CPSC 304 Project Cover Page

Milestone #: 2

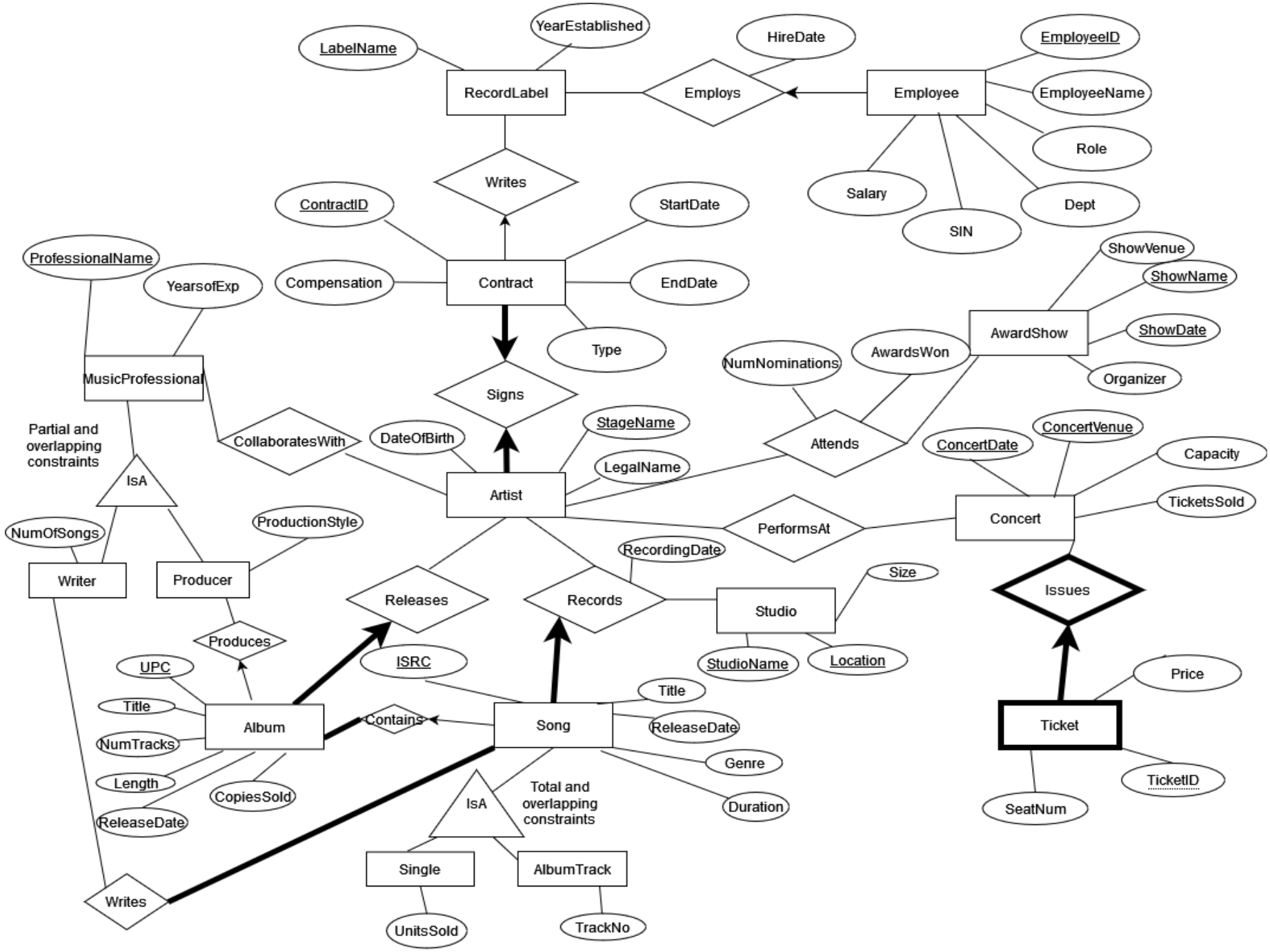
Date: October 10, 2024

Group Number: 71

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Andrew Xie	23613136	x3s5u	adxie12@gmail.com
Alice Sin	16582144	g2z0b	sin.alicee@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia


LabelName
YearEstablished
RecordLabel
HireDate
Employs
EmployeeID
Employee
EmployeeName
Role
Salary
SIN
Dept
Writes
ContractID
StartDate
Compensation
Contract
EndDate
Type
ProfessionalName
YearsofExp
MusicProfessional
Signs
ShowVenue
ShowName
ShowDate
Organizer
AwardShow
NumNominations
AwardsWon
Attends
Partial and overlapping constraints
IsA
CollaboratesWith
DateOfBirth
StageName
LegalName
Artist
ConcertVenue
ConcertDate
Capacity
TicketsSold
Concert
PerformsAt
NumOfSongs
ProductionStyle
Writer
Producer
RecordingDate
Releases
Records
Size
Studio
Issues
Produces
UPC
ISRC
StudioName
Location
Price
Title
Album
Contains
Song
Title
ReleaseDate
Ticket
NumTracks
Genre
Length
CopiesSold
IsA
Total and overlapping constraints
Duration
TicketID
ReleaseDate
SeatNum
Single
AlbumTrack
Writes
UnitsSold
TrackNo

Project Summary:

Our application is used to model how a record label operates. The application is used to show the business and creative processes of a record label with a focus on sales information.

