University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ____2____

Date: ___27/10/23_____

Group Number: ___126_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tyler Kerswell	52101672	z8v1d	tylerkerswell@gmail.com
Julie Ryu	16183253	t4v5i	julieryu2@gmail.com
Natalia Garcia-Arias	54821806	m1p7e	garciarias.natalia@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

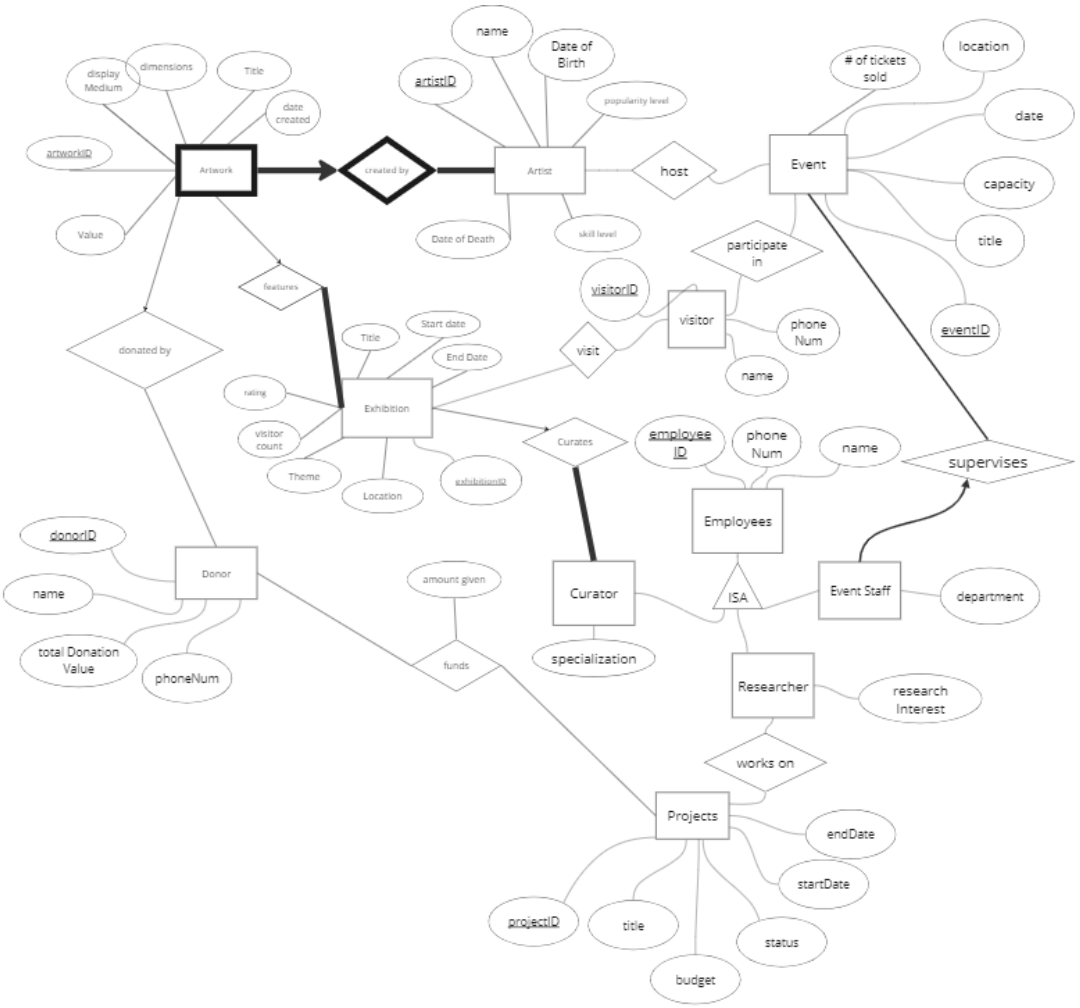
University of British Columbia, Vancouver
Department of Computer Science

Project Description

Project Domain: Art gallery operations and event management

Our database aims to organize data related to various aspects of art gallery operations including event organization, exhibit planning, employee management.
People will be able to manage their art gallery with this database. It will allow managers to keep track of events hosted at their art galleries as well as the artwork and exhibits they showcase there.

