalesRevenue: INTEGER, **Band**: CHAR(20), Album: CHAR(20))

SongName,Band -> ReleaseDate, TotalSalesRevenue, Album

Streaming_Platform (**StreamingPlatformName**: CHAR(20), RevenuePerStream: DECIMAL(5,4))

StreamingPlatformName -> RevenuePerStream

Released_On (#ofStreams: INTEGER, **SongName**: CHAR(30), **BandName**: CHAR(20), **StreamingPlatform**: CHAR(20))

SongName, BandName, StreamingPlatform -> #ofStreams

Played_At (**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), **SongName**: CHAR(30), **BandName**: CHAR(20))

No non-trivial functional dependencies

Consists_Of (**Band**: CHAR(20), **BandMemberName**: CHAR(25), **BandMemberBirthDate**: INTEGER)

No non-trivial functional dependencies

Contains (**AlbumName**: CHAR(20), **SongName**: CHAR(30), **BandName**: CHAR(20))

No non-trivial functional dependencies

BandMembers (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

Vocalist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

Drummer (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

Guitarist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

Bassist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

Pianist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

No non-trivial functional dependencies

** notes on the relational schema

- For Contains, need a participation constraint assert on Albums
- For Released_On need a participation constraint assert on Streaming_Platform
- For Played_At need a participation constraint assert on Past_Concerts
- For Consists_Of need a participation constraint assert on Bands and BandMembers

4.) Functional Dependencies

In blue under each relational schema

5.) Normalization

Sold_Merchandise (**MerchandiseID**: INTEGER, Price: INTEGER, MerchandiseType: CHAR(10), Band: CHAR(20))

MerchandiseID -> Price, MerchandiseType, Band
MerchandiseType -> Price

Violates 3NF as MerchandiseType (X) is not a key in Sold_Merchandise, and Price (b) is not part of a key. We will decompose into 3NF using the lossless join method

First let's get the minimal cover of our FDs:

1.) We put our FDs into standard form:

MerchandiseID -> Price
MerchandiseID -> MerchandiseType
MerchandiseID -> Band
MerchandiseType -> Price

2.) We minimize the left hand side, already done

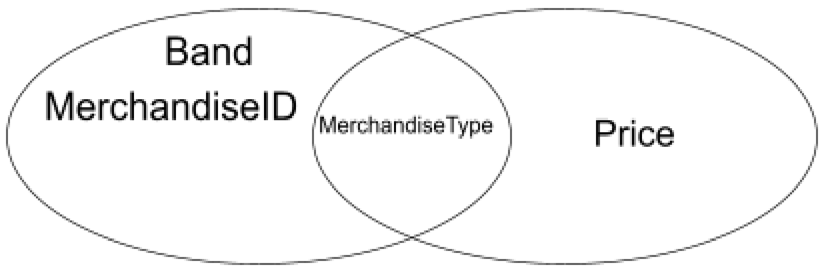
3.) Delete redundant FDs:

~~MerchandiseID -> Price~~
MerchandiseID -> MerchandiseType
MerchandiseID -> Band
MerchandiseType -> Price

With our minimal cover

MerchandiseID -> MerchandiseType
MerchandiseID -> Band
MerchandiseType -> Price

We note that MerchandiseType -> Price is the only FD that violates 3NF, so we decompose using it:



Which yields:

R_1(**MerchandiseType**: CHAR(10), Price: INTEGER)

R_2(MerchandiseType: CHAR(10), **MerchandiseID**: INTEGER, Band: CHAR(15))

We will call R_1 : Sold_Merchandise_1 and R_2 : Sold_Merchandise_2.