Changes to ER Diagram:

- Change RecordLabel key: Label -> LabelName
- Change Employee key: Name -> EmployeeName
- Change MusicProfessional key -> ProfessionalName
- Change Records attribute: date -> RecordingDate
- Change AwardShow: Name -> ShowName
- Change Studio: name -> StudioName
- Added IsA constraints next to the triangles as per TA's feedback
- Changed it into many-to-one relationship for employee and record label per TA's feedback
- Album to song changed to total participation since albums cannot be without any songs
- Removed Event and made AwardShow/Concert individual Entities; added attributes and relations to both because we wanted to relate the NumNominations and AwardsWon attributes to the artist

Relational Schema from the ER diagram:

Notes:
Underline - PRIMARY KEY
Bold - **FOREIGN KEY**
Candidate keys will be stated within tables

RecordLabel(LabelName: VARCHAR, YearEstablished: INT)

EmploysEmployee(EmployeeID: INT, EmployeeName: VARCHAR, Role: VARCHAR, Dept: VARCHAR, SIN: VARCHAR, Salary: INT, HireDate: DATE, **LabelName**: VARCHAR)

- SIN is unique
- SIN is candidate key
- employeeName, role, dept, SIN, salary, hireDate are not null

WritesContract(ContractID: VARCHAR, Compensation: INT, StartDate: DATE, EndDate: DATE, Type: VARCHAR, **LabelName**: VARCHAR, **StageName:** VARCHAR)

- StageName must be unique and not NULL
- Compensation, startDate, type, labelName are not null

ArtistSigns(StageName: VARCHAR, LegalName: VARCHAR, DateOfBirth: DATE, **ContractID**: VARCHAR)

- ContractID must be unique
- ContractID is a candidate key
- Contract ID, dateOfBirth, legalName are not null

CollaboratesWith(**ProfessionalName**: VARCHAR, **StageName**: VARCHAR)

Attends (**StageName**: VARCHAR, **ShowName**: VARCHAR, **ShowDate**: DATE, NumNominations: INT, NumAwards: INT)

MusicProfessional(ProfessionalName: VARCHAR, YearsofExp: INT)

Writer(**ProfessionalName**: VARCHAR**,** NumOfSongs: INT)

Producer(**ProfessionalName**: VARCHAR, ProductionStyle: VARCHAR)

Album(UPC: INT, **ProfessionalName**: VARCHAR, **StageName**: VARCHAR, Title: VARCHAR, NumTracks: INT, Length: INTERVAL, ReleaseDate: DATE, CopiesSold: INT)

- Candidate Key (StageName, Title)
- (StageName, Title) is unique
- UPC, professionalName, stageName, title, numTracks, length, releaseDate are not null
- Can't enforce total participation on song without assertion

Song(ISRC: VARCHAR, Title: VARCHAR, ReleaseDate: DATE, Genre: VARCHAR, Duration: INTERVAL, **StageName**: VARCHAR**,** RecordingDate: DATE, **StudioName**: VARCHAR, **Location:** VARCHAR, **UPC**: INT)

- Candidate Key (StageName, Title)
- (StageName, Title) is unique
- Title, releaseDate, genre, duration, stageName, recordingDate, studioName, location, UPC are not null

Writes(**ProfessionalName**: VARCHAR, **ISRC**: VARCHAR)

- Can't enforce total participation on song without assertion
- ISRC is not null

Single(**ISRC**: VARCHAR, UnitsSold: INT)

AlbumTrack(**ISRC**: VARCHAR, TrackNo: INT)

Studio(StudioName: VARCHAR**,** Location: VARCHAR, Size: VARCHAR)

AwardShow (ShowName: VARCHAR**,** ShowDate: DATE**,** ShowVenue: VARCHAR, Organizer: VARCHAR)

PerformsAt(**ConcertDate**: DATE, **ConcertVenue**: VARCHAR, **StageName**: VARCHAR)

Concert(ConcertVenue: VARCHAR, ConcertDate: DATE, Capacity: INT, TicketsSold: INT)

IssueTicket(**Venue**: VARCHAR, **ConcertDate**: DATE, TicketID: VARCHAR, SeatNum: VARCHAR, Price: INT)

- Candidate key (Venue, ConcertDate, SeatNum)
- (Venue, ConcertDate, SeatNum) is UNIQUE

Functional Dependencies:

RecordLabel:
LabelName -> YearEstablished

EmploysEmployee:
EmployeeID -> EmployeeName, Role, Dept, SIN, HireDate, LabelName, Salary
SIN-> EmployeeID, EmployeeName, Role, Dept, Salary, HireDate, LabelName
Role, Dept -> Salary
Role -> Dept
HireDate -> Salary

WritesContract:
ContractID -> Compensation, StartDate, EndDate, Type, StageName, LabelName
StageName->ContractID
Type -> Compensation

ArtistSigns:
StageName -> LegalName, DateOfBirth, ContractID
ContractID->StageName
LegalName->DateOfBirth

CollaboratesWith:
No non-trivial FDs

Attends:
ShowName, ShowDate, StageName -> NumNominations, AwardsWon

MusicProfessional:
ProfessionalName -> YearsofExp

Writer
ProfessionalName -> NumOfSongs

Producer:

ProfessionalName -> ProductionStyle

Album:
UPC->Title, Release Date, CopiesSold, NumTracks, ProfessionalName, StageName, Length, copiesSold
StageName, Title -> Title, Release Date, CopiesSold, NumTracks, ProfessionalName, UPC, Length, copiesSold

Song
ISRC -> Title, ReleaseDate, Genre, Duration, StageName, RecordingDate, StudioName, Location, UPC
StageName, Title -> ReleaseDate, Genre, Duration, RecordingDate, StudioName, Location, UPC

Writes
No non-trivial FDs

Single
ISRC -> UnitsSold

AlbumTrack
ISRC -> TrackNo

Studio
StudioName, Location -> Size

AwardShow
ShowName, ShowDate -> ShowVenue, Organizer

Concert:
ConcertVenue, ConcertDate -> Capacity, TicketsSold
ConcertVenue-> Capacity

IssueTicket

ConcertVenue, ConcertDate, TicketID -> SeatNum, Price

TicketID -> SeatNum, Price

ConcertVenue, ConcertDate, SeatNum -> Price

Normalization Steps:

All the relations other than EmploysEmployee, WritesContract, ArtistSigns, Concert, and IssueTicket are in BCNF.

1. We decompose on **EmploysEmployee**:

For R(EmployeeID, EmployeeName, Role, Dept, SIN, HireDate, LabelName, Salary), the FD Role->Dept violates BCNF.