ER Diagram



Changes Made:

- We have changed the relationship between donor and funds to be many-to-many, multiple donors should be able to fund one event and vice-versa.

University of British Columbia, Vancouver

Department of Computer Science

- Removed donation type and donation date, added donation value to donor entity to add the ability for a single donor to donate multiple things
- Added ratings to exhibitions
- Added skill level and popularity level to artists
- Added participation constraint between event entity and supervises relationship to ensure every event is supervised by an event staff
- Changed contactInfo to phonenum

Schemas

Artist(artistID: int, name: char(32), Date of Birth: char(30), Date of Death: char(30), popularity level: int, skill level: int)

CANDIDATE KEYS: - {name, Date of Birth}
- artistID

PRIMARY KEY (artistID)
name NOT NULL
Date of Birth NOT NULL
UNIQUE (name, date of birth)

Artwork(artworkID: int, artistID: int, dimensions: char(40), title: char(100), date created: char(40), display medium: char(60), value: int, donorID: int, featureID: int)

CANDIDATE KEYS:
- {title, artistID}
- {artworkID, artistID}

PRIMARY KEY (artworkID, artistID)
FORIGN KEY (artistID) REFERENCES Artist(artistID)
ON DELETE CASCADE
FORIGN KEY (donorID) REFERENCES Donor(donorID)
ON DELETE SET NULL
FORIGN KEY (features) REFERENCES Exhibition(exhibitionID)
ON DELETE SET NULL
ON UPDATE CASCADE
date created NOT NULL
display medium NOT NULL
title NOT NULL
UNIQUE (date created, title, artistID)

Donor(donorID: int, name: char(50), Total Donation Value: int, phone number: char(12))

CANDIDATE KEYS: -donorID
PRIMARY KEY (donorID)
Total Donation Value NOT NULL
phone number NOT NULL

Exhibition(exhibitionID: int, title: char(100), start date: char(50), end date: char(50), visitor count: int, theme: char(80), location: char(70), curatorID: int, rating: int)

CANDIDATE KEYS:
- {start date, location}

- exhibitionID
PRIMARY KEY (exhibitionID)
FORIGN KEY (curatorID) REFERENCES Curator(employeeID)
ON UPDATE CASCADE
ON DELETE SET DEFAULT
UNIQUE (start date, location)

Visitor(name: varchar, phoneNum: integer, visitorID: integer, eventID: integer, exhibitionID: integer)
PK: visitorID
FK: eventID, exhibitionID
CK: (name, phoneNum), visitorID, UNIQUE: (name, phoneNum)

Event(ticketsSold: integer, location: varchar, date: varchar, capacity: integer, title: varchar, eventID: integer, employeeID: integer)
PK: eventID
FK: employeeID
CK: (title, date, location), UNIQUE: (title,date,location)

ParticipateIn(eventID: integer, visitorID: integer)
PK: eventID, visitorID
FK: eventID, visitorID

Host(eventID: integer, artistID: integer)
PK: artistID, eventID
FK: artistID, eventID

Visit(visitorID: integer, exhibitionID: integer)
PK: visitorID, exhibitionID
FK: visitorID, exhibitionID

Employees(employeeID: integer, phoneNum: varchar, name: varchar)
PK: employeeID
CK: (phoneNum, name)
UNIQUE: (phoneNum, name)

Curator(employeeID: integer, specialization: varchar)
PK: employeeID
FK: employeeID
*Curator's total participation constraint cannot be indicated with NOT NULL because it is in a many to many relationship.

EventStaffSupervises(employeeID: integer, department: varchar, eventID: integer)
PK: employeeID
FK: employeeID, eventID

University of British Columbia, Vancouver

Department of Computer Science

Researcher(employeeID: integer, researchInterest: varchar)
PK: employeeID
FK: employeeID

worksOn(employeeID: integer, projectID: varchar)
PK: (employeeID, projectID)
FK: employeeID, projectID

Projects(projectID: integer, title: varchar, budget: double, status: varchar, startDate: varchar, endDate: varchar)
PK: projectID
CK: (title, status)
UNIQUE: (title, status)

Funds(donorID: integer, projectID: integer, amountgiven: double)
PK: (donorID, projectID)
FK: donorID, projectID

