ISYS2120 Sem2, 2025

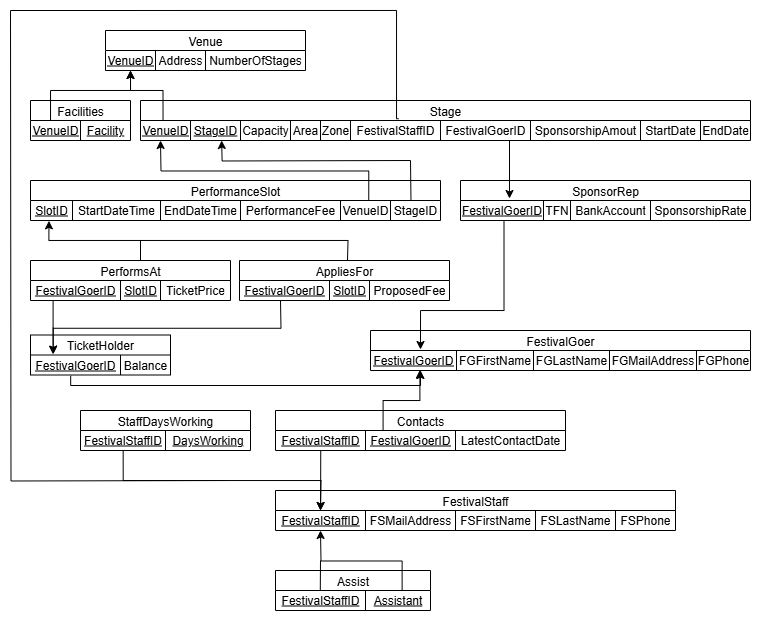
Assignment 1 (Relational schema)

540243105

SID:

PART A

Your relational schema diagram here (add additional pages as necessary)



PART B

Your SQL DDL here (exactly as in the .sql text file)

CREATE TABLE Venue (

    VenueID         INTEGER PRIMARY KEY,

    Address         VARCHAR(120) NOT NULL,   -- Address must exist

    NumberOfStages  INTEGER NOT NULL CHECK (NumberOfStages >= 0)

);

CREATE TABLE Facilities (

    VenueID   INTEGER REFERENCES Venue(VenueID),

    Facility  VARCHAR(60),

    PRIMARY KEY(VenueID, Facility)

);

CREATE TABLE FestivalGoer (

    FestivalGoerID  INTEGER PRIMARY KEY,

    FGFirstName     VARCHAR(40) NOT NULL,

    FGLastName      VARCHAR(40) NOT NULL,

    FGMailAddress   VARCHAR(120),

    FGPhone         VARCHAR(30)

);

CREATE TABLE TicketHolder (

    FestivalGoerID  INTEGER PRIMARY KEY REFERENCES FestivalGoer(FestivalGoerID),

    Balance         NUMERIC(10,2) CHECK (Balance >= 0)   -- Balance must be non-negative

);

CREATE TABLE SponsorRep (

    FestivalGoerID  INTEGER PRIMARY KEY REFERENCES FestivalGoer(FestivalGoerID),

    TFN             CHAR(9) UNIQUE CHECK (TFN ~ '^[0-9]{9}$'),   -- Tax File Number, must be unique

    BankAccount     VARCHAR(40) NOT NULL,

    SponsorshipRate NUMERIC(5,4)

);

CREATE TABLE FestivalStaff (

    FestivalStaffID INTEGER PRIMARY KEY,

    FSMailAddress   VARCHAR(120),

    FSFirstName     VARCHAR(40) NOT NULL,

    FSLastName      VARCHAR(40) NOT NULL,

    FSPhone         VARCHAR(30)

);

CREATE TABLE Assist (

    FestivalStaffID INTEGER REFERENCES FestivalStaff(FestivalStaffID),

    Assistant       INTEGER REFERENCES FestivalStaff(FestivalStaffID),

    PRIMARY KEY (FestivalStaffID, Assistant)

);

CREATE TABLE StaffDaysWorking (

    FestivalStaffID INTEGER REFERENCES FestivalStaff(FestivalStaffID),

    DaysWorking     VARCHAR(120),

    PRIMARY KEY (FestivalStaffID, DaysWorking)

);

CREATE TABLE Stage (

    VenueID   INTEGER REFERENCES Venue(VenueID),

    StageID   INTEGER,

    Capacity  INTEGER CHECK (Capacity IS NULL OR Capacity >= 0), -- Capacity can be NULL or non-negative

    Area      NUMERIC(10,2),

    Zone      VARCHAR(120),

    FestivalStaffID     INTEGER REFERENCES FestivalStaff(FestivalStaffID),

    FestivalGoerID      INTEGER REFERENCES SponsorRep(FestivalGoerID),

    SponsorshipAmount   NUMERIC(10,2) CHECK (SponsorshipAmount IS NULL OR SponsorshipAmount >= 0),

    StartDate           DATE,

    EndDate             DATE,

    PRIMARY KEY (VenueID, StageID),

    CHECK (EndDate IS NULL OR StartDate IS NULL OR EndDate >= StartDate)

);

CREATE TABLE PerformanceSlot (

    SlotID           INTEGER PRIMARY KEY,

    StartDateTime    TIMESTAMP NOT NULL,

    EndDateTime      TIMESTAMP NOT NULL,

    PerformanceFee   NUMERIC(10,2) CHECK (PerformanceFee IS NULL OR PerformanceFee >= 0),

    VenueID          INTEGER NOT NULL,

    StageID          INTEGER NOT NULL,

    FOREIGN KEY(VenueID, StageID) REFERENCES Stage(VenueID, StageID),

    CHECK (EndDateTime > StartDateTime)

);

CREATE TABLE AppliesFor (

    FestivalGoerID  INTEGER UNIQUE REFERENCES TicketHolder(FestivalGoerID),

    SlotID          INTEGER REFERENCES PerformanceSlot(SlotID),

    ProposedFee     NUMERIC(10,2) NOT NULL CHECK (ProposedFee >= 0),

    PRIMARY KEY (FestivalGoerID, SlotID)

);

CREATE TABLE PerformsAt (

    FestivalGoerID  INTEGER UNIQUE REFERENCES TicketHolder(FestivalGoerID),

    SlotID          INTEGER UNIQUE REFERENCES PerformanceSlot(SlotID),

    TicketPrice     NUMERIC(10,2) NOT NULL CHECK (TicketPrice >= 0),

    PRIMARY KEY (FestivalGoerID, SlotID)

);

CREATE TABLE Contacts (

    FestivalStaffID   INTEGER REFERENCES FestivalStaff(FestivalStaffID),

    FestivalGoerID    INTEGER REFERENCES FestivalGoer(FestivalGoerID),

    LatestContactDate DATE,

    PRIMARY KEY (FestivalStaffID, FestivalGoerID)

);