We first take closure of EmployeeID, SIN, (Role, Dept), Role, HireDate.
EmployeeID and SIN is a primary key and candidate key respectively, so they are superkeys and do not violate the BCNF.
$EmployeeID^+$ = {EmployeeID, EmployeeName, Role, Dept, SIN, HireDate, LabelName, Salary}
SIN^+ = {SIN, EmployeeID, EmployeeName, Role, Dept, Salary, HireDate, LabelName}
$(Role, Dept)^+$ = {Role, Dept, Salary}
$Role^+$ = {Role, Dept, Salary}
$HireDate^+$ = {HireDate, Salary}

We decompose using Role->Dept:
We get R1(Role, Dept) and R2(Role, EmployeeID, EmployeeName, SIN, HireDate, LabelName, Salary).
R1 is a two attribute relation and does not violate BCNF. HireDate -> Salary violates BCNF as it is not a superkey of R2. We decompose again.
R3(HireDate, Salary) and R4(HireDate, employeeID, employeeName, SIN, **LabelName**, Role).
None of the FDs violate R4.

So our final decomposition is as such:
R1(Role, Dept), R2(HireDate, Salary), R3(employeeID, employeeName, SIN, HireDate, **LabelName**, Role)

2. We decompose on **WritesContract**

For R(ContractID,Compensation, StartDate, EndDate, Type, StageName, LabelName), the FD StageName->ContractID violates BCNF.

We first take the closures of StageName and Type. We know that ContractID is a primary key and a superkey.
ContractID $^+$ = {ContractID, Compensation, StartDate, EndDate, Type, StageName, LabelName}
StageName$^+$ = {StageName, ContractID, Compensation, StartDate, EndDate, Type, LabelName}
Type$^+$ = {Type, Compensation}

We decompose using Type -> Compensation:
We get R1(Type, Compensation), R2(ContractID, Type, **StageName**, **LabelName**, StartDate, EndDate). Since none of the FDs violate R1 (two attribute relation) or R2 we stop.

Final decomposition:
R1(Type, Compensation), R2(ContractID, Type, **StageName**, **LabelName**, StartDate, EndDate)

3. We decompose on **ArtistSigns**

For R(StageName, LegalName, DateOfBirth, ContractID), the FD LegalName->DateOfBirth violates the BCNF.

We first take the closure of ContractID and LegalName. StageName is a primary key and superkey and do not violate the BCNF.
StageName$^+$ = {StageName, LegalName, DateOfBirth, ContractID}
ContractID$^+$ = {ContractID, StageName}
LegalName$^+$ = {LegalName, DateOfBirth}

We decompose using LegalName->DateOfBirth
We get R1(LegalName, DateOfBirth) and R2(StageName, LegalName, **ContractID**)
R1 is a two attribute relation and does not violate BCNF. None of the FDs violate R2.

So our final decomposition is as such:
R1(LegalName, DateOfBirth), R2(StageName, LegalName, **ContractID**)

4. We decompose on **Concert**

For R(ConcertVenue, ConcertDate, Capacity, TicketsSold), the FD ConcertVenue -> Capacity violates BCNF.

We first take the closures of (ConcertVenue, ConcertDate) and ConcertVenue:

(ConcertVenue, ConcertDate)$^+$ = {ConcertVenue, ConcertDate, Capacity, TicketsSold} - superkey that does not violate BCNF
ConcertVenue$^+$ = {ConcertVenue, Capacity} - violates BCNF

We decompose using ConcertVenue -> Capacity:

R1(<u>ConcertVenue</u>, Capacity), R2(<u>ConcertVenue</u>, <u>ConcertDate</u>, TicketsSold)
R1 is a two attribute relation and does not violate BCNF. There are no FDs that violate BCNF in R2.
Final decomposition:
R1(<u>ConcertVenue</u>, Capacity), R2(<u>ConcertVenue</u>, <u>ConcertDate</u>, TicketsSold)

5. We decompose on **IssueTicket**

For R(ConcertVenue, ConcertDate, TicketID, SeatNum, Price), the FDs TicketID -> SeatNum, Price and (ConcertVenue, ConcertDate, TicketID) -> Price both violate BCNF.

Closures of the FDs:
(ConcertVenue, ConcertDate, TicketID)$^+$ = {ConcertVenue, ConcertDate, TicketID, SeatNum, Price} - superkey that does not violate BCNF
TicketID$^+$ = {TicketID, SeatNum, Price} - violates BCNF
(ConcertVenue, ConcertDate, SeatNum)$^+$ = {ConcertVenue, ConcertDate, SeatNum, Price} - violates BCNF

We decompose on TicketID -> SeatNum, Price:

R1(<u>TicketID</u>, SeatNum, Price), R2(<u>**ConcertVenue**</u>, <u>**ConcertDate**</u>, TicketID)
No FDs violate R1 or R2. Therefore, our final decomposition is:

R1(<u>TicketID</u>, SeatNum, Price), R2(<u>**ConcertVenue**</u>, <u>**ConcertDate**</u>, TicketID)

Normalized Tables

Notes:
Underline - PRIMARY KEY
Bold - **FOREIGN KEY**
Candidate keys will be stated within tables

RecordLabel(LabelName: VARCHAR, YearEstablished: INT)

EmploysEmployee1(Role: VARCHAR, Dept: VARCHAR)

EmploysEmployee2(HireDate: DATE, Salary: INT)

- Salary is not null

EmploysEmployee3(employeeID: INT, employeeName: VARCHAR, SIN: VARCHAR, HireDate: DATE, **LabelName**: VARCHAR, Role: VARCHAR)

- SIN is unique, candidate key
- employeeName, SIN, hireDate, labelName, role are not null

WritesContract1(Type: VARCHAR, Compensation: INT)

- Compensation not null

WritesContract2(ContractID: VARCHAR, Type: VARCHAR, **StageName**: VARCHAR, **LabelName**: VARCHAR, StartDate: DATE, EndDate: DATE)