Functional Dependencies

artist: skill level -> popularity level
name, date of birth -> date of Death, skill level, artistID
artistID -> name, date of birth, date of death, skill level, popularity level

artwork:
title, artistID -> value, date created, dimensions, date created, donorID, featureID, arkworkID

artistID, artworkID -> title, value, date created, dimensions, date created, donorID, featureID

date created, display medium -> value

Donor:
donorId -> name, phone number, total donation value
phone number -> name

Exhibition:
exhibitionID -> title, start date, end date, visitor count, theme, location, curatorID, rating

start date, location -> title, end date, visitor count, theme, curatorID, rating, exhibitionID

curator, title -> theme

University of British Columbia, Vancouver
Department of Computer Science

Visitor(name: varchar, phoneNum: integer, visitorID: integer, eventID: integer, exhibitionID: integer)

FDs:
- visitorID -> phoneNum, vName, eventID, exhibitionID
- name, phoneNum -> visitorID

Event(ticketsSold: integer, location: varchar, date: varchar, capacity: integer, title: varchar, eventID: integer, employeeID: integer)

FDs:
- eventID -> title, capacity, date, location, ticketsSold
- etitle, edate, elocation -> eventID
- eventID → employeeID

ParticipateIn(eventID: integer, visitorID: integer)

FDs:
- eventID, visitorID → eventID, visitorID

Host(eventID: integer, artistID: integer)

FDs:
- artistID, eventID → artistID, eventID

Visit(visitorID: integer, exhibitionID: integer)
- visitorID, exhibitionID → visitorID, exhibitionID

Employees(employeeID: integer, phoneNum: varchar, name: varchar)
FDs:
employeeID -> employeeID, phoneNum, name
phoneNum, name -> employeeID, phoneNum, name

Curator(employeeID: integer, specialization: varchar)
FDs:
employeeID -> employeeID, specialization

EventStaffSupervises(employeeID: integer, department: varchar, eventID: integer)
FDs:
employeeID -> employeeID, department, eventID

Researcher(employeeID: integer, researchInterest: varchar)
FDs:

employeeID -> employeeID, researchInterest

worksOn(employeeID: integer, projectID: varchar)
FDs:
employeeID, projectID -> employeeID, projectID

Projects(projectID: integer, title: varchar, budget: double, status: varchar, startDate: varchar, endDate: varchar)
FDs:
projectID -> projectID, title, budget, status, startDate, endDate
title, status -> projectID, title, budget, status, startDate, endDate
startDate, endDate -> status

Funds(donorID: integer, projectID: integer, amountgiven: double)
FDs:
donorID -> donorID, projectId, amountgiven

Normalization

6-1. Show work for normalizing relations not in BCNF/3NF

artist:
the decomposition to make artist BCNF:
R_1(popularity level, skill level)
R_2(skill level, artistID, name, date of birth, date of death)

artwork:

BCNF:
R_1(value, date created, display medium)
R_2(title, date created, dimensions, date created, donorID, featureID, artistID, artworkID)

donor:
BCNF:
R_1(phone number, name)
R_2(donorID, phone number, total donation value)

Exhibition:
BCNF:
R_1(curatorID, title, theme)
R_2(title, start date, end date, visitor count, location, curatorID, rating, exhibitionID)

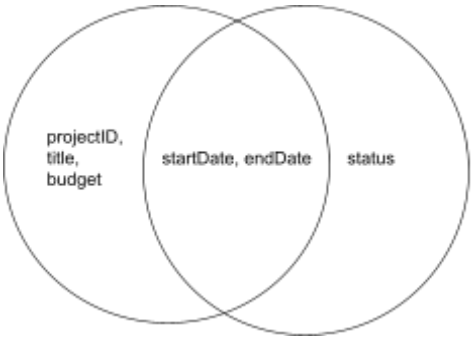
University of British Columbia, Vancouver
Department of Computer Science

Projects(projectID, title, budge, status, startDate, endDate)
Keys:
projectID, {title, status, startDate, endDate}
FDs:
projectID -> projectID, title, budget, status, startDate, endDate
title, status -> projectID, title, budget, status, startDate, endDate
startDate, endDate -> status

