

```
/*
Project 1
Group 5
Bala Gaurav Reddy Pasam
Raja Krishna Srivastav Arra
Khushbu Singh
Altaf Khan
*/
/*
Project Discussion and Summary
```

Our project focused on creating UniversityDB, a streamlined database system that brings together key university operations—managing users, resources, events, and academic support—into a single, efficient structure. We started by designing the Users Table as the heart of the system, assigning roles like students, faculty, and alumni while introducing a unique Privilege Level concept. This level determines access to resources, borrowing limits, and campus facilities. Unlike traditional systems, we thought outside the box by linking privilege levels to resource access policies, ensuring fair usage and simplified management. It's a flexible approach that adapts to different user needs while maintaining control. The Resource Management System tracks books, e-books, and multimedia, ensuring resources are always accounted for. Smart triggers prevent double bookings and update availability automatically. For events and facilities, the system allows seamless booking of labs, study rooms, and workshops, handling conflicts and ensuring smooth scheduling. On the academic side, Support Services and Research Assistance modules connect students and researchers with staff for tutoring, consultations, and research guidance. This ensures that academic support is organized and accessible. Overall, our approach prioritized simplicity and scalability. By testing it with realistic data and embedding checks at every stage, we built a system that not only works efficiently but also handles the complexity of university life with ease. The privilege-level innovation, in particular, sets this system apart as a thoughtful and practical solution for access control.

```
*/
```

```
-- #####
-- DATABASE CREATION AND INITIALIZATION
-- #####
```

```
-- Drop the database if it exists
```

```
IF EXISTS (SELECT * FROM sys.databases WHERE name = 'UniversityDB')
BEGIN
    DROP DATABASE UniversityDB;
END;

-- Create the database
CREATE DATABASE UniversityDB;
GO

-- Use the new database
USE UniversityDB;
GO

-- #####
-- USER MANAGEMENT SCHEMA
-- #####

-- Users Table: Core user table capturing general details and privilege levels
CREATE TABLE Users (
    UniversalId VARCHAR(20) PRIMARY KEY NOT NULL,
    Name VARCHAR(100) NOT NULL,
    Role NVARCHAR(50) CHECK (Role IN ('Student', 'Faculty', 'Staff', 'Alumni',
    'Community')) NOT NULL,
    PrivilegeLevel INT CHECK (PrivilegeLevel >= 0) NOT NULL, -- Access level for
    resource borrowing
    MaxBorrowableItems INT DEFAULT 0 NOT NULL, -- Maximum borrowable items based on
    privilege
    CampusAccess NVARCHAR(50) CHECK (CampusAccess IN ('Own', 'All')) NOT NULL, --
    Campus access policy
    MembershipStatus NVARCHAR(50) CHECK (MembershipStatus IN ('Active',
    'Inactive')) DEFAULT 'Active' -- Global user status
);

-- Students Table: Additional details specific to students
CREATE TABLE Students (
    UniversalId VARCHAR(20) PRIMARY KEY REFERENCES Users(UniversalId),
    Email VARCHAR(100) UNIQUE NOT NULL,
    PhoneNumber VARCHAR(15),
    Address NVARCHAR(MAX),
    StudentLevel NVARCHAR(50) CHECK (StudentLevel IN ('Undergrad', 'Graduate',
    'PhD')) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE,
    Major NVARCHAR(50),
    CampusLocation NVARCHAR(50) CHECK (CampusLocation IN ('Stockton', 'Sacramento',
    'San Francisco'))
);

-- Faculty Table: Additional details specific to faculty members
```

```
CREATE TABLE Faculty (
    UniversalId VARCHAR(20) PRIMARY KEY REFERENCES Users(UniversalId),
    Email VARCHAR(100) UNIQUE NOT NULL,
    PhoneNumber VARCHAR(15),
    Address NVARCHAR(MAX),
    Department NVARCHAR(50),
    Position NVARCHAR(50) CHECK (Position IN ('Professor', 'TA', 'Dean', 'HOD')),
    OfficeNumber NVARCHAR(20),
    CampusLocation NVARCHAR(50) CHECK (CampusLocation IN ('Stockton', 'Sacramento', 
        'San Francisco'))
);

-- Staff Table: Additional details specific to staff members
CREATE TABLE Staff (
    UniversalId VARCHAR(20) PRIMARY KEY REFERENCES Users(UniversalId),
    Email VARCHAR(100) UNIQUE NOT NULL,
    PhoneNumber VARCHAR(15),
    Address NVARCHAR(MAX),
    JobTitle NVARCHAR(50) NOT NULL,
    Department NVARCHAR(50),
    CampusLocation NVARCHAR(50) CHECK (CampusLocation IN ('Stockton', 'Sacramento', 
        'San Francisco'))
);

-- Alumni Table: Additional details specific to alumni
CREATE TABLE Alumni (
    UniversalId VARCHAR(20) PRIMARY KEY REFERENCES Users(UniversalId),
    Email VARCHAR(100) UNIQUE NOT NULL,
    PhoneNumber VARCHAR(15),
    GraduationYear INT NOT NULL,
    Degree NVARCHAR(50) NOT NULL,
    AlumniStatus NVARCHAR(50) CHECK (AlumniStatus IN ('Active', 'Inactive')) NOT 
        NULL,
    CampusLocation NVARCHAR(50) CHECK (CampusLocation IN ('Stockton', 'Sacramento', 
        'San Francisco'))
);

-- Community Members Table: Additional details specific to community members
CREATE TABLE CommunityMembers (
    UniversalId VARCHAR(20) PRIMARY KEY REFERENCES Users(UniversalId),
    Email VARCHAR(100) UNIQUE NOT NULL,
    PhoneNumber VARCHAR(15),
    MembershipStartDate DATE NOT NULL,
    MembershipEndDate DATE,
    CampusLocation NVARCHAR(50) CHECK (CampusLocation IN ('Stockton', 'Sacramento', 
        'San Francisco'))
);

-- #####
```

```
-- RESOURCE MANAGEMENT SCHEMA
-- #####

-- Resources Table: Captures resource metadata and accessibility requirements
CREATE TABLE Resources (
    ResourceId INT PRIMARY KEY IDENTITY(1,1),
    Title NVARCHAR(255) NOT NULL,
    Author NVARCHAR(100),
    ResourceType NVARCHAR(50) CHECK (ResourceType IN ('Book', 'E-Book', 'Journal', 'Multimedia')) NOT NULL,
    PublicationDate DATE,
    Genre NVARCHAR(50),
    OverdueFinePerDay DECIMAL(10, 2) DEFAULT 0,
    RequiredAccessLevel INT CHECK (RequiredAccessLevel >= 0) NOT NULL, -- Minimum privilege level to access
    AvailabilityStatus NVARCHAR(50) CHECK (AvailabilityStatus IN ('Available', 'Unavailable')) DEFAULT 'Available' -- Overall availability
);

-- Libraries Table: Captures library branch details
CREATE TABLE Libraries (
    LibraryId INT PRIMARY KEY IDENTITY(1,1), --Each space including conference rooms has seperate libraryID
    LibraryName NVARCHAR(100) NOT NULL,
    BranchName NVARCHAR(100) NOT NULL,
    LocationDetails NVARCHAR(100) NOT NULL --Exact Location Address including room numbers or shelf and asile numbers
);

-- Resource Locations Table: Tracks the location and availability of resources
CREATE TABLE ResourceLocations (
    UnitId INT PRIMARY KEY IDENTITY(1,1),
    ResourceId INT NOT NULL REFERENCES Resources(ResourceId),
    LibraryId INT NOT NULL REFERENCES Libraries(LibraryId),
    CopiesAvailable INT DEFAULT 0,
    BorrowScope NVARCHAR(20) CHECK (BorrowScope IN ('OwnCampus', 'AllCampus')), -- Borrowing restrictions
    MinPrivilegeToCrossCampus INT CHECK (MinPrivilegeToCrossCampus >= 0) NOT NULL -- Access level required to lend to other Campus User.
);

-- #####
-- BORROWING AND RESERVATIONS SCHEMA
-- #####

-- Borrowing Records Table: Tracks borrowing activity and statuses
CREATE TABLE BorrowingRecords (
    BorrowingId INT PRIMARY KEY IDENTITY(1,1),
    ResourceId INT NOT NULL REFERENCES Resources(ResourceId),
```

```

    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    LibraryId INT NOT NULL REFERENCES Libraries(LibraryId),
    BorrowDate DATE NOT NULL,
    DueDate DATE NOT NULL,
    ReturnDate DATE,
    Status NVARCHAR(50) CHECK (Status IN ('Active', 'Returned', 'Overdue')) NOT
    NULL
);