- Type, stageName, labelName not null

ArtistSigns1(LegalName: VARCHAR, DateOfBirth: DATE)

- legalName and dateOfBirth not null

ArtistSigns2(StageName: VARCHAR, LegalName: VARCHAR, **ContractID**: VARCHAR)

- ContractID must be unique
- ContractID is a candidate key
- Contract ID, legalName is not null

CollaboratesWith(**ProfessionalName**: VARCHAR, **StageName**: VARCHAR)

Attends (**StageName**: VARCHAR, **ShowName**: VARCHAR, **ShowDate**: DATE, NumNominations: INT, NumAwards: INT)

MusicProfessional(ProfessionalName: VARCHAR, YearsofExp: INT)

Writer(**ProfessionalName**: VARCHAR**,** NumOfSongs: INT)

Producer(**ProfessionalName**: VARCHAR, ProductionStyle: VARCHAR)

Album(UPC: INT, **ProfessionalName**: VARCHAR, **StageName**: VARCHAR, Title: VARCHAR, NumTracks: INT, Length: INTERVAL, ReleaseDate: DATE, CopiesSold: INT)

- Candidate Key (StageName, Title)
- (StageName, Title) is unique
- UPC, professionalName, stageName, title, numTracks, length, releaseDate are not null
- Can't enforce total participation on song without assertion

Song(ISRC: VARCHAR, Title: VARCHAR, ReleaseDate: DATE, Genre: VARCHAR, Duration: INTERVAL, **StageName**: VARCHAR**,** RecordingDate: DATE, **StudioName**: VARCHAR, **Location:** VARCHAR, **UPC**: INT)

- Candidate Key (StageName, Title)
- (StageName, Title) is unique
- Title, releaseDate, genre, duration, stageName, recordingDate, studioName, location, UPC are not null

Writes(**ProfessionalName**: VARCHAR, **ISRC**: VARCHAR)

- Can't enforce total participation on song without assertion
- ISRC is not null

Single(**ISRC**: VARCHAR, UnitsSold: INT)

AlbumTrack(**ISRC**: VARCHAR, TrackNo: INT)

Studio(StudioName: VARCHAR**,** Location: VARCHAR, Size: VARCHAR)

AwardShow (ShowName: VARCHAR**,** ShowDate: DATE**,** ShowVenue: VARCHAR, Organizer: VARCHAR)

PerformsAt(**ConcertDate**: DATE, **ConcertVenue**: VARCHAR, **StageName**: VARCHAR)

Concert1(ConcertVenue: VARCHAR, Capacity: INT)

Concert2(ConcertVenue: VARCHAR, ConcertDate: DATE, TicketsSold: INT)

IssueTicket1(**ConcertVenue**: VARCHAR, **ConcertDate**: DATE, TicketID: VARCHAR)

IssueTicket2(TicketID: VARCHAR, SeatNum: VARCHAR, Price: INT)

SQL DDL:

```
CREATE TABLE RecordLabel(
        labelName               VARCHAR     PRIMARY KEY,
        yearEstablished         INT
);

CREATE TABLE EmploysEmployee1(
        role:                   VARCHAR     PRIMARY KEY,
        dept:                   VARCHAR     NOT NULL
);

CREATE TABLE EmploysEmployee2(
        hireDate:               DATE        PRIMARY KEY,
        salary:                 INT         NOT NULL
);

CREATE TABLE EmploysEmployee3(
        employeeID:             VARCHAR     PRIMARY KEY,
        employeeName:           VARCHAR     NOT NULL,
        SIN:                    VARCHAR     NOT NULL,
        hireDate:               DATE        NOT NULL,
        labelName:              VARCHAR     NOT NULL,
        role:                   VARCHAR     NOT NULL,
        FOREIGN KEY (labelName) REFERENCES RecordLabel(labelName) ON UPDATE
CASCADE, ON DELETE CASCADE
        UNIQUE(SIN)
);

CREATE TABLE WritesContract1(
        type:                   VARCHAR     PRIMARY KEY,
        compensation:               INT             NOT NULL
);

CREATE TABLE WritesContract2(
        contractID:             VARCHAR     PRIMARY KEY,
```

```
        type:               VARCHAR     NOT NULL,
        stageName:          VARCHAR     NOT NULL, UNIQUE,
        labelName:          VARCHAR     NOT NULL,
        startDate:          DATE        NOT NULL,
        endDate:            DATE
        FOREIGN KEY (labelName) REFERENCES RecordLabel(labelName) ON DELETE
CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (stageName) REFERENCES Artist(stageName) ON DELETE CASCADE ON
UPDATE CASCADE
);

CREATE TABLE ArtistSigns1(
        legalName:          VARCHAR     PRIMARY KEY,
        dateOfBirth:        DATE        NOT NULL
);

CREATE TABLE ArtistSigns2(
        stageName:          VARCHAR     PRIMARY KEY,
        legalName:          VARCHAR     NOT NULL,
        contractID:         VARCHAR     NOT NULL, UNIQUE,
        FOREIGN KEY (contractID) REFERENCES WritesContract(contractId) ON DELETE
CASCADE
);


CREATE TABLE CollaboratesWith(
        professionalName    VARCHAR,
        stageName           VARCHAR,
        PRIMARY KEY (professionalName, stageName),
        FOREIGN KEY (professionalName) REFERENCES MusicProfessional(professionalName),
        FOREIGN KEY (stageName) REFERENCES Artist(stageName)
);



CREATE TABLE Attends (
        showName            VARCHAR,
```

```
    showDate            DATE,
    stageName           VARCHAR,
    numNominations      INT,
    awardsWon           INT,
    PRIMARY KEY (showName, showDate, stageName),
);


CREATE TABLE MusicProfessional(
    professionalName    VARCHAR     PRIMARY KEY,
    yearsOfExp          INT
);

CREATE TABLE Writer(
    professionalName    VARCHAR     PRIMARY KEY,
    numOfSongs          INT,
    FOREIGN KEY (professionalName) REFERENCES MusicProfessional(professionalName)
);

CREATE TABLE Producer(
    professionalName    VARCHAR     PRIMARY KEY,
    productionStyle     VARCHAR,
    FOREIGN KEY (professionalName) REFERENCES MusicProfessional(professionalName)
);

CREATE TABLE Album(
    UPC                 INT         PRIMARY KEY,
    professionalName    VARCHAR,
    stageName           VARCHAR     NOT NULL
    title               VARCHAR     NOT NULL,
    numTracks           INT         NOT NULL,
    length              INTERVAL    NOT NULL,
    releaseDate         DATE        NOT NULL,
    copiesSold          INT,
    FOREIGN KEY (professionalName) REFERENCES Producer(professionalName) ON
DELETE SET NULL,
```

```
    FOREIGN KEY (stageName) REFERENCES Artist(stageName) ON DELETE CASCADE ON
UPDATE CASCADE,
    UNIQUE (stageName, title)
);

CREATE TABLE Song(
    ISRC            VARCHAR     PRIMARY KEY,
    title           VARCHAR     NOT NULL,
    releaseDate     DATE        NOT NULL,
    genre           VARCHAR     NOT NULL,
    duration        INTERVAL    NOT NULL,
    stageName       VARCHAR     NOT NULL,
    recordingDate   DATE        NOT NULL,
    studioName      VARCHAR     NOT NULL,
    location        VARCHAR     NOT NULL,
    UPC             INT         NOT NULL,
    FOREIGN KEY (stageName) REFERENCES Artist(stageName) ON DELETE CASCADE ON
UPDATE CASCADE,
    FOREIGN KEY (studioName, location) REFERENCES Studio(studioName, location) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (UPC) REFERENCES Album(UPC) ON DELETE CASCADE,
    UNIQUE (stageName, title)
);



CREATE TABLE Writes(
    professionalName    VARCHAR,
    ISRC                VARCHAR     NOT NULL,
    PRIMARY KEY (professionalName, ISRC)
    FOREIGN KEY (professionalName) REFERENCES Writer(professionalName),
    FOREIGN KEY (ISRC) REFERENCES Song(ISRC)
    );
```