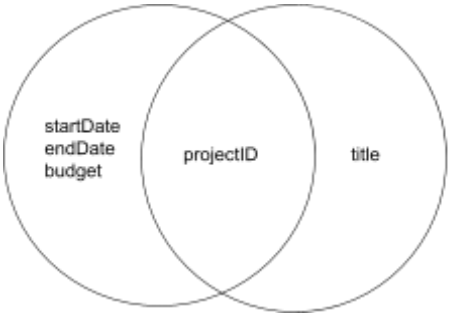
Closures:
projectID+ = {projectID, title, budget, status, startDate, endDate}
(title, status)+ = {projectID, title, budget, status, startDate, endDate}
(startDate, endDate)+ = {startDate, endDate, status}

The relation Projects is in neither BCNF or 3NF: in the FD startDate, endDate -> status, the LHS is not a superkey, and the attribute on the RHS is not part of a key either.

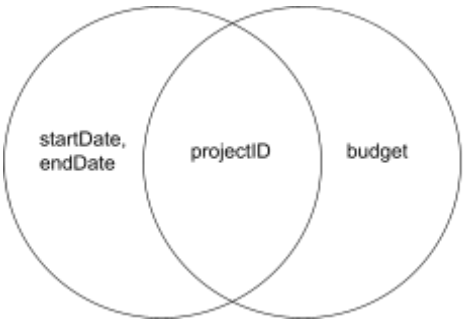
Thus, we decompose on startDate, endDate -> status



R1(startDate, endDate, status), R2(startDate, endDate, projectID, title, budget)
R2 is not in BCNF, so decompose on FD projectID -> title



R3(projectID, title), R4(projectID, startDate, endDate, budget)
R4 is not in BCNF, so decompose on FD projectID -> budget



R5(projectID, budget), R6(projectID, startDate, endDate)

Final relations: R1(startDate, endDate, status), R3(projectID, title), R5(projectID, budget)

6-2. list of all tables, including ones post normalization

artist:
the decomposition to make artist BCNF:
Artist_1(popularity level, skill level) PK: skill level FK: skill level
Artist_2(skill level, artistID, name, date of birth, date of death) PK: artistID CK: {name, date of birth}

artwork:

BCNF:
Artwork_1(value, date created, display medium) PK: {value, date created} FK: {value, date created}
Artwork_2(title, date created, dimensions, date created, donorID, featureID, artistID, artworkID) PK: {artworkID, artistID} FK: donorID, featureID, artistID CK: {title, artistID}

donor:
BCNF:
Donor_1(phone number, name) PK: phone number FK: phone number
Donor_2(donorID, phone number, total donation value) PK: donorID

Exhibition:
BCNF:
Exhibition_1(curatorID, title, theme) PK: {curatorID, title} FK: {curatorID, title}
Exhibition_2(title, start date, end date, visitor count, location, curatorID, rating, exhibitionID) PK: exhibitionID CK: {start date, location} FK: curatorID

Visitor(name: varchar, phoneNum: integer, visitorID: integer, eventID: integer, exhibitionID: integer)
PK: visitorID
FK: eventID, exhibitionID
CK: (name, phoneNum), visitorID, UNIQUE: (name, phoneNum)

University of British Columbia, Vancouver

Department of Computer Science

Event(ticketsSold: integer, location: varchar, date: varchar, capacity: integer, title: varchar, eventID: integer, employeeID: integer)
PK: eventID
FK: employeeID
CK: (title, date, location), UNIQUE: (title,date,location)

ParticipateIn(eventID: integer, visitorID: integer)
PK: eventID, visitorID
FK: eventID, visitorID

Host(eventID: integer, artistID: integer)
PK: artistID, eventID
FK: artistID, eventID

Visit(visitorID: integer, exhibitionID: integer)
PK: visitorID, exhibitionID
FK: visitorID, exhibitionID

Employees(employeeID: integer, phoneNum: varchar, name: varchar)
PK: employeeID
CK: (phoneNum, name)

Curator(employeeID: integer, specialization: varchar)
PK: employeeID
FK: employeeID

EventStaffSupervises(employeeID: integer, department: varchar, eventID: integer)
PK: employeeID
FK: employeeID, eventID