Past_Concerts (**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), TicketsSold: INTEGER, PricePerTicket: INTEGER, ConcertRevenue: INTEGER, BandPlayed: CHAR(20))

Date,Time,Venue -> TicketsSold, PricePerTicket, ConcertRevenue, BandPlayed
TicketsSold, PricePerTicket -> ConcertRevenue

Violates 3NF as TicketsSold,PricePerTickert (X) is not a key in Sold_Merchandise, and ConcertRevenue (b) is not part of a key. We will decompose into 3NF using the lossless join method

First let's get the minimal cover of our FDs:

1.) We put our FDs into standard form:

Date,Time,Venue -> TicketsSold
Date,Time,Venue -> PricePerTicket
Date,Time,Venue -> ConcertRevenue
Date,Time,Venue -> BandPlayed
TicketsSold,PricePerTicket -> ConcertRevenue

2.) We minimize the left hand side, already done

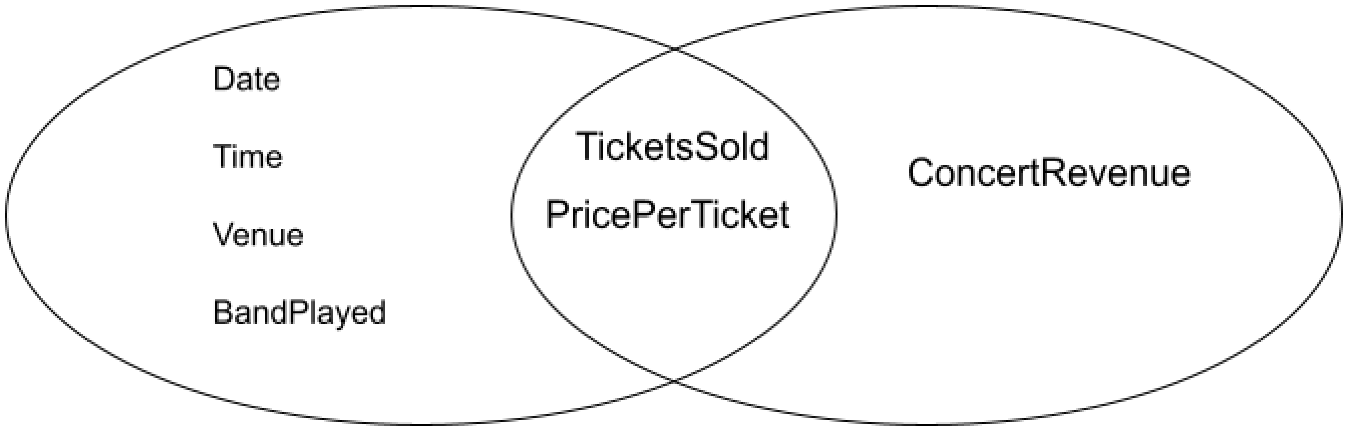
3.) Delete redundant FDs:

Date,Time,Venue -> TicketsSold
Date,Time,Venue -> PricePerTicket
~~Date,Time,Venue -> ConcertRevenue~~
Date,Time,Venue -> BandPlayed
TicketsSold,PricePerTicket -> ConcertRevenue

With our minimal cover

Date,Time,Venue -> TicketsSold
Date,Time,Venue -> PricePerTicket
Date,Time,Venue -> BandPlayed
TicketsSold,PricePerTicket -> ConcertRevenue

We note that TicketsSold,PricePerTicket -> ConcertRevenue is the only FD that violates 3NF, so we decompose using it:



Which yields:

R_1(**TicketsSold**: INTEGER, **PricePerTicket**: INTEGER, ConcertRevenue: INTEGER)

R_2(**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), <u>BandPlayed</u>: CHAR(20), <u>TicketsSold</u>: INTEGER, <u>PricePerTicket</u>: INTEGER)

We will call R_1 : Past_Concerts_1 and R_2 : Past_Concerts_2.