- Need assertion for the total participation constraint

```
CREATE TABLE Studio(
        studioName          VARCHAR,
        location            VARCHAR,
        size                VARCHAR
        PRIMARY KEY (studioName, location)
);

CREATE TABLE Single(
        ISRC                VARCHAR     PRIMARY KEY,
        unitsSold           INT,
        FOREIGN KEY (ISRC) REFERENCES Song(ISRC)
);


CREATE TABLE AlbumTrack(
        ISRC                VARCHAR     PRIMARY KEY,
        TrackNo             INT,
        FOREIGN KEY (ISRC) REFERENCES Song(ISRC)
);


CREATE TABLE AwardShow(
        showVenue           VARCHAR,
        showDate            DATE,
        showName            VARCHAR,
        organizer           VARCHAR,
        PRIMARY KEY (showName, showDate)
);

CREATE TABLE PerformsAt(
        stageName           VARCHAR,
        concertDate         VARCHAR,
        concertVenue        VARCHAR,
        PRIMARY KEY (concertDate, concertVenue, stageName),
        FOREIGN KEY (concertDate, concertVenue) REFERENCES Concert(concertDate,
concertVenue)
        FOREIGN KEY (stageName) REFERENCES Artist(stageName)
```

```
);

CREATE TABLE Concert1(
        concertVenue:       VARCHAR, PRIMARY KEY,
        capacity:           INT
);

CREATE TABLE Concert2(
        concertVenue:       VARCHAR,
        concertDate:        DATE,
        ticketsSold:        INT,
        PRIMARY KEY (concertVenue, concertDate)
);

CREATE TABLE IssueTicket1(
        concertVenue:       VARCHAR,
        concertDate         DATE,
        ticketID            VARCHAR,
        PRIMARY KEY (concertVenue, concertDate, ticketID)
        FOREIGN KEY (concertVenue, concertDate) REFERENCES Concert(concertVenue,
concertDate) ON DELETE CASCADE ON UPDATE CASCADE,
);

CREATE TABLE IssueTicket2(
        ticketID            VARCHAR     PRIMARY KEY,
        seatNum             VARCHAR,
        price               INT,
);
```