Researcher(employeeID: integer, researchInterest: varchar)
PK: employeeID
FK: employeeID

worksOn(employeeID: integer, projectID: varchar, startDate: varchar, endDate: varchar)
PK: (employeeID, projectID, startDate, endDate)
FK: employeeID, projectID, (startDate, endDate)

DateStatus(startDate: varchar, endDate:varchar, status: varchar)
PK: (startDate,endDate)

ProjectTitle(projectID: integer, title: varchar)
PK: projectID

ProjectBudget(projectID: integer, budget: double)
PK: projectID

University of British Columbia, Vancouver
Department of Computer Science

Funds(donorID: integer, projectID: integer, startDate: varchar, endDate: varchar, amountgiven: double)
PK: (donorID, projectID, startDate, endDate)
FK: donorID, projectID, (startDate, endDate)

SQL DDL Statements

```
CREATE TABLE Artist1 (
        popularityLevel INTEGER,
        skillLevel INTEGER,
        PRIMARY KEY (skillLevel),
        FORIGN KEY (skillLevel) REFERENCES Artist2(skillLevel))

CREATE TABLE Artist2 (
        artistID INTEGER,
        name VARCHAR NOT NULL,
        DateOfBirth VARCHAR NOT NULL,
        DateOfDeath VARCHAR,
        skillLevel INTEGER,
        PRIMARY KEY (artistID),
        UNIQUE (name, date of birth))

CREATE TABLE Artwork1 (
        value INTEGER,
        dateCreated VARCHAR,
        displayMedium VARCHAR NOT NULL,
        PRIMARY KEY (dateCreated, displayMedium),
        FORIGN KEY (dateCreated, displayMedium) REFERENCES Artwork2(dateCreated,
displayMedium))


CREATE TABLE Artwork2 (
        artworkID INTEGER,
        artistID INTEGER,
        title VARCHAR NOT NULL,
        dimensions VARCHAR,
        dateCreated VARCHAR NOT NULL,
        displayMedium VARCHAR NOT NULL,
        donorID INTEGER,
        featureID INTEGER,
        PRIMARY KEY (artworkID, artistID),
        FORIGN KEY (artistID) REFERENCES Artist(artistID)
                ON DELETE CASCADE,
        FORIGN KEY (donorID) REFERENCES Donor(donorID)
```

```
            ON DELETE SET NULL,
        FORIGN KEY (features) REFERENCES Exhibition(exhibitionID)
            ON DELETE SET NULL
            ON UPDATE CASCADE,
        UNIQUE (date created, title, artistID))

CREATE TABLE Donor1 (
        phoneNum INTEGER,
        name VARCHAR NOT NULL,
        PRIMARY KEY (phoneNum),
        FORIGN KEY (phoneNume) REFERENCES Donor2(phoneNum))

CREATE TABLE Donor2 (
        donorID INTEGER,
        TotalDonationValue INTEGER NOT NULL,
        phoneNumber VARCHAR NOT NULL,
        PRIMARY KEY (donorID))
CREATE TABLE Exhibition1 (
        curatorID INTEGER,
        title VARCHAR,
        theme VARCHAR NOT NULL,
        PRIMARY KEY (curatorID, title),
        FORIGN KEY (curatorID, title) REFERENCES Exhibition2(curatorID, title))

CREATE TABLE Exhibition2 (
        exhibitionID INTEGER,
        title VARCHAR,
        startDate VARCHAR,
        endDate VARCHAR,
        visitor count INTEGER,
        location VARCHAR,
        curatorID INTEGER,
        rating INTEGER,
        PRIMARY KEY (exhibitionID),
        FORIGN KEY (curatorID) REFERENCES Curator(employeeID)
            ON UPDATE CASCADE