<u>New Relational Schema:</u>

Attribute = Primary key
<u>Attribute</u> = Foreign key
No candidate keys present

Band (**BandName**: CHAR(20), ChartsRating: INTEGER, <u>RecordLabel</u>: CHAR(20))

Record_Label (**RecordLabelName**: CHAR(20))

Sold_Merchandise_1 (**MerchandiseType**: CHAR(10), Price: INTEGER)

Sold_Merchandise_2 (MerchandiseType: CHAR(10), **MerchandiseID**: INTEGER, Band: CHAR(20))

Past_Concerts_1 (**TicketsSold**: INTEGER, **PricePerTicket**: INTEGER, ConcertRevenue: INTEGER)

Past_Concerts_2 (**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), BandPlayed: CHAR(20), TicketsSold: INTEGER, PricePerTicket: INTEGER)

Albums (**AlbumName**: CHAR(20), ReleaseDate: INTEGER, TotalSalesRevenue: INTEGER, Band : CHAR(20))

Songs (**SongName**: CHAR(30), ReleaseDate: INTEGER, TotalSalesRevenue: INTEGER, **Band**: CHAR(20), Album: CHAR(20))

Streaming_Platform (**StreamingPlatformName**: CHAR(20), RevenuePerStream: DECIMAL(5,4))

Released_On (#ofStreams: INTEGER, **SongName**: CHAR(30), **BandName**: CHAR(20), **StreamingPlatform**: CHAR(20))

Played_At (**Date**: INTEGER, **Time**: INTEGER, **Venue**: CHAR(20), **SongName**: CHAR(30), **BandName**: CHAR(20))

Consists_Of (**Band**: CHAR(20), **BandMemberName**: CHAR(25), **BandMemberBirthDate**: INTEGER)

Contains (**AlbumName**: CHAR(20), **SongName**: CHAR(30), **BandName**: CHAR(20))

BandMembers (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

Vocalist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

Drummer (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

Guitarist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

Bassist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

Pianist (**BandMemberName**: CHAR(25), **BirthDate**: INTEGER)

** <u>notes on the relational schema</u>

- For Contains, need a participation constraint assert on Albums
- For Released_On need a participation constraint assert on Streaming_Platform
- For Played_At need a participation constraint assert on Past_Concerts_2
- For Consists_Of need a participation constraint assert on Bands and BandMembers