INSERT Statements

```
INSERT INTO RecordLabel VALUES ('Sony Music', 1987);
INSERT INTO RecordLabel VALUES ('Warner Bros', 1958);
INSERT INTO RecordLabel VALUES ('Universal Music', 1934);
INSERT INTO RecordLabel VALUES ('Atlantic Records', 1947);
INSERT INTO RecordLabel VALUES ('Capitol Records', 1942);
INSERT INTO RecordLabel VALUES ('Republic Records', 1995);
INSERT INTO RecordLabel VALUES ('Interscope Records', 1990);
INSERT INTO RecordLabel VALUES ('Top Dawg Entertainment', 2004);
```

```
INSERT INTO EmploysEmployee1 VALUES ('Manager', 'HR');
INSERT INTO EmploysEmployee1 VALUES ('Producer', 'Production');
INSERT INTO EmploysEmployee1 VALUES ('Engineer', 'Studio');
INSERT INTO EmploysEmployee1 VALUES ('Marketing Head', 'Marketing');
INSERT INTO EmploysEmployee1 VALUES ('Artist Manager', 'Talent Management');
INSERT INTO EmploysEmployee1 VALUES ('Legal Advisor', 'Legal');
INSERT INTO EmploysEmployee1 VALUES ('Audio Engineer', 'Production');
INSERT INTO EmploysEmployee1 VALUES ('A&R Manager', 'Artist & Repertoire');
INSERT INTO EmploysEmployee1 VALUES ('Financial Analyst', 'Finance');
INSERT INTO EmploysEmployee1 VALUES ('Sound Designer', 'Studio');
INSERT INTO EmploysEmployee1 VALUES ('HR Manager', 'HR');
INSERT INTO EmploysEmployee1 VALUES ('Marketing Specialist', 'Marketing');
INSERT INTO EmploysEmployee1 VALUES ('Sales Manager', 'Sales');
INSERT INTO EmploysEmployee1 VALUES ('Social Media Manager', 'Marketing');
```

```
INSERT INTO EmploysEmployee2 VALUES ('2021-06-12', 55000);
INSERT INTO EmploysEmployee2 VALUES ('2019-05-30', 72000);
INSERT INTO EmploysEmployee2 VALUES ('2020-11-21', 60000);
INSERT INTO EmploysEmployee2 VALUES ('2018-08-13', 95000);
INSERT INTO EmploysEmployee2 VALUES ('2022-03-22', 47000);
INSERT INTO EmploysEmployee2 VALUES ('2017-04-15', 88000);
```

```
INSERT INTO EmploysEmployee2 VALUES ('2020-10-05', 67000);
INSERT INTO EmploysEmployee2 VALUES ('2021-02-12', 72000);
INSERT INTO EmploysEmployee2 VALUES ('2019-12-01', 85000);
INSERT INTO EmploysEmployee2 VALUES ('2020-09-14', 64000);
INSERT INTO EmploysEmployee2 VALUES ('2022-01-07', 56000);
INSERT INTO EmploysEmployee2 VALUES ('2018-11-23', 78000);
INSERT INTO EmploysEmployee2 VALUES ('2021-05-02', 82000);
INSERT INTO EmploysEmployee2 VALUES ('2022-06-18', 60000);
```

```
INSERT INTO EmploysEmployee3 VALUES ('E123', 'Harry Potter', '111-22-3333', '2021-06-12',
'Sony Music', 'Manager');
INSERT INTO EmploysEmployee3 VALUES ('E124', 'Hermione Granger', '222-33-4444',
'2019-05-30', 'Warner Bros', 'Producer');
INSERT INTO EmploysEmployee3 VALUES ('E125', 'Ron Weasley', '333-44-5555', '2020-11-21',
'Universal Music', 'Engineer');
INSERT INTO EmploysEmployee3 VALUES ('E126', 'Parvati Patil', '444-55-6666', '2018-08-13',
'Atlantic Records', 'Marketing Head');
INSERT INTO EmploysEmployee3 VALUES ('E127', 'Severus Snape', '555-66-7777', '2022-03-22',
'Capitol Records', 'Artist Manager');
INSERT INTO EmploysEmployee3 VALUES ('E128', 'Draco Malfoy', '666-77-8888', '2017-04-15',
'Sony Music', 'Legal Advisor');
INSERT INTO EmploysEmployee3 VALUES ('E129', 'Rubeus Hagrid', '777-88-9999', '2020-10-05',
'Warner Bros', 'Audio Engineer');
INSERT INTO EmploysEmployee3 VALUES ('E130', 'Sirius Black', '888-99-1111', '2021-02-12',
'Universal Music', 'A&R Manager');
INSERT INTO EmploysEmployee3 VALUES ('E131', 'Bellatrix Lestrange', '999-11-2222',
'2019-12-01', 'Atlantic Records', 'Financial Analyst');
INSERT INTO EmploysEmployee3 VALUES ('E132', 'Neville Longbottom', '111-22-3333',
'2020-09-14', 'Capitol Records', 'Sound Designer');
INSERT INTO EmploysEmployee3 VALUES ('E133', 'Luna Lovegood', '222-33-4444', '2022-01-07',
'Sony Music', 'HR Manager');
INSERT INTO EmploysEmployee3 VALUES ('E134', 'Minerva McGonagall', '333-44-5555',
'2018-11-23', 'Warner Bros', 'Marketing Specialist');
INSERT INTO EmploysEmployee3 VALUES ('E135', 'Ginny Weasley', '444-55-6666', '2021-05-02',
'Universal Music', 'Sales Manager');
```

INSERT INTO EmploysEmployee3 VALUES ('E136', 'Cho Chang', '555-66-7777', '2022-06-18', 'Atlantic Records', 'Social Media Manager');

INSERT INTO WritesContract1 VALUES ('Exclusive', 1000000);
INSERT INTO WritesContract1 VALUES ('Non-Exclusive', 750000);
INSERT INTO WritesContract1 VALUES ('Single-Album Deal', 500000);
INSERT INTO WritesContract1 VALUES ('Multi-Album Deal', 2000000);
INSERT INTO WritesContract1 VALUES ('Tour Support', 300000);

INSERT INTO WritesContract2 VALUES ('C101', 'Exclusive', 'Taylor Swift', 'Republic Records', '2018-01-01', '2024-12-31');
INSERT INTO WritesContract2 VALUES ('C102', 'Non-Exclusive', 'Ariana Grande', 'Republic Records', '2017-01-01', '2022-12-31');
INSERT INTO WritesContract2 VALUES ('C103', 'Single-Album Deal', 'Bruno Mars', 'Atlantic Records', '2020-01-01', NULL);
INSERT INTO WritesContract2 VALUES ('C104', 'Multi-Album Deal', 'Lady Gaga', 'Interscope Records', '2019-01-01', '2025-12-31');
INSERT INTO WritesContract2 VALUES ('C105', 'Tour Support', 'Billie Eilish', 'Interscope Records', '2022-01-01', '2023-12-31');

INSERT INTO ArtistSigns1 VALUES ('Taylor Swift', '1989-12-13');
INSERT INTO ArtistSigns1 VALUES ('Ariana Grande', '1993-06-26');
INSERT INTO ArtistSigns1 VALUES ('Bruno Mars', '1985-10-08');
INSERT INTO ArtistSigns1 VALUES ('Lady Gaga', '1986-03-28');
INSERT INTO ArtistSigns1 VALUES ('Billie Eilish', '2001-12-18');
INSERT INTO ArtistSigns1 VALUES ('Sabrina Carpenter', '1999-05-11');
INSERT INTO ArtistSigns1 VALUES ('Chappell Roan', '1998-02-19');
INSERT INTO ArtistSigns1 VALUES ('Kendrick Lamar', '1987-06-17');
INSERT INTO ArtistSigns1 VALUES ('Drake', '1986-10-24');
INSERT INTO ArtistSigns1 VALUES ('Linkin Park', '1996-03-01');