```

-- Overdue Records Table: Tracks overdue borrowings and associated fines

```

CREATE TABLE OverdueRecords (
    OverdueId INT PRIMARY KEY IDENTITY(1,1),
    BorrowingId INT NOT NULL REFERENCES BorrowingRecords(BorrowingId),
    OverdueDays INT NOT NULL,
    FineAmount DECIMAL(10, 2) NOT NULL,
    DateOverdueCalculated DATE NOT NULL DEFAULT GETDATE()
);

```

-- Renewal Records Table: Tracks borrowing renewals

```

CREATE TABLE RenewalRecords (
    RenewalId INT PRIMARY KEY IDENTITY(1,1),
    RenewalCount INT CHECK (RenewalCount >=0 AND RenewalCount <= 3 ),
    BorrowingId INT NOT NULL REFERENCES BorrowingRecords(BorrowingId),
    RenewalDate DATE NOT NULL DEFAULT GETDATE(),
    NewDueDate DATE NOT NULL
);

```

-- Reservations Table: Tracks resource reservations

```

CREATE TABLE Reservations (
    ReservationId INT PRIMARY KEY IDENTITY(1,1),
    ResourceId INT NOT NULL REFERENCES Resources(ResourceId),
    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    LibraryId INT NOT NULL REFERENCES Libraries(LibraryId),
    ReservationDate DATE NOT NULL,
    PickupDate DATETIME NOT NULL,
    Status NVARCHAR(50) CHECK (Status IN ('Active', 'Cancelled', 'No response'))
    NOT NULL
);

```

-- #####

-- EVENT MANAGEMENT SCHEMA

-- #####

```

CREATE TABLE Events (
    EventId INT PRIMARY KEY IDENTITY(1,1),
    EventName NVARCHAR(100) NOT NULL,
    Description NVARCHAR(MAX),
    EventDate DATE NOT NULL,
    Starttime TIME NOT NULL,
    Endtime TIME NOT NULL,

```

```

    LocationId INT NOT NULL REFERENCES Libraries(LibraryId),
    MaxParticipants INT NOT NULL,
    EventStatus Varchar(20) CHECK (EventStatus IN
        ('Active', 'Cancelled', 'Completed')),
    CONSTRAINT CHK_Events_EndTime CHECK (EndTime > StartTime) -- Table-level CHECK
        constraint
);

```

```

CREATE TABLE EventReservations (
    Fullname varchar(20),
    ReservationId INT PRIMARY KEY IDENTITY(1,1),
    EventId INT NOT NULL REFERENCES Events(EventId),
    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    CONSTRAINT UniqueParticipantReservation UNIQUE (EventId, UniversalId)
);

```

```

-- #####
-- FACILITY MANAGEMENT SCHEMA
-- #####

```

```

-- Facilities Table: Captures facility details and availability

```

```

CREATE TABLE Facilities (
    FacilityId INT PRIMARY KEY IDENTITY(1,1),
    FacilityName NVARCHAR(100) NOT NULL,
    LocationId INT NOT NULL REFERENCES Libraries(LibraryId),
    FacilityType NVARCHAR(50) CHECK (FacilityType IN ('Study Room', 'Lab',
        'Conference Hall')) NOT NULL,
    SlotDuration INT DEFAULT 60 NOT NULL, -- Duration in minutes
    AvailabilityStatus NVARCHAR(50) CHECK (AvailabilityStatus IN ('Available',
        'Reserved', 'UnderMaintenance')) NOT NULL,
    RequiredLevel INT CHECK (RequiredLevel >= 0) NOT NULL, -- Minimum privilege
        level to access
    MaxOccupancy INT DEFAULT 1 CHECK (MaxOccupancy > 0) NOT NULL -- Maximum allowed
        occupants
);

