VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DATABASE SYSTEM (CO2014)

Lab Assignment

Lab 5

Instructor(s): Nguyễn Minh Tâm

Students: Phạm Trần Minh Trí - 2313622 *(Group TN01)*
Lê Nguyễn Kim Khôi - 2311671 *(Group TN01)*

HO CHI MINH CITY, DECEMBER 2025

```
CREATE DATABASE COMPANY;
GO
USE COMPANY;
```

1 Exercise 1

First, we use the script from database lab 4 to create and populate the database.

```
USE master;
CREATE DATABASE COMPANY;
GO
USE COMPANY;


CREATE TABLE EMPLOYEE (
    FNAME           VARCHAR(15)     NOT NULL,
    MINIT           CHAR,
    LNAME           VARCHAR(15)     NOT NULL,
    SSN             CHAR(9),
    BDATE           DATE,
    ADDRESS         VARCHAR(50),
    SEX             CHAR,
    SALARY          DECIMAL(10,2),
    SUPER_SSN       CHAR(9),
    DNO             INT,
    PRIMARY KEY (SSN),
    FOREIGN KEY (SUPER_SSN) REFERENCES EMPLOYEE (SSN)
);

CREATE TABLE DEPARTMENT (
    DNAME           VARCHAR(15)     NOT NULL UNIQUE,
    DNUMBER         INT,
    MGR_SSN         CHAR(9)         NOT NULL,
    MGR_START_DATE  DATE,
    PRIMARY KEY (DNUMBER)
);

CREATE TABLE DEPT_LOCATIONS (
    DNUMBER         INT,
    DLOCATION       VARCHAR(15),
    PRIMARY KEY (DNUMBER, DLOCATION),
    FOREIGN KEY (DNUMBER) REFERENCES DEPARTMENT (DNUMBER)
);

CREATE TABLE PROJECT (
    PNAME           VARCHAR(15)     NOT NULL,
    PNUMBER         INT,
    PLOCATION       VARCHAR(15)     NOT NULL,
    DNUM            INT,
    PRIMARY KEY (PNUMBER),
    FOREIGN KEY (DNUM) REFERENCES DEPARTMENT (DNUMBER)
);

CREATE TABLE WORKS_ON (
    ESSN            CHAR(9),
    PNO             INT,
    HOURS           DECIMAL(3,1),
    PRIMARY KEY (ESSN, PNO),
    FOREIGN KEY (ESSN) REFERENCES EMPLOYEE (SSN),
    FOREIGN KEY (PNO) REFERENCES PROJECT (PNUMBER)
```

```sql
);

CREATE TABLE DEPENDENT (
    ESSN            CHAR(9),
    DEPENDENT_NAME  VARCHAR(15),
    SEX             CHAR,
    BDATE           DATE,
    RELATIONSHIP    VARCHAR(15),
    PRIMARY KEY (ESSN, DEPENDENT_NAME),
    FOREIGN KEY (ESSN) REFERENCES EMPLOYEE (SSN)
);

ALTER TABLE EMPLOYEE
ADD FOREIGN KEY (DNO) REFERENCES DEPARTMENT (DNUMBER);

ALTER TABLE DEPARTMENT
ADD FOREIGN KEY (MGR_SSN) REFERENCES EMPLOYEE (SSN);
```