INSERT INTO ArtistSigns2 VALUES ('Taylor Swift', 'Taylor Swift', 'C101');
INSERT INTO ArtistSigns2 VALUES ('Ariana Grande', 'Ariana Grande-Butera', 'C102');
INSERT INTO ArtistSigns2 VALUES ('Bruno Mars', 'Peter Hernandez', 'C103');

```
INSERT INTO ArtistSigns2 VALUES ('Lady Gaga', 'Stefani Germanotta', 'C104');
INSERT INTO ArtistSigns2 VALUES ('Billie Eilish', 'Billie O’Connell, 'C105');

INSERT INTO CollaboratesWith VALUES ('Max Martin', 'Taylor Swift');
INSERT INTO CollaboratesWith VALUES ('Tommy Brown', 'Ariana Grande');
INSERT INTO CollaboratesWith VALUES ('Mark Ronson', 'Bruno Mars');
INSERT INTO CollaboratesWith VALUES ('RedOne', 'Lady Gaga');
INSERT INTO CollaboratesWith VALUES ('Finneas', 'Billie Eilish');

INSERT INTO Attends VALUES ('Grammy Awards', '2022-04-03', 'Taylor Swift', 6, 3);
INSERT INTO Attends VALUES ('Grammy Awards', '2022-04-03', 'Ariana Grande', 2, 1);
INSERT INTO Attends VALUES ('MTV Video Music Awards', '2024-03-15', 'Bruno Mars', 4, 3);
INSERT INTO Attends VALUES ('American Music Awards', '2016-11-02', 'Lady Gaga', 5, 2);
INSERT INTO Attends VALUES ('Billboard Music Awards', '2022-11-15', 'Billie Eilish', 7, 3);

INSERT INTO MusicProfessional VALUES ('Max Martin', 25);
INSERT INTO MusicProfessional VALUES ('Tommy Brown', 10);
INSERT INTO MusicProfessional VALUES ('Mark Ronson', 15);
INSERT INTO MusicProfessional VALUES ('RedOne', 12);
INSERT INTO MusicProfessional VALUES ('Finneas', 5);
INSERT INTO MusicProfessional VALUES ('Greg Kurstin', 20);
INSERT INTO MusicProfessional VALUES ('Jack Antonoff', 12);
INSERT INTO MusicProfessional VALUES ('Sounwave', 15);
INSERT INTO MusicProfessional VALUES ('Noah Goldstein', 10);
INSERT INTO MusicProfessional VALUES ('Rik Simpson', 8);

INSERT INTO Writer VALUES ('Max Martin', 100);
INSERT INTO Writer VALUES ('Tommy Brown', 50);
INSERT INTO Writer VALUES ('Mark Ronson', 80);
INSERT INTO Writer VALUES ('RedOne', 60);
INSERT INTO Writer VALUES ('Finneas', 40);
INSERT INTO Writer VALUES ('Jack Antonoff', 120);
INSERT INTO Writer VALUES ('Sounwave', 45);

INSERT INTO Producer VALUES ('Max Martin', 'Pop');
INSERT INTO Producer VALUES ('Tommy Brown', 'Pop');
INSERT INTO Producer VALUES ('Mark Ronson', 'Funk');
```

```
INSERT INTO Producer VALUES ('RedOne', 'Pop');
INSERT INTO Producer VALUES ('Finneas', 'Alternative');
INSERT INTO Producer VALUES ('Greg Kurstin', 'Pop');
INSERT INTO Producer VALUES ('Rik Simpson', 'Rock');

INSERT INTO Album VALUES (602435648583, 'Max Martin', 'Taylor Swift', 'Evermore', 15,
'01:01:00', '2020-12-11', 2000000);
INSERT INTO Album VALUES (602435864563, 'Tommy Brown', 'Ariana Grande', 'Positions', 14,
'00:41:00', '2020-10-30', 1200000);
INSERT INTO Album VALUES (075678662737, 'Mark Ronson', 'Bruno Mars', '24K Magic', 9,
'00:33:00', '2016-11-18', 1500000);
INSERT INTO Album VALUES (00602508854064, 'RedOne', 'Lady Gaga', 'Chromatica', 16,
'00:43:00', '2020-05-29', 1300000);
INSERT INTO Album VALUES (602438241644, 'Finneas', 'Billie Eilish', 'Happier Than Ever', 16,
'00:56:00', '2021-07-30', 1800000);

INSERT INTO Song VALUES ('USUG12004699', 'Willow', '2020-12-11', 'Pop', '00:03:35', 'Taylor
Swift', '2020-11-25', 'Capital Records Recording Studios', 'NY', 2001);
INSERT INTO Song VALUES (USUM72019412', 'Positions', '2020-10-30', 'Pop', '00:02:52', 'Ariana
Grande', '2020-09-15', 'Clear Lake Recording Studios', 'LA', 2002);
INSERT INTO Song VALUES ('USAT21602944', '24K Magic', '2016-11-18', 'Funk', '00:03:45',
'Bruno Mars', '2016-10-20', 'Electro-Vox', 'LA', 2003);
INSERT INTO Song VALUES ('USUM72004304', 'Rain on Me', '2020-05-29', 'Pop', '00:03:02', 'Lady
Gaga', '2020-03-15', 'Paramount Recording Studios', 'LA', 2004);
INSERT INTO Song VALUES ('USUM72105934', 'Your Power', '2021-07-30', 'Alternative',
'00:04:05', 'Billie Eilish', '2021-05-10', 'Sunset Sound', 'NY', 2005);

INSERT INTO Studio VALUES ('Capital Records Recording Studios', 'NY', '1200');
INSERT INTO Studio VALUES ('Clear Lake Recording Studios', 'LA', 500);
INSERT INTO Studio VALUES ('Electro-Vox', 'LA', 720);
INSERT INTO Studio VALUES ('Paramount Recording Studios', 'LA', 1500);