```

```

-- Facility Reservations Table: Tracks reservations for facilities

```

```

CREATE TABLE FacilityReservations (
    ReservationId INT PRIMARY KEY IDENTITY(1,1),
    FacilityId INT NOT NULL REFERENCES Facilities(FacilityId),
    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    ReservationDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME,
    Status NVARCHAR(50) CHECK (Status IN ('Active', 'Cancelled', 'Completed')) NOT
        NULL,
    CONSTRAINT CHK_RESERVATION_EndTime CHECK (EndTime > StartTime) -- Table-level
        CHECK constraint

```

```
);

-- #####
-- ACADEMIC SUPPORT SERVICES SCHEMA
-- #####

-- Support Services Table: Captures academic support services offered
CREATE TABLE SupportServices (
    ServiceId INT PRIMARY KEY IDENTITY(1,1),
    ServiceName NVARCHAR(100) NOT NULL,
    ServiceType NVARCHAR(50) CHECK (ServiceType IN ('Tutoring', 'Writing
        Assistance', 'Advising', 'Consultation')) NOT NULL, -- Classify service types
    Description NVARCHAR(MAX)
);

-- Academic Support Staff Table: Tracks staff assigned to academic support
CREATE TABLE AcademicSupportStaff (
    SupportId INT PRIMARY KEY IDENTITY(1,1),
    StaffId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    ServiceId INT NOT NULL REFERENCES SupportServices(ServiceId),
    ProficiencyLevel NVARCHAR(50) CHECK (ProficiencyLevel IN ('Beginner',
        'Intermediate', 'Expert')) NOT NULL
);

-- Appointments Table: Tracks appointments for support services
CREATE TABLE Appointments (
    AppointmentId INT PRIMARY KEY IDENTITY(1,1),
    ServiceId INT NOT NULL REFERENCES SupportServices(ServiceId),
    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    StaffId VARCHAR(20) REFERENCES Users(UniversalId),
    AppointmentDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    LocationId INT NOT NULL REFERENCES Libraries(LibraryId),
    Status NVARCHAR(50) CHECK (Status IN ('Scheduled', 'Cancelled', 'Completed'))
    NOT NULL
);

-- #####
-- RESEARCH ASSISTANCE SCHEMA
-- #####

-- Research Topics Table: Captures research topics with category classification
CREATE TABLE ResearchTopics (
    TopicId INT PRIMARY KEY IDENTITY(1,1),
    TopicName NVARCHAR(100) NOT NULL,
    ResearchCategory NVARCHAR(50) CHECK (ResearchCategory IN ('STEM', 'Humanities',
        'Social Sciences', 'Arts', 'Business')) NOT NULL, -- Classify research
    topics
```

```
Description NVARCHAR(MAX)
);

-- Support Staff Table: Tracks staff proficiency for research topics
CREATE TABLE SupportStaff (
    SupportId INT PRIMARY KEY IDENTITY(1,1),
    StaffId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    TopicId INT NOT NULL REFERENCES ResearchTopics(TopicId),
    ProficiencyLevel NVARCHAR(50) CHECK (ProficiencyLevel IN ('Beginner',
        'Intermediate', 'Expert')) NOT NULL
);

-- Consultations Table: Tracks research consultations with feedback
CREATE TABLE Consultations (
    ConsultationId INT PRIMARY KEY IDENTITY(1,1),
    UniversalId VARCHAR(20) NOT NULL REFERENCES Users(UniversalId),
    StaffId VARCHAR(20) REFERENCES Users(UniversalId),
    TopicId INT NOT NULL REFERENCES ResearchTopics(TopicId),
    ConsultationDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    LocationId INT NOT NULL REFERENCES Libraries(LibraryId),
    Status NVARCHAR(50) CHECK (Status IN ('Scheduled', 'Cancelled', 'Completed'))
    NOT NULL,
    Feedback NVARCHAR(MAX)
);

-- #####
-- INDEX CREATION
-- #####

-- User Management
CREATE INDEX idx_users_universalid ON Users(UniversalId);
CREATE INDEX idx_users_role ON Users(Role);

-- Resource Management
CREATE INDEX idx_resources_resourceid ON Resources(ResourceId);
CREATE INDEX idx_resources_title ON Resources(Title);
CREATE INDEX idx_resources_author ON Resources(Author);

-- Borrowing and Reservations
CREATE INDEX idx_borrowingrecords_borrowingid ON BorrowingRecords(BorrowingId);
CREATE INDEX idx_borrowingrecords_universalid ON BorrowingRecords(UniversalId);
CREATE INDEX idx_reservations_universalid ON Reservations(UniversalId);

-- Event Management
CREATE INDEX idx_events_eventid ON Events(EventId);
CREATE INDEX idx_eventreservations_universalid ON EventReservations(UniversalId);
```



```
-- Facility Management
CREATE INDEX idx_facilities_facilityid ON Facilities(FacilityId);
CREATE INDEX idx_facilities_locationid ON Facilities(LocationId);

-- #####
-- TRIGGER CREATION
-- #####

-- Trigger: Update Resource Availability
GO
CREATE TRIGGER trg_update_resource_availability
ON ResourceLocations
AFTER INSERT, UPDATE
AS
BEGIN
    UPDATE Resources
    SET AvailabilityStatus =
        CASE
            WHEN (SELECT SUM(CopiesAvailable)
                  FROM ResourceLocations
                  WHERE ResourceLocations.ResourceId = Resources.ResourceId) = 0
            THEN 'Unavailable'
            ELSE 'Available'
        END
    WHERE ResourceId IN (SELECT ResourceId FROM inserted);
END;
GO

-- Trigger: Calculate Overdue Fines
GO
CREATE TRIGGER trg_calculate_overdue
ON BorrowingRecords
AFTER UPDATE
AS
BEGIN
    INSERT INTO OverdueRecords (BorrowingId, OverdueDays, FineAmount,
                                DateOverdueCalculated)
    SELECT BorrowingId, DATEDIFF(DAY, DueDate, GETDATE()),
           DATEDIFF(DAY, DueDate, GETDATE()) * r.OverdueFinePerDay, GETDATE()
    FROM BorrowingRecords br
    JOIN Resources r ON br.ResourceId = r.ResourceId
    WHERE br.Status = 'Overdue' AND br.ReturnDate IS NULL;
END;
GO

-- Trigger: Prevent Double Reservations
GO
CREATE TRIGGER trg_prevent_double_reservations
ON Reservations
```

```

INSTEAD OF INSERT
AS
BEGIN
    -- Check for conflicts between the new reservations and existing active
    reservations
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN Reservations r
        ON i.UniversalId = r.UniversalId
        AND i.ResourceId = r.ResourceId
        AND r.Status = 'Active'
    )
    BEGIN
        RAISERROR('User already has an active reservation for this resource.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        -- Insert the new reservations if no conflict is found
        INSERT INTO Reservations (ResourceId, UniversalId, LibraryId,
            ReservationDate, PickupDate, Status)
        SELECT ResourceId, UniversalId, LibraryId, ReservationDate, PickupDate,
            Status
        FROM inserted;
    END
END;

GO

--#####
---End Of Database Creation
--#####

--*****
----Populating Database
--*****
INSERT INTO Users (UniversalId, Name, Role, PrivilegeLevel, MaxBorrowableItems,
    CampusAccess, MembershipStatus) VALUES
('10001', 'John Doe', 'Student', 2, 5, 'Own', 'Active'),
('10002', 'Jane Smith', 'Faculty', 4, 10, 'All', 'Active'),
('10003', 'Mark Taylor', 'Staff', 3, 7, 'Own', 'Active'),
('10004', 'Emily Johnson', 'Alumni', 1, 2, 'Own', 'Inactive'),
('10005', 'Michael Brown', 'Community', 0, 1, 'Own', 'Active'),
('10006', 'Laura Wilson', 'Student', 2, 6, 'Own', 'Active'),
('10007', 'Chris Martin', 'Faculty', 5, 12, 'All', 'Active'),
('10008', 'Samantha Green', 'Staff', 3, 8, 'Own', 'Active'),