            ON DELETE SET DEFAULT,
        UNIQUE (startDate, location))


CREATE TABLE Visitor (
   visitorID INTEGER PRIMARY KEY,
   name VARCHAR,
   phoneNum INTEGER,
   eventID INTEGER,
   exhibitionID INTEGER,
   UNIQUE (name, phoneNum),
   FOREIGN KEY (eventID) REFERENCES Event(eventID),
```

```
    FOREIGN KEY (exhibitionID) REFERENCES Exhibition(exhibitionID)
)

CREATE TABLE Event (
    eventID INTEGER PRIMARY KEY,
    ticketsSold INTEGER,
    location VARCHAR,
    date VARCHAR,
    capacity INTEGER,
    title VARCHAR,
    employeeID INTEGER,
    UNIQUE (title, date, location),
    FOREIGN KEY (employeeID) REFERENCES Employee(employeeID)
)

CREATE TABLE ParticipateIn (
    eventID INTEGER,
    visitorID INTEGER,
    PRIMARY KEY (eventID, visitorID),
    FOREIGN KEY (eventID) REFERENCES Event(eventID),
    FOREIGN KEY (visitorID) REFERENCES Visitor(visitorID)
)

CREATE TABLE Host (
    eventID INTEGER,
    artistID INTEGER,
    PRIMARY KEY (eventID, artistID),
    FOREIGN KEY (eventID) REFERENCES Event(eventID),
    FOREIGN KEY (artistID) REFERENCES Artist(artistID)
)

CREATE TABLE Visit (
    visitorID INTEGER,
    exhibitionID INTEGER,
    PRIMARY KEY (visitorID, exhibitionID),
    FOREIGN KEY (visitorID) REFERENCES Visitor(visitorID),
    FOREIGN KEY (exhibitionID) REFERENCES Exhibition(exhibitionID)
)


CREATE TABLE Employees
(employeeID INTEGER PRIMARY KEY,
 phoneNum VARCHAR,
 name VARCHAR,
 UNIQUE (phoneNum, name))

CREATE TABLE Curator
(employeeID INTEGER PRIMARY KEY,
```

```
 specialization VARCHAR,
FOREIGN KEY (employeeID) REFERENCES Employees(employeeID))

CREATE TABLE EventStaffSupervises
(employeeID INTEGER PRIMARY KEY,
 department VARCHAR,
 eventID INTEGER NOT NULL,
 FOREIGN KEY (employeeID) REFERENCES Employees,
 FOREIGN KEY (eventID) REFERENCES Event(eventID))

CREATE TABLE Researcher
(employeeID INTEGER PRIMARY KEY,
 researchInterest VARCHAR,
FOREIGN KEY (employeeID) REFERENCES Employees(employeeID))

CREATE TABLE worksOn
(employeeID INTEGER,
 projectID VARCHAR,
 startDate VARCHAR,
 endDate VARCHAR,
 PRIMARY KEY (employeeID, projectID, StartDate, endDate),
 FOREIGN KEY (employeeID) REFERENCES Employees(employeeID)),
 FOREIGN KEY (ProjectID) REFERENCES ProjectTitle(projectID)
 FOREIGN KEY (ProjectID REFERENCES ProjectBudget(projectID)
 FOREIGN KEY (startDate, endDate) REFERENCES DateStatus)

CREATE TABLE DateStatus
(startDate VARCHAR,
 endDate VARCHAR,
 status VARCHAR,
 PRIMARY KEY (startDate, endDate))

CREATE TABLE ProjectTitle
(projectID PRIMARY KEY,
 title VARCHAR)

CREATE TABLE ProjectBudget
(projectID INTEGER PRIMARY KEY,
 budget DOUBLE)


CREATE TABLE Funds
(donorID INTEGER,
 projectID INTEGER,
 startDate VARCHAR,
 endDate VARCHAR,
 amountgiven DOUBLE,
 PRIMARY KEY (donorID, projectID, startDate, endDate)
```

FOREIGN KEY (ProjectID) REFERENCES ProjectTitle(projectID)
FOREIGN KEY (ProjectID) REFERENCES ProjectBudget(projectID)
FOREIGN KEY (startDate, endDate) REFERENCES DateStatus)

INSERT Statements

INSERT INTO Artist1(popularityLevel, skillLevel)
VALUES (89, 76),
(100, 96),
(68, 86),
(96, 92),
(53, 68)