INSERT INTO Studio VALUES ('Sunset Sound', 'NY', 600);
INSERT INTO Studio VALUES ('Studio6', Hensen Recording Studios, 650);
INSERT INTO Studio VALUES ('Studio7', 'Miami Recording Studios', 823);
```

```
INSERT INTO Single VALUES ('GB9TP1800335', 500000);
INSERT INTO Single VALUES ('AUUS00625883', 1200000);
INSERT INTO Single VALUES ('GBARL9300135', 1500000);
INSERT INTO Single VALUES ('USDM31400016', 1000000);
INSERT INTO Single VALUES ('QZPEW2294134', 1800000);
INSERT INTO Single VALUES ('QZ5FN2060906, 300000);
INSERT INTO Single VALUES ('QZTAZ2366934', 700000);

INSERT INTO AlbumTrack VALUES ('USUG12002835', 1);
INSERT INTO AlbumTrack VALUES ('USUG12002836', 2);
INSERT INTO AlbumTrack VALUES ('USUG12002837', 3);
INSERT INTO AlbumTrack VALUES ('USUG12002838', 4);
INSERT INTO AlbumTrack VALUES ('USUG12002839', 5);
INSERT INTO AlbumTrack VALUES ('USUG12002840', 6);
INSERT INTO AlbumTrack VALUES ('USUG12002841', 7);

INSERT INTO AwardShow VALUES ('MGM Grand Garden Arena', '2022-04-03', 'Grammy Awards',
'Recording Academy');
INSERT INTO AwardShow VALUES ('Staples Center', '2021-01-31', 'Grammy Awards', 'Recording
Academy');
INSERT INTO AwardShow VALUES ('Radio City Music Hall', '2019-08-26', 'MTV Video Music
Awards', 'MTV');
INSERT INTO AwardShow VALUES ('Microsoft Theater', '2021-06-27', 'BET Awards', 'BET
Network');
INSERT INTO AwardShow VALUES ('Microsoft Theater', '2020-11-22', 'American Music Awards',
'ABC');
INSERT INTO AwardShow VALUES ('Barclays Center', '2018-08-20', 'MTV Video Music Awards',
'MTV');
INSERT INTO AwardShow VALUES ('Staples Center', '2023-02-05', 'Grammy Awards', 'Recording
Academy');

INSERT INTO PerformsAt VALUES ('Taylor Swift', '2023-07-22', 'Lumen Field');
INSERT INTO PerformsAt VALUES ('Ariana Grande', '2019-05-06', 'Staples Center');
INSERT INTO PerformsAt VALUES ('Bruno Mars', '2018-07-14', 'Hyde Park');
INSERT INTO PerformsAt VALUES ('Lady Gaga', '2022-09-10', 'Dodgers Stadium');
INSERT INTO PerformsAt VALUES ('Billie Eilish', '2022-03-12', 'Little Caesars Arena');
INSERT INTO PerformsAt VALUES ('Sabrina Carpenter', '2024-11-04', 'Pacific Coliseum');
```

```
INSERT INTO PerformsAt VALUES ('Drake', '2023-07-05', 'United Center');

INSERT INTO Concert1 VALUES ('Lumen Field', 72000);
INSERT INTO Concert1 VALUES ('Staples Center', '2019-05-06', 18000);
INSERT INTO Concert1 VALUES ('Hyde Park', '2018-07-14', 65000);
INSERT INTO Concert1 VALUES ('Dodgers Stadium', '2022-09-10', 70000);
INSERT INTO Concert1 VALUES ('Little Caesars Arena', '2022-03-12', 20000);
INSERT INTO Concert1 VALUES ('Pacific Coliseum', '2024-11-04', 17713);
INSERT INTO Concert1 VALUES ('United Center', '2023-07-05', 19000);

INSERT INTO Concert2 VALUES ('Lumen Field', '2023-07-22', 71500);
INSERT INTO Concert2 VALUES ('Staples Center', '2019-05-06', 17500);
INSERT INTO Concert2 VALUES ('Hyde Park', '2018-07-14', 64000);
INSERT INTO Concert2 VALUES ('Dodgers Stadium', '2022-09-10', 65000);
INSERT INTO Concert2 VALUES ('Little Caesars Arena', '2022-03-12', 19500);
INSERT INTO Concert2 VALUES ('Pacific Coliseum', '2024-11-04', 15500);
INSERT INTO Concert2 VALUES ('United Center', '2023-07-05', 18000);

INSERT INTO IssueTicket1 VALUES ('Lumen Field', '2023-07-22', 'D1201');
INSERT INTO IssueTicket1 VALUES ('Staples Center', '2019-05-06', 'F12302');
INSERT INTO IssueTicket1 VALUES ('Hyde Park', '2018-07-14', 'A12303');
INSERT INTO IssueTicket1 VALUES ('Dodgers Stadium', '2022-09-10', 'D12304');
INSERT INTO IssueTicket1 VALUES ('Little Caesars Arena', '2022-03-12', 'B123205');
INSERT INTO IssueTicket1 VALUES ('Pacific Coliseum', '2024-11-04', 'D12306');
INSERT INTO IssueTicket1 VALUES ('United Center', '2023-07-05', 'F12307');

INSERT INTO IssueTicket2 VALUES ('D1201', 'FLR3', 1500);
INSERT INTO IssueTicket2 VALUES ('F12302', 'B15', 250);
INSERT INTO IssueTicket2 VALUES ('A12303', 'C10', 350);
INSERT INTO IssueTicket2 VALUES ('D12304', 'D1', 500);
INSERT INTO IssueTicket2 VALUES ('B123205', 'E20', 400);
INSERT INTO IssueTicket2 VALUES ('D12306', 'F25', 200);
INSERT INTO IssueTicket2 VALUES ('F12307', 'G5', 275);
```