```

```
( '10009', 'Robert White', 'Alumni', 1, 3, 'Own', 'Inactive'),
( '10010', 'Jessica Black', 'Community', 0, 2, 'Own', 'Active'),
( '10011', 'David Gray', 'Student', 2, 5, 'Own', 'Active'),
( '10012', 'Amy Evans', 'Faculty', 4, 10, 'All', 'Active'),
( '10013', 'Rachel Scott', 'Staff', 3, 7, 'Own', 'Inactive'),
( '10014', 'Oliver Harris', 'Alumni', 1, 2, 'Own', 'Active'),
( '10015', 'Nathan King', 'Community', 0, 1, 'Own', 'Active'),
( '10016', 'Sophia Lewis', 'Student', 2, 6, 'Own', 'Active'),
( '10017', 'Mia Carter', 'Faculty', 5, 12, 'All', 'Active'),
( '10018', 'William Hughes', 'Staff', 3, 8, 'Own', 'Active'),
( '10019', 'Isabella Clark', 'Alumni', 1, 3, 'Own', 'Inactive'),
( '10020', 'Ethan Lopez', 'Community', 0, 2, 'Own', 'Active');
-----
-- #####
-- Populate Students Table
-- #####
INSERT INTO Students (UniversalId, Email, PhoneNumber, Address, StudentLevel,
StartDate, EndDate, Major, CampusLocation) VALUES
( '10001', 'john.doe@university.com', '1234567890', '123 Main St, Stockton',
'Undergrad', '2022-08-15', '2026-05-10', 'Computer Science', 'Stockton'),
( '10006', 'laura.wilson@university.com', '9876543210', '456 Elm St, Sacramento',
'Graduate', '2021-08-15', '2023-12-15', 'Biology', 'Sacramento'),
( '10011', 'david.gray@university.com', '1122334455', '789 Cedar St, San Francisco',
'PhD', '2020-09-01', NULL, 'Mathematics', 'San Francisco'),
( '10016', 'sophia.lewis@university.com', '6677889900', '654 Pine Ave, Stockton',
'Undergrad', '2023-01-10', '2027-06-20', 'Business', 'Stockton'),
( '10018', 'william.hughes@university.com', '4433221100', '321 Oak Blvd,
Sacramento', 'Graduate', '2019-07-01', '2021-06-30', 'Chemistry', 'Sacramento');

-- #####
-- Populate Faculty Table
-- #####
INSERT INTO Faculty (UniversalId, Email, PhoneNumber, Address, Department,
Position, OfficeNumber, CampusLocation) VALUES
( '10002', 'jane.smith@university.com', '3216549870', '123 Birch Rd, Stockton',
'Mathematics', 'Professor', 'MATH101', 'Stockton'),
( '10007', 'chris.martin@university.com', '2345678901', '456 Maple Dr, Sacramento',
'Computer Science', 'Dean', 'CS301', 'Sacramento'),
( '10012', 'amy.evans@university.com', '3456789012', '789 Willow Ln, San Francisco',
'Biology', 'TA', 'BI0210', 'San Francisco'),
( '10017', 'mia.carter@university.com', '4567890123', '321 Aspen Ave, Stockton',
'Chemistry', 'Professor', 'CHEM102', 'Stockton'),
( '10020', 'ethan.lopez@university.com', '5678901234', '654 Palm Dr, Sacramento',
'Physics', 'HOD', 'PHY403', 'Sacramento');

-- #####
-- Populate Staff Table
-- #####
INSERT INTO Staff (UniversalId, Email, PhoneNumber, Address, JobTitle, Department,
```

```
CampusLocation) VALUES
('10003', 'mark.taylor@university.com', '7890123456', '789 Spruce Blvd, Stockton',
'Lab Technician', 'Biology', 'Stockton'),
('10008', 'samantha.green@university.com', '8901234567', '654 Redwood Ln,
Sacramento', 'Administrative Assistant', 'Library', 'Sacramento'),
('10013', 'rachel.scott@university.com', '9012345678', '321 Cypress St, San
Francisco', 'IT Specialist', 'Computer Services', 'San Francisco'),
('10018', 'william.hughes@university.com', '1234567891', '123 Juniper Rd,
Stockton', 'Research Assistant', 'Chemistry', 'Stockton'),
('10019', 'isabella.clark@university.com', '2345678902', '456 Sequoia Dr,
Sacramento', 'Facilities Manager', 'Operations', 'Sacramento');

-- #####
-- Populate Alumni Table
-- #####
INSERT INTO Alumni (UniversalId, Email, PhoneNumber, GraduationYear, Degree,
AlumniStatus, CampusLocation) VALUES
('10004', 'emily.johnson@university.com', '8901234567', 2020, 'MBA', 'Active',
'Stockton'),
('10009', 'robert.white@university.com', '9012345678', 2015, 'BSc in Physics',
'Inactive', 'San Francisco'),
('10014', 'oliver.harris@university.com', '1234567890', 2018, 'BA in History',
'Active', 'Sacramento'),
('10019', 'isabella.clark@university.com', '2345678901', 2012, 'MSc in Chemistry',
'Active', 'Stockton'),
('10015', 'nathan.king@university.com', '3456789012', 2019, 'BSc in Computer
Science', 'Inactive', 'Sacramento');

-- #####
-- Populate CommunityMembers Table
-- #####
INSERT INTO CommunityMembers (UniversalId, Email, PhoneNumber, MembershipStartDate,
MembershipEndDate, CampusLocation) VALUES
('10005', 'michael.brown@university.com', '5678901234', '2023-01-01', NULL,
'Stockton'),
('10010', 'jessica.black@university.com', '6789012345', '2022-06-15', '2023-12-31',
'Sacramento'),
('10015', 'nathan.king@university.com', '7890123456', '2021-03-10', NULL, 'San
Francisco'),
('10013', 'rachel.scott@university.com', '8901234567', '2020-08-20', '2022-07-01',
'Stockton'),
('10020', 'ethan.lopez@university.com', '9012345678', '2019-09-15', '2023-01-01',
'Sacramento');

----

-- #####
-- Populate Resources Table
-- #####
INSERT INTO Resources (Title, Author, ResourceType, PublicationDate, Genre,
```

```

OverdueFinePerDay, RequiredAccessLevel, AvailabilityStatus) VALUES
('Introduction to Data Science', 'Alice Roberts', 'Book', '2018-06-15',
'Education', 1.50, 2, 'Available'),
('Advanced Python Programming', 'Bob Smith', 'E-Book', '2020-03-20', 'Programming',
0.75, 3, 'Available'),
('Art History: Volume 1', 'Cathy Johnson', 'Journal', '2015-11-05', 'Art', 2.00, 1,
'Unavailable'),
('Interactive Multimedia Design', 'David Lee', 'Multimedia', '2021-01-10',
'Design', 0.50, 4, 'Available'),
('Machine Learning Essentials', 'Eve Brown', 'Book', '2019-09-12', 'Technology',
1.00, 3, 'Available'),
('Statistics for Scientists', 'Frank Miller', 'E-Book', '2017-08-01',
'Mathematics', 0.80, 2, 'Unavailable'),
('World History: 20th Century', 'George Lucas', 'Journal', '2014-05-22', 'History',
1.50, 0, 'Available'),
('3D Animation Principles', 'Hannah White', 'Multimedia', '2023-02-14', 'Art',
0.60, 5, 'Available'),
('Robotics for Engineers', 'Ian Gray', 'Book', '2022-11-18', 'Engineering', 1.20,
4, 'Unavailable'),
('Cybersecurity Basics', 'Jack Brown', 'E-Book', '2020-03-30', 'Technology', 1.00,
3, 'Available'),
('Artificial Intelligence Overview', 'Kelly Green', 'Book', '2021-05-10', 'Computer
Science', 1.25, 3, 'Available'),
('Healthcare Innovations', 'Liam Clark', 'Journal', '2020-12-15', 'Medicine', 0.75,
2, 'Available'),
('Environmental Studies', 'Mia Carter', 'Book', '2018-07-01', 'Environment', 1.30,
2, 'Available'),
('Digital Marketing Strategies', 'Nathan King', 'Multimedia', '2021-06-25',
'Business', 0.50, 0, 'Unavailable'),
('Ethics in Research', 'Olivia Taylor', 'Journal', '2016-01-05', 'Philosophy',
2.50, 3, 'Available'),
('Fundamentals of Chemistry', 'Paul Harris', 'Book', '2019-02-20', 'Science', 1.40,
2, 'Available'),
('Historical Biographies', 'Quinn Evans', 'Journal', '2013-09-15', 'History', 1.00,
1, 'Unavailable'),
('Renewable Energy Sources', 'Rachel Scott', 'E-Book', '2020-03-11', 'Technology',
0.60, 4, 'Available'),
('Project Management Basics', 'Sophia Lewis', 'Multimedia', '2022-07-08',
'Management', 0.75, 3, 'Available'),
('Cultural Studies: Asia', 'Tom Carter', 'Journal', '2017-12-05', 'Anthropology',
1.20, 2, 'Available');

-- #####
-- Populate Libraries Table
-- #####
INSERT INTO Libraries (LibraryName, BranchName, LocationDetails) VALUES
('Central Library', 'Stockton', 'Main Campus, Building A, Room 101'),
('Digital Resource Hub', 'Stockton', 'Main Campus, Building B, 2nd Floor'),
('Conference Room Alpha', 'Stockton', 'Main Campus, Building A, Room 201'),