INSERT INTO Artist2(artistID, name, DateOfBirth, DateOfDeath, skillLevel)
VALUES (51, 'Andy Warhol', 'August 6, 1928', 'February 22, 1987', 76),
(52, 'Leonardo Da Vinci', 'April 15, 1452', 'May 2, 1519', 96),
(53, 'Pablo Picasso', 'October 25, 1881', 'April 8, 1973', 86),
(54, 'Vincent van Gogh', 'March 30, 1853', 'July 29, 1890', 92),
(55, 'Gerhard Richter', ''February 9, 1932', NULL, 68)

INSERT INTO Artwork1(value, dateCreated, displayMedium)
VALUES (78000000, '1962', 'painting'),
(5780000000, '1510', 'painting'),
(88000000, '1962', 'painting'),
(9000000, '1982','painting'),
(54000000, '1885', 'painting')

INSERT INTO Artwork2(artworkID, artistID, title, dimensions, dateCreated, value, donorID, featureID)
VALUES (91, 51, 'Green Coca-Cola Bottles', '82.8 in by 57.1 in', '1962', 78000000, 14, 302),
(92, 52, 'The Virgin and Child with Saint Anne', '51 in by 66.3 in', '1510', 5780000000, 13, 301),
(93, 51, 'Big Campbell's Soup Can 19c (Beef Noodle)', '8.3 x 4.3 in.', '1962', 88000000, 11, 303),
(94, 55, 'Two Candles (Zwei Kerzen)', '47 1/4 × 39 1/2 in.', ''1982', 9000000, 12, 305),
(95, 54, 'The Potato Eaters', '82 x 114 cm', '1885', 54000000, 14, 304)

INSERT INTO Donor1 (phoneNumber, name)
VALUES ('473-928-7782', 'Billy Bob),
('110-273-6271', 'Jim Jones'),
('182-999-9085', 'Harry Dewit'),

('111-521-1829', 'Gustavo Fring'),
('902-892-7782', 'Jonas Cruz')

INSERT INTO Donor2 (donorID, TotalDonationValue, phoneNumber)
VALUES (11, 64000000, '473-928-7782'),
(12, 109000000, '110-273-6271'),
(13, 1112000, '182-999-9085'),
(14, 810200000, '111-521-1829'),
(15, 7718000, '902-892-7782')

INSERT INTO Exhibition1(curatorID, title, theme)
VALUES (2000, 'Emerging Echoes: New Artists Unveiled', 'nouveau'),
(2004, 'Urban Odyssey: Cityscape Creations', 'cities'),
(2003, '3D Art Extravaganza', '3D art'),
(2002, 'Echoes of the Past: Historical Art Revival', 'classical'),
(2001, 'Art for Environmental Change', 'green art')

INSERT INTO Exhibition(exhibitionID, title, startDate, endDate, visitor count, location, curatorID, rating)
VALUES (301, 'Emerging Echoes: New Artists Unveiled', '25/10/22', '9/8/23', 966178, 'room 15', 2000, 67),
(302, 'Urban Odyssey: Cityscape Creations', '15/7/15', '18/8/17', 67551092, 'Bianca Ballroom', 2004, 89),
(303, '3D Art Extravaganza', '18/2/18', '19/5/20', 11000000, 'Ferguson Garden', 2003, 99),
(304, 'Echoes of the Past: Historical Art Revival', '19/4/20', '20/7/25', 73827361, 'room 209', 2002, 78),
(305, 'Art for Environmental Change', '23/12/9', '15/6/17', 161657192, 'room 112', 2001, 92)

INSERT INTO Visitor (visitorID, name, phoneNum, eventID, exhibitionID)
VALUES
(1, 'John Smith', '123-456-7890', 1, 101),
(2, 'Alice Johnson', '987-654-3210', 2, 102),
(3, 'Bob Davis', '555-111-2222', 3, 103);

INSERT INTO Event (eventID, ticketsSold, location, date, capacity, title, employeeID)
VALUES
(1, 100, 'Venue A', '15/05/23', 150, 'Art Show A', 201),
(2, 200, 'Venue B', '20/06/23', 250, 'Concert B', 202),
(3, 50, 'Venue C', '10/07/23', 100, 'Exhibition C', 203);