6.) SQL DDL Statements

```
CREATE TABLE Record_Label (
        RecordLabelName CHAR(20) PRIMARY KEY
        );

CREATE TABLE Band (
        BandName CHAR(20) PRIMARY KEY,
        ChartsRating INTEGER,
        RecordLabel CHAR(20),
        FOREIGN KEY (RecordLabel) REFERENCES Record_Label(RecordLabelName)
                ON DELETE SET NULL
        );


CREATE TABLE Sold_Merchandise_1 (
        MerchandiseType CHAR(10) PRIMARY KEY,
        Price INTEGER
        );

CREATE TABLE Sold_Merchandise_2 (
        MerchandiseType CHAR(10),
        MerchandiseID INTEGER PRIMARY KEY,
        Band CHAR(20) NOT NULL,
        FOREIGN KEY (MerchandiseType) REFERENCES
                Sold_Merchandise_1(MerchandiseType)
                ON DELETE CASCADE,
        FOREIGN KEY (Band) REFERENCES Band(BandName)
                ON DELETE CASCADE
        );

CREATE TABLE Past_Concerts_1 (
        TicketsSold INTEGER,
        PricePerTicket INTEGER,
        ConcertRevenue INTEGER,
        PRIMARY KEY (TicketsSold, PricePerTicket)
        );

CREATE TABLE Past_Concerts_2 (
        Date INTEGER,
        Time INTEGER,
        Venue CHAR(20),
        BandPlayed CHAR(20) NOT NULL,
        TicketsSold INTEGER,
        PricePerTicket INTEGER,
        PRIMARY KEY (Date, Time, Venue),
        FOREIGN KEY (BandPlayed) REFERENCES Band(BandName),
        FOREIGN KEY (TicketsSold, PricePerTicket) REFERENCES
                Past_Concerts_1(TicketsSold, PricePerTicket)
                ON DELETE CASCADE
        );

CREATE TABLE Albums (
        AlbumName CHAR(20) PRIMARY KEY,
        ReleaseDate INTEGER,
        TotalSalesRevenue INTEGER,
        Band CHAR(20) NOT NULL,
        FOREIGN KEY(Band) REFERENCES Band(BandName)
                ON DELETE CASCADE
        );

CREATE TABLE Songs(
        SongName CHAR(30),
        RelseaseDate INTEGER,
        TotalSalesRevenue INTEGER,
        Band CHAR(20),
        Album CHAR(20),
        PRIMARY KEY (SongName, Band),
        FOREIGN KEY (Band) REFERENCES Band(BandName)
                ON DELETE CASCADE,
        FOREIGN KEY (Album) REFERENCES Albums(AlbumName)
                ON DELETE SET NULL
        );

CREATE TABLE Streaming_Platform(
        StreamingPlatformName CHAR(20) PRIMARY KEY,
        RevenuePerStream DECIMAL(5,4)
        );

CREATE TABLE Released_On(
        `#ofStreams` INTEGER,
        SongName CHAR(30),
        BandName CHAR(20),
        StreamingPlatform CHAR(20),
        PRIMARY KEY (SongName, BandName, StreamingPlatform),
        FOREIGN KEY (SongName, BandName) REFERENCES Songs(SongName,Band)
                ON DELETE CASCADE,
        FOREIGN KEY (StreamingPlatform) REFERENCES
                Streaming_Platform(StreamingPlatformName)
                ON DELETE CASCADE
        );
        #Need a participation constraint assertion on Streaming_Platform

CREATE TABLE Played_At(
        Date INTEGER,
        Time INTEGER,
        Venue CHAR(20),
        SongName CHAR(30),
        BandName CHAR(20),
        PRIMARY KEY (Date, Time, Venue, SongName, BandName),
        FOREIGN KEY (SongName, BandName) REFERENCES Songs(SongName,Band)
                ON DELETE CASCADE,
        FOREIGN KEY (Date, Time, Venue) REFERENCES Past_Concerts_2(Date, Time,
                Venue)
                ON DELETE CASCADE
        );
        #Need a participation constraint assertion on Past_Concerts_2



CREATE TABLE Contains (
        AlbumName CHAR(20),
        SongName CHAR(30),
        BandName CHAR(20),
        PRIMARY KEY (AlbumName, SongName, BandName),
        FOREIGN KEY (AlbumName) REFERENCES Albums (AlbumName)
                ON DELETE CASCADE,
        FOREIGN KEY (SongName, BandName) REFERENCES Songs(SongName,Band)
                ON DELETE CASCADE
        );
        #Need a participation constraint assertion on Albums

CREATE TABLE BandMembers (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate)
        );

CREATE TABLE Vocalist (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate),
        FOREIGN KEY (BandMemberName, BirthDate) REFERENCES
                BandMembers(BandMemberName, BirthDate)
                ON DELETE CASCADE
        );

CREATE TABLE Drummer (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate),
        FOREIGN KEY (BandMemberName, BirthDate) REFERENCES
                BandMembers(BandMemberName, BirthDate)
                ON DELETE CASCADE
        );

CREATE TABLE Guitarist (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate),
        FOREIGN KEY (BandMemberName, BirthDate) REFERENCES
                BandMembers(BandMemberName, BirthDate)
                ON DELETE CASCADE
        );

CREATE TABLE Bassist (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate),
        FOREIGN KEY (BandMemberName, BirthDate) REFERENCES
                BandMembers(BandMemberName, BirthDate)
                ON DELETE CASCADE
        );

CREATE TABLE Pianist (
        BandMemberName CHAR(25),
        BirthDate INTEGER,
        PRIMARY KEY (BandMemberName, BirthDate),
        FOREIGN KEY (BandMemberName, BirthDate) REFERENCES
                BandMembers(BandMemberName, BirthDate)
                ON DELETE CASCADE
        );
```

```
CREATE TABLE Consists_Of (
        Band CHAR(20),
        BandMemberName CHAR(25),
        BandMemberBirthDate INTEGER,
        PRIMARY KEY (Band, BandMemberName, BandMemberBirthDate),
        FOREIGN KEY (Band) REFERENCES Band (BandName)
                ON DELETE CASCADE,
        FOREIGN KEY (BandMemberName, BandMemberBirthDate)
                REFERENCES BandMembers (BandMemberName, BirthDate)
                ON DELETE CASCADE
        );
        #Need a participation constraint assertion on Bands and BandMembers
```