```

```

('Engineering Lab', 'Sacramento', 'Engineering Building, Room 305'),
('Business Center', 'Sacramento', 'Business School, Room 102'),
('Medical Research Lab', 'Sacramento', 'Medical Campus, Room 120'),
('Art Library', 'San Francisco', 'Art Campus, Building D, Room 110'),
('Tech Library', 'San Francisco', 'Technology Building, Room 203'),
('History Archives', 'San Francisco', 'History Campus, Building C, Room 50'),
('Research Lab Delta', 'Stockton', 'Research Wing, Room 202');

-- #####
-- Populate ResourceLocations Table
-- #####
INSERT INTO ResourceLocations (ResourceId, LibraryId, CopiesAvailable, BorrowScope,
    MinPrivilegeToCrossCampus) VALUES
(1, 1, 5, 'AllCampus', 2),
(2, 1, 10, 'OwnCampus', 3),
(3, 3, 2, 'OwnCampus', 1),
(4, 4, 3, 'AllCampus', 4),
(5, 5, 7, 'AllCampus', 2),
(6, 6, 1, 'OwnCampus', 2),
(7, 7, 4, 'AllCampus', 1),
(8, 8, 8, 'AllCampus', 5),
(9, 9, 6, 'OwnCampus', 4),
(10, 10, 9, 'AllCampus', 3),
(11, 2, 2, 'OwnCampus', 3),
(12, 3, 3, 'OwnCampus', 2),
(13, 4, 1, 'AllCampus', 3),
(14, 5, 5, 'AllCampus', 4),
(15, 6, 10, 'OwnCampus', 2),
(16, 7, 7, 'AllCampus', 1),
(17, 8, 4, 'OwnCampus', 5),
(18, 9, 2, 'AllCampus', 3),
(19, 10, 3, 'OwnCampus', 2),
(20, 1, 6, 'AllCampus', 2);
-----

-- #####
-- Populate BorrowingRecords Table
-- #####
INSERT INTO BorrowingRecords (ResourceId, UniversalId, LibraryId, BorrowDate,
    DueDate, ReturnDate, Status) VALUES
(1, '10001', 1, '2023-11-01', '2023-11-15', NULL, 'Active'),
(2, '10002', 2, '2023-10-20', '2023-11-03', '2023-11-05', 'Returned'),
(3, '10003', 3, '2023-11-05', '2023-11-20', NULL, 'Overdue'),
(4, '10004', 4, '2023-11-10', '2023-11-25', '2023-11-24', 'Returned'),
(5, '10005', 5, '2023-11-15', '2023-12-01', NULL, 'Active'),
(6, '10006', 6, '2023-10-25', '2023-11-10', '2023-11-09', 'Returned'),
(7, '10007', 7, '2023-11-01', '2023-11-15', NULL, 'Overdue'),
(8, '10008', 8, '2023-11-05', '2023-11-20', '2023-11-18', 'Returned'),
(9, '10009', 9, '2023-11-15', '2023-12-01', NULL, 'Active'),