INSERT INTO ParticipateIn (eventID, visitorID)
VALUES
(1, 1),
(1, 2),

```
    (2, 2),
    (3, 3);

INSERT INTO Host (eventID, artistID)
VALUES
    (1, 301),
    (2, 302),
    (3, 303);

INSERT INTO Visit (visitorID, exhibitionID)
VALUES
    (1, 101),
    (1, 102),
    (2, 102),
    (3, 103);


INSERT
INTO Employees(employeeID, phoneNum, name)
VALUES
(2000, '111-222-333', 'John Smith'),
(2001, '111-222-334', 'Daniel Lee'),
(2002, '111-222-335', 'Mary Jane'),
(2003, '111-222-336','Jordan Johnson'),
(2004, '111-222-337','Sarah Jones'),
(2005, '111-222-338', 'Michael Kim'),
(2006, '111-222-339','Bianca Ng'),
(2007, '111-222-340', 'Emma Watson'),
(2008, '111-222-341', 'Emma Stone'),
(2009, '111-222-342', 'Margot Robbie'),
(2010, '111-222-343,'Chris Hemsworth'),
(2011, '111-222-344', 'Chris Pratt'),
(2012, '111-222-345', 'Chris Pine'),
(2013, '111-222-346', 'Chris Brown'),
(2014, '111-222-347,'Chris Paul');


INSERT
INTO Curator(employeeID, specialization)
VALUES
(2000, 'Contemporary'),
(2001, 'Modernism'),
(2002, 'Post Impressionism'),
(2003, 'Naturalism'),
(2004, 'Renaissance');


INSERT
```

```
INTO EventStaffSupervises(employeeID, department, eventID)
VALUES
(2005, 'Education', 101),
(2006, 'Marketing','101),
(2007, 'Marketing', 102),
(2008, 'Finance', 102),
(2009, 'Communications', 103),

INSERT
INTO Researcher(employeeID, researchInterest)
VALUES
(2010, 'Art History'),
(2011, 'Collection Studies'),
(2012, 'Cultural Context'),
(2013, 'Market and Value'),
(2014, 'Conservation');

INSERT
INTO worksOn(employeeID, projectID, startDate, endDate)
VALUES
(2010, 1000, '23/12/21', '11/11/24'),
(2011, 1001, '01/2/20', '3/4/22'),
(2012, 1002, '20/6/24', '3/11/26'),
(2013, 1003, '4/5/22', '2/2/23'),
(2014, 1004, '1/1/26', '4/4/29');

INSERT
INTO DateStatus(startDate, endDate, status)
VALUES
('23/12/21', '11/11/24', 'ongoing'),
('01/2/20', '3/4/22', 'completed'),
('20/6/24', '3/11/26', 'not started'),
('4/5/22', '2/2/23', 'completed'),
('1/1/26', '4/4/29', 'not started');

INSERT
INTO ProjectTitle(projectID, title)
VALUES
(1000, 'Women artists in 19C'),
(1001, 'Asian Contemporary Art'),
(1002, 'Evolution of Printmaking Techniques'),
(1003, 'Digital Evolution in Conservation'),
(1004, 'Cultural Heritage Preservation');

INSERT
INTO ProjectBudget(projectID, budget)
VALUES
(1000, 10.0),
```

University of British Columbia, Vancouver

Department of Computer Science

```
(1001, 20.5),
(1002, 30.0),
(1003, 30.5),
(1004, 40.0);

CREATE TABLE DonorFunds
(donorID INTEGER PRIMARY KEY,
 projectID INTEGER,
 startDate VARCHAR,
 endDate VARCHAR,
 amountgiven DOUBLE,
 FOREIGN KEY (ProjectID) REFERENCES ProjectTitle(projectID)
 FOREIGN KEY (ProjectID) REFERENCES ProjectBudget(projectID)
 FOREIGN KEY (startDate, endDate) REFERENCES DateStatus)


INSERT
INTO DonorFunds(donorID, projectID, startDate, endDate, amountgiven)
VALUES
(4000, 1000, '23/12/21', '11/11/24', 10.0),
(4001, 1001, '01/2/20', '3/4/22', 20.0),
(4002, 1002, '20/6/24', '3/11/26',  30.0),
(4003, 1003, '4/5/22', '2/2/23', 40.0),
(4004, 1004, '1/1/26', '4/4/29', 50.0);
```