7.) Insert Tuples

```
INSERT INTO Record_Label(RecordLabelName)
VALUES (‘Atlantic Records’);

INSERT INTO Record_Label(RecordLabelName)
VALUES (‘EMI’);

INSERT INTO Record_Label(RecordLabelName)
VALUES (‘Apple Records’);

INSERT INTO Record_Label(RecordLabelName)
VALUES (‘Warner Records’);

INSERT INTO Record_Label(RecordLabelName)
VALUES (‘Interscope Records’);


INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘AC/DC’, ‘65’, ‘Atlantic Records’);

INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘Queen’, ‘57’, ‘EMI’’);

INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘The Beatles’, ‘80’, ‘Apple Records’);

INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘Fleetwood Mac, ‘20’, ‘Warner Records’);

INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘Imagine Dragons’, ‘28’, ‘Interscope Records’);

INSERT INTO Band(BandName, ChartsRating, RecordLabel)
VALUES (‘One Republic’, ‘26’, ‘Interscope Records’);



INSERT INTO Sold_Merchandise_1(MerchandiseType, Price)
VALUES (‘Hoodie’, ‘59’);

INSERT INTO Sold_Merchandise_1(MerchandiseType, Price)
VALUES (‘T-Shirt’, ‘30’);
```

```
INSERT INTO Sold_Merchandise_1(MerchandiseType, Price)
VALUES (‘Poster’, ‘10’);

INSERT INTO Sold_Merchandise_1(MerchandiseType, Price)
VALUES (‘Vinyl’, ‘23’);

INSERT INTO Sold_Merchandise_1(MerchandiseType, Price)
VALUES (‘CD’, ‘22’);


INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Hoodie’, ‘14’, ‘AC/DC’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Poster’, ‘29’, ‘AC/DC’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘T-Shirt’, ‘1002’, ‘Queen’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Poster’, ‘1003’, ‘Queen’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Vinyl’, ‘1004’, ‘Queen’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘T-Shirt’, ‘2054’, ‘The Beatles’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Vinyl’, ‘2055’, ‘The Beatles’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘T-Shirt’, ‘3011’, ‘Fleetwood Mac’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Hoodie’, ‘4225’, ‘Imagine Dragons’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘Poster’, ‘4222’, ‘Imagine Dragons’);

INSERT INTO Sold_Merchandise_2(MerchandiseType, Merchandise ID, Band)
VALUES (‘CD’, ‘5008’, ‘One Republic’);
```

```
INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘19223’, ‘100’, ’1922300’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘20000’, ‘129’, ‘2580000’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘20000’, ‘119’, ‘2380000’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘5272’, ‘140’, ‘738080’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘5200’, ‘50’, ‘260000’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘18533’, ‘75’, ‘1389975’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘6008’, ‘70’, ‘420560’);

INSERT INTO Past_Concerts_1(TicketsSold, PricePerTicket, ConcertRevenue)
VALUES (‘5724’, ‘70’, ‘400680’);


INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘21062013’, ‘1900’, ’Rogers Arena’, ‘AC/DC’, ‘19223’, ‘100’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘04072015’, ‘1830’, ’O2 Arena’, ‘Queen’, ‘20000’, ‘129’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘16082007’, ‘2000’, ’The Gorge Amphitheater’, ‘The Beatles’, ‘20000’, ‘119’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘06112009’, ‘1930’, ’Royal Albert Hall’, ‘The Beatles’, ‘5272’, ‘140’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘20122018’, ‘1800’, ’Royal Albert Hall’, ‘Fleetwood Mac’, ‘5200’, ‘50’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘09012022’, ‘1800’, ’O2 Arena’, ‘Imagine Dragons’, ‘18533’, ‘75’);