```

```
(10, '10010', 10, '2023-10-20', '2023-11-05', '2023-11-04', 'Returned');

-- #####
-- Populate OverdueRecords Table
-- #####
INSERT INTO OverdueRecords (BorrowingId, OverdueDays, FineAmount,
    DateOverdueCalculated) VALUES
(3, 5, 7.50, '2023-11-25'),
(7, 3, 4.50, '2023-11-18'),
(9, 2, 3.00, '2023-12-03');

-- #####
-- Populate RenewalRecords Table
-- #####
INSERT INTO RenewalRecords (RenewalCount, BorrowingId, RenewalDate, NewDueDate)
VALUES
(1, 1, '2023-11-15', '2023-11-30'),
(2, 5, '2023-12-01', '2023-12-15'),
(3, 9, '2023-12-01', '2023-12-15');

-- #####
-- Populate Reservations Table
-- #####
INSERT INTO Reservations (ResourceId, UniversalId, LibraryId, ReservationDate,
    PickupDate, Status) VALUES
(1, '10001', 1, '2023-11-25', '2023-11-28 10:00:00', 'Active'),
(2, '10002', 2, '2023-11-20', '2023-11-23 14:00:00', 'Cancelled'),
(3, '10003', 3, '2023-11-22', '2023-11-25 12:00:00', 'No response'),
(4, '10004', 4, '2023-11-18', '2023-11-21 09:00:00', 'Active'),
(5, '10005', 5, '2023-11-10', '2023-11-13 11:00:00', 'Cancelled'),
(6, '10006', 6, '2023-11-08', '2023-11-11 15:00:00', 'Active'),
(7, '10007', 7, '2023-11-06', '2023-11-09 13:00:00', 'No response'),
(8, '10008', 8, '2023-11-04', '2023-11-07 10:00:00', 'Active'),
(9, '10009', 9, '2023-11-02', '2023-11-05 16:00:00', 'Cancelled'),
(10, '10010', 10, '2023-10-28', '2023-10-31 14:00:00', 'Active');
-----

-- #####
-- Populate Events Table
-- #####
INSERT INTO Events (EventName, Description, EventDate, Starttime, Endtime,
    LocationId, MaxParticipants, EventStatus) VALUES
('AI Workshop', 'A workshop on artificial intelligence and machine learning
    trends.', '2024-01-15', '10:00:00', '12:00:00', 1, 50, 'Active'),
('Cybersecurity Seminar', 'Insights into modern cybersecurity practices.',
    '2024-02-10', '14:00:00', '16:00:00', 2, 100, 'Completed'),
('Art History Lecture', 'Discussion on Renaissance art and its influence.',
    '2024-03-05', '09:00:00', '11:00:00', 3, 30, 'Cancelled'),
('Data Science Bootcamp', 'Hands-on sessions for aspiring data scientists.',
```

```

    '2024-04-12', '13:00:00', '17:00:00', 4, 40, 'Active'),
('Business Management Conference', 'Conference on business strategy and
management.', '2024-05-20', '09:00:00', '15:00:00', 5, 200, 'Completed'),
('Medical Research Symposium', 'Presentations on cutting-edge medical research.',
'2024-06-18', '10:00:00', '14:00:00', 6, 80, 'Active'),
('Tech Talk', 'A talk on emerging technologies and their applications.',
'2024-07-25', '16:00:00', '18:00:00', 7, 60, 'Cancelled'),
('Cultural Studies Seminar', 'Exploring cultural studies in the modern world.',
'2024-08-14', '11:00:00', '13:00:00', 8, 20, 'Active'),
('Leadership Workshop', 'Training on leadership and organizational skills.',
'2024-09-22', '14:00:00', '18:00:00', 9, 150, 'Completed'),
('Environmental Awareness Drive', 'Initiative to promote awareness about
environmental conservation.', '2024-10-10', '08:00:00', '10:30:00', 10, 100,
'Active');

```

```

-- #####
-- Populate EventReservations Table
-- #####
INSERT INTO EventReservations (FullName, EventId, UniversalId) VALUES
('John Doe', 1, '10001'),
('Jane Smith', 1, '10002'),
('Mark Taylor', 2, '10003'),
('Emily Johnson', 2, '10004'),
('Michael Brown', 3, '10005'),
('Laura Wilson', 3, '10006'),
('Chris Martin', 4, '10007'),
('Samantha Green', 5, '10008'),
('Robert White', 6, '10009'),
('Jessica Black', 7, '10010'),
('David Gray', 8, '10011'),
('Amy Evans', 9, '10012'),
('Rachel Scott', 10, '10013'),
('Oliver Harris', 5, '10014'),
('Nathan King', 6, '10015'),
('Sophia Lewis', 4, '10016'),
('Mia Carter', 8, '10017'),
('William Hughes', 2, '10018'),
('Isabella Clark', 9, '10019'),
('Ethan Lopez', 10, '10020');

```

```

-----

-- #####
-- Populate Facilities Table
-- #####
INSERT INTO Facilities (FacilityName, LocationId, FacilityType, SlotDuration,
AvailabilityStatus, RequiredLevel, MaxOccupancy) VALUES
('Study Room Alpha', 1, 'Study Room', 60, 'Available', 1, 4),
('Computer Lab Delta', 2, 'Lab', 120, 'Reserved', 2, 20),
('Conference Hall Gamma', 3, 'Conference Hall', 180, 'UnderMaintenance', 3, 50),

```



```

('Study Room Beta', 4, 'Study Room', 30, 'Available', 2, 6),
('Research Lab Theta', 5, 'Lab', 90, 'Available', 3, 15),
('Meeting Room Sigma', 6, 'Conference Hall', 120, 'Reserved', 1, 25),
('Quiet Study Room', 7, 'Study Room', 45, 'Available', 1, 2),
('Media Lab Omega', 8, 'Lab', 150, 'UnderMaintenance', 4, 30),
('Boardroom Zeta', 9, 'Conference Hall', 240, 'Available', 5, 12),
('Solo Study Booth', 10, 'Study Room', 60, 'Available', 1, 1);

-- #####
-- Populate FacilityReservations Table
-- #####
INSERT INTO FacilityReservations (FacilityId, UniversalId, ReservationDate,
    StartTime, EndTime, Status) VALUES
(1, '10001', '2023-12-01', '10:00:00', '11:00:00', 'Active'),
(2, '10002', '2023-12-02', '09:00:00', '11:00:00', 'Cancelled'),
(3, '10003', '2023-12-03', '13:00:00', '15:00:00', 'Completed'),
(4, '10004', '2023-12-04', '14:00:00', '15:30:00', 'Active'),
(5, '10005', '2023-12-05', '12:00:00', '14:00:00', 'Cancelled'),
(6, '10006', '2023-12-06', '08:00:00', '09:30:00', 'Completed'),
(7, '10007', '2023-12-07', '11:00:00', '12:30:00', 'Active'),
(8, '10008', '2023-12-08', '10:00:00', '11:30:00', 'Cancelled'),
(9, '10009', '2023-12-09', '15:00:00', '17:00:00', 'Completed'),
(10, '10010', '2023-12-10', '16:00:00', '18:00:00', 'Active');
----

-- #####
-- Populate SupportServices Table
-- #####
INSERT INTO SupportServices (ServiceName, ServiceType, Description) VALUES
('Math Tutoring', 'Tutoring', 'Assistance with mathematics and statistics.'),
('Writing Lab', 'Writing Assistance', 'Help with academic writing and essays.'),
('Career Advising', 'Advising', 'Guidance on career options and job search
    strategies.'),
('Mental Health Consultation', 'Consultation', 'Support for mental health and well-
    being.'),
('Science Tutoring', 'Tutoring', 'Help with biology, chemistry, and physics
    topics.'),
('Resume Workshop', 'Writing Assistance', 'Help with resume writing and cover
    letters.'),
('Academic Advising', 'Advising', 'Support for course selection and academic
    planning.'),
('Financial Aid Consultation', 'Consultation', 'Assistance with financial aid and
    scholarships.'),
('Programming Help', 'Tutoring', 'Support for coding and software development.'),
('Public Speaking Workshop', 'Writing Assistance', 'Training for effective public
    speaking.');
```