```

```
INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘09062018’, ‘1800’, ’PNE Amphitheater’, ‘Imagine Dragons’, ‘6008’, ‘70’);

INSERT INTO Past_Concerts_2(Date, Time, Venue, BandPlayed, TicketsSold, PricePerTicket)
VALUES (‘08062018’, ‘1800’, ’PNE Amphitheater’, ‘One Republic’, ‘5724’, ‘70’);


INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘Highway to Hell’, ‘27071979’, ‘90890023’, ‘AC/DC’);

INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘A Night at the Opera’, ‘12121975’, ‘129849085’, ‘Queen’);

INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘Abbey Road’, ‘05071969’, ‘189623765’, ‘The Beatles’);

INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘Rumours’, ‘11041977’, ‘8429019’, ‘Fleetwood Mac’);

INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘Night Visions’, ‘10082012’, ‘52987543’, ‘Imagine Dragons’);

INSERT INTO Albums(AlbumName, ReleaseDate, TotalSalesRevenue, Band)
VALUES (‘Waking Up’, ‘15022009’, ‘39546903’, ‘One Republic’);



INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Walk All Over You’, ‘27071979’, ‘827300’, ‘AC/DC’,‘Highway to Hell’);

INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Bohemian Rhapsody’, ‘12121975’, ‘9986831’, ‘Queen’,‘A Night at the Opera’);

INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Here Comes the Sun’, ‘05071969’, ‘3024365’, ‘The Beatles’,‘Abbey Road’);

INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘The Chain’, ‘11041977’, ‘102948’, ‘Fleetwood Mac’,‘Rumours’);

INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Radioactive’, ‘10082012’, ‘1653112’, ‘Imagine Dragons’,‘Night Visions’);

INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Good Life’, ‘15022009’, ‘1186534’, ‘One Republic’,‘Waking Up’);
```

```
INSERT INTO Songs(SongName, ReleaseDate, TotalSalesRevenue, Band, Album)
VALUES (‘Face It Alone’, ‘16081985’, ‘3000’, ‘Queen’, NULL);


INSERT INTO Streaming_Platform(StreamingPlatformName, RevenuePerStream)
VALUES (‘Youtube Music’, ‘0.0020’);

INSERT INTO Streaming_Platform(StreamingPlatformName, RevenuePerStream)
VALUES (‘Apple Music’, ‘0.0010’);

INSERT INTO Streaming_Platform(StreamingPlatformName, RevenuePerStream)
VALUES (‘Spotify’, ‘0.0025’);

INSERT INTO Streaming_Platform(StreamingPlatformName, RevenuePerStream)
VALUES (‘iHeartRadio’, ‘0.0010’);

INSERT INTO Streaming_Platform(StreamingPlatformName, RevenuePerStream)
VALUES (‘Amazon Music’, ‘0.0015’);


INSERT INTO Released_On(#ofStream, SongName, BandName, StreamingPlatform)
VALUES (‘1000000000’, ‘Bohemian Rhapsody’, ‘Queen’, ‘Spotify’);

INSERT INTO Released_On(#ofStream, SongName, BandName, StreamingPlatform)
VALUES (‘700938212’, ‘Here Comes The Sun’, ‘The Beatles’, ‘Apple Music’);

INSERT INTO Released_On(#ofStream, SongName, BandName, StreamingPlatform)
VALUES (‘1571220’, ‘The Chain’, ‘Fleetwood Mac’, ‘Amazon Music’);

INSERT INTO Released_On(#ofStream, SongName, BandName, StreamingPlatform)
VALUES (‘1300086234’, ‘Radioactive’, ‘Imagine Dragons’, ‘iHeartRadio’);

INSERT INTO Released_On(#ofStream, SongName, BandName, StreamingPlatform)