-- #####

-- Populate AcademicSupportStaff Table

```
-- #####
INSERT INTO AcademicSupportStaff (StaffId, ServiceId, ProficiencyLevel) VALUES
('10003', 1, 'Expert'),
('10008', 2, 'Intermediate'),
('10013', 3, 'Beginner'),
('10018', 4, 'Expert'),
('10019', 5, 'Intermediate'),
('10014', 6, 'Beginner'),
('10015', 7, 'Intermediate'),
('10020', 8, 'Expert'),
('10012', 9, 'Intermediate'),
('10009', 10, 'Beginner');

-- #####
-- Populate Appointments Table
-- #####
INSERT INTO Appointments (ServiceId, UniversalId, StaffId, AppointmentDate,
    StartTime, EndTime, LocationId, Status) VALUES
(1, '10001', '10003', '2023-12-01', '10:00:00', '11:00:00', 1, 'Scheduled'),
(2, '10002', '10008', '2023-12-02', '09:30:00', '10:30:00', 2, 'Cancelled'),
(3, '10003', '10013', '2023-12-03', '13:00:00', '14:00:00', 3, 'Completed'),
(4, '10004', '10018', '2023-12-04', '14:00:00', '15:30:00', 4, 'Scheduled'),
(5, '10005', '10019', '2023-12-05', '12:00:00', '13:30:00', 5, 'Cancelled'),
(6, '10006', '10014', '2023-12-06', '08:00:00', '09:30:00', 6, 'Completed'),
(7, '10007', '10015', '2023-12-07', '11:00:00', '12:30:00', 7, 'Scheduled'),
(8, '10008', '10020', '2023-12-08', '10:00:00', '11:30:00', 8, 'Cancelled'),
(9, '10009', '10012', '2023-12-09', '15:00:00', '16:30:00', 9, 'Completed'),
(10, '10010', '10009', '2023-12-10', '16:00:00', '17:30:00', 10, 'Scheduled');

-----

-- #####
-- Populate ResearchTopics Table
-- #####
INSERT INTO ResearchTopics (TopicName, ResearchCategory, Description) VALUES
('Artificial Intelligence', 'STEM', 'Exploration of AI technologies and
    applications.'),
('Renaissance Art', 'Arts', 'Study of Renaissance artwork and its cultural
    significance.'),
('Economic Development', 'Social Sciences', 'Research on global economic trends and
    growth factors.'),
('Business Ethics', 'Business', 'Study of ethical practices in corporate
    environments.'),
('Climate Change', 'STEM', 'Research on global warming and its impact on
    ecosystems.'),
('Historical Biographies', 'Humanities', 'Study of influential historical
    figures.'),
('Social Media Trends', 'Social Sciences', 'Research on the evolution of social
    media platforms.');
```

```
( 'Renewable Energy', 'STEM', 'Investigation into sustainable energy solutions.' ),
( 'Creative Writing', 'Arts', 'Study of literary techniques and storytelling.' ),
( 'Entrepreneurship Strategies', 'Business', 'Research on effective business start-
up methods.' );
```

```
-- #####
```

```
-- Populate SupportStaff Table
```

```
-- #####
```

```
INSERT INTO SupportStaff (StaffId, TopicId, ProficiencyLevel) VALUES
```

```
( '10003', 1, 'Expert' ),
( '10008', 2, 'Intermediate' ),
( '10013', 3, 'Beginner' ),
( '10018', 4, 'Expert' ),
( '10019', 5, 'Intermediate' ),
( '10014', 6, 'Beginner' ),
( '10015', 7, 'Intermediate' ),
( '10020', 8, 'Expert' ),
( '10012', 9, 'Intermediate' ),
( '10009', 10, 'Beginner' );
```

```
-- #####
```

```
-- Populate Consultations Table
```

```
-- #####
```

```
INSERT INTO Consultations (UniversalId, StaffId, TopicId, ConsultationDate,
StartTime, EndTime, LocationId, Status, Feedback) VALUES
```

```
( '10001', '10003', 1, '2023-12-01', '10:00:00', '11:00:00', 1, 'Scheduled',
'Looking forward to discussing AI trends.' ),
( '10002', '10008', 2, '2023-12-02', '09:30:00', '10:30:00', 2, 'Cancelled', 'Unable
to attend the art consultation.' ),
( '10003', '10013', 3, '2023-12-03', '13:00:00', '14:00:00', 3, 'Completed',
'Insightful session on economic policies.' ),
( '10004', '10018', 4, '2023-12-04', '14:00:00', '15:30:00', 4, 'Scheduled',
'Excited for the discussion on business ethics.' ),
( '10005', '10019', 5, '2023-12-05', '12:00:00', '13:30:00', 5, 'Cancelled',
'Meeting postponed due to scheduling conflicts.' ),
( '10006', '10014', 6, '2023-12-06', '08:00:00', '09:30:00', 6, 'Completed', 'Great
insights into historical figures.' ),
( '10007', '10015', 7, '2023-12-07', '11:00:00', '12:30:00', 7, 'Scheduled',
'Excited to discuss social media strategies.' ),
( '10008', '10020', 8, '2023-12-08', '10:00:00', '11:30:00', 8, 'Cancelled',
'Consultation on renewable energy was rescheduled.' ),
( '10009', '10012', 9, '2023-12-09', '15:00:00', '16:30:00', 9, 'Completed',
'Interesting session on creative writing techniques.' ),
( '10010', '10009', 10, '2023-12-10', '16:00:00', '17:30:00', 10, 'Scheduled',
'Looking forward to discussing entrepreneurship.' );
```

```
-----*****
```

```
--Check ALL Data
```

```
-----*****
```

```
-- #####
-- Retrieve Data from Users and Related Tables
-- #####
SELECT * FROM Users;

SELECT * FROM Students;

SELECT * FROM Faculty;

SELECT * FROM Staff;

SELECT * FROM Alumni;

SELECT * FROM CommunityMembers;

-- #####
-- Retrieve Data from Resource Management Tables
-- #####
SELECT * FROM Resources;

SELECT * FROM Libraries;

SELECT * FROM ResourceLocations;

-- #####
-- Retrieve Data from Borrowing and Reservations Tables
-- #####
SELECT * FROM BorrowingRecords;

SELECT * FROM OverdueRecords;

SELECT * FROM RenewalRecords;

SELECT * FROM Reservations;

-- #####
-- Retrieve Data from Event Management Tables
-- #####
SELECT * FROM Events;

SELECT * FROM EventReservations;

-- #####
-- Retrieve Data from Facility Management Tables
-- #####
SELECT * FROM Facilities;

SELECT * FROM FacilityReservations;
```

```
-- #####
-- Retrieve Data from Support Services Tables
-- #####
SELECT * FROM SupportServices;

SELECT * FROM AcademicSupportStaff;

SELECT * FROM Appointments;

-- #####
-- Retrieve Data from Research Assistance Tables
-- #####
SELECT * FROM ResearchTopics;

SELECT * FROM SupportStaff;

SELECT * FROM Consultations;
```