VALUES (‘200865412’, ‘Good Life’, ‘One Republic’, ‘Youtube Music’);


INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (‘21062013’, ‘1900’, ’Rogers Arena’, ‘Walk All Over You’, ‘AC/DC’);

INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (‘04072015’, ‘1830’, ’O2 Arena’, ‘Bohemian Rhapsody’, ‘Queen’);
```

```
INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (16082007’, ‘2000’, ’The Gorge Amphitheater’, ‘Here Comes The Sun’, ‘The Beatles’);

INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (‘20122018’, ‘1800’, ’Royal Albert Hall’, ‘The Chain’, ‘Fleetwood Mac’);

INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (‘09012022’, ‘1800’, ’O2 Arena’, ‘Radioactive’, ‘Imagine Dragons’);

INSERT INTO Played_At(Date, Time, Venue, SongName, BandName )
VALUES (‘08062018’, ‘1800’, ’PNE Amphitheater’, ‘Good Life’, ‘One Republic’);


INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘Highway to Hell’, ‘Walk All Over You’, ‘AC/DC’);

INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘A Night at the Opera’, ‘Bohemian Rhapsody’, ‘Queen’);

INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘Abbey Road’, ‘Here Comes The Sun’, ‘The Beatles’);

INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘Rumours’, ‘The Chain’, ‘Fleetwood Mac’);

INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘Night Visions’, ‘Radioactive’, ‘Imagine Dragons’);

INSERT INTO Contains(AlbumName, SongName, BandName )
VALUES (‘Waking Up’, ‘Good Life’, ‘One Republic’);


INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘Chris Slade’, ‘30101946’);

INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘Freddie Mercury’, ‘05091946’);

INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘John Lennon’, ‘09101940’);

INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘Lindsey Buckingham’, ‘03101949’);
```

```
INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘Ben McKee’, ‘07041985’);

INSERT INTO BandMembers(BandMemberName, BirthDate)
VALUES (‘Ryan Tedder’, ‘26061979’);


INSERT INTO Vocalist(BandMemberName, BirthDate)
VALUES (‘Freddie Mercury’, ‘05091946’);

INSERT INTO Drummer(BandMemberName, BirthDate)
VALUES (‘Chris Slade’, ‘30101946’);

INSERT INTO Guitarist(BandMemberName, BirthDate)
VALUES (‘John Lennon’, ‘09101940’);

INSERT INTO Bassist(BandMemberName, BirthDate)
VALUES (‘Ben McKee’, ‘07041985’);

INSERT INTO Pianist(BandMemberName, BirthDate)
VALUES (‘Ryan Tedder’, ‘26061979’);
```

Note: We believe it’s unrealistic for a band to have, for example, 5 drummers or 5 pianists. Thus, the 5 tables above (Vocalist, Drummer, Guitarist, Bassist, and Pianist) have less than 5 tuples because we think it’s more representative of our domain. The parent class (BandMembers) still has 5 tuples.

```
INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘AC/DC’, ‘Chris Slade’, ‘30101946’);

INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘Queen’, ‘Freddie Mercury’ ’, ‘05091946’);

INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘The Beatles’, ‘John Lennon’ ’, ‘09101940’);

INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘Fleetwood Mac’, ‘Lindsey Buckingham’, ‘03101949’);

INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘Imagine Dragons’, ‘Ben McKee’, ‘07041985’);

INSERT INTO Consists_Of(Band, BandMemberName, BandMemberBirthDate)
VALUES (‘One Republic’, ‘Ryan Tedder’, ‘26061979’);
```