Listing 1: Create database

```sql
USE COMPANY;


INSERT INTO EMPLOYEE VALUES
    ('John', 'B', 'Smith', '123456789', '1965-01-09', '731 Fondren, Houston, TX',
    'M', 30000, NULL, NULL),
    ('Franklin', 'T', 'Wong', '333445555', '1955-12-08', '638 Voss, Houston, TX',
    'M', 40000, NULL, NULL),
    ('Alicia', 'J', 'Zelaya', '999887777', '1968-07-19', '3321 Castle, Spring, TX'
    , 'F', 25000, NULL, NULL),
    ('Jennifer', 'S', 'Wallace', '987654321', '1941-06-20', '291 Berry, Bellaire,
    TX', 'F', 43000, NULL, NULL),
    ('Ramesh', 'K', 'Narayan', '666884444', '1962-09-15', '975 Fire Oak, Humble,
    TX', 'M', 38000, NULL, NULL),
    ('Joyce', 'A', 'English', '453453453', '1972-07-31', '5631 Rice, Houston, TX',
     'F', 25000, NULL, NULL),
    ('Ahmad', 'V', 'Jabbar', '987987987', '1969-03-29', '980 Dallas, Houston, TX',
     'M', 25000, NULL, NULL),
    ('James', 'E', 'Borg', '888665555', '1937-11-10', '450 Stone, Houston, TX', 'M
    ', 55000, NULL, NULL);

INSERT INTO DEPARTMENT VALUES
    ('Research', 5, 333445555, '1988-05-22'),
    ('Administration', 4, 987654321, '1995-01-01'),
    ('Headquarters', 1, 888665555, '1981-06-19');


UPDATE EMPLOYEE SET SUPER_SSN = '333445555', DNO = 5 WHERE SSN = '123456789';
UPDATE EMPLOYEE SET SUPER_SSN = '888665555', DNO = 5 WHERE SSN = '333445555';
UPDATE EMPLOYEE SET SUPER_SSN = '987654321', DNO = 4 WHERE SSN = '999887777';
UPDATE EMPLOYEE SET SUPER_SSN = '888665555', DNO = 4 WHERE SSN = '987654321';
UPDATE EMPLOYEE SET SUPER_SSN = '333445555', DNO = 5 WHERE SSN = '666884444';
UPDATE EMPLOYEE SET SUPER_SSN = '333445555', DNO = 5 WHERE SSN = '453453453';
UPDATE EMPLOYEE SET SUPER_SSN = '987654321', DNO = 4 WHERE SSN = '987987987';
UPDATE EMPLOYEE SET SUPER_SSN = NULL, DNO = 1 WHERE SSN = '888665555';

INSERT INTO DEPT_LOCATIONS VALUES
    (1, 'Houston'),
    (4, 'Stafford'),
    (5, 'Bellaire'),
    (5, 'Sugarland'),
    (5, 'Houston');
```

```
INSERT INTO PROJECT VALUES
    (’ProductX’, 1, ’Bellaire’, 5),
    (’ProductY’, 2, ’Sugarland’, 5),
    (’ProductZ’, 3, ’Houston’, 5),
    (’Computerization’, 10, ’Stafford’, 4),
    (’Reorganization’, 20, ’Houston’, 1),
    (’Newbenefits’, 30, ’Stafford’, 4);

INSERT INTO WORKS_ON VALUES
    (’123456789’, 1, 32.5),
    (’123456789’, 2, 7.5),
    (’666884444’, 3, 40.0),
    (’453453453’, 1, 20.0),
    (’453453453’, 2, 20.0),
    (’333445555’, 2, 10.0),
    (’333445555’, 3, 10.0),
    (’333445555’, 10, 10.0),
    (’333445555’, 20, 10.0),
    (’999887777’, 30, 30.0),
    (’999887777’, 10, 10.0),
    (’987987987’, 10, 35.0),
    (’987987987’, 30, 5.0),
    (’987654321’, 30, 20.0),
    (’987654321’, 20, 15.0),
    (’888665555’, 20, NULL);

INSERT INTO DEPENDENT VALUES
    (’333445555’, ’Alice’, ’F’, ’1986-04-05’, ’DAUGHTER’),
    (’333445555’, ’Theodore’, ’M’, ’1983-10-25’, ’SON’),
    (’333445555’, ’Joy’, ’F’, ’1958-05-03’, ’SPOUSE’),
    (’987654321’, ’Abner’, ’M’, ’1942-02-28’, ’SPOUSE’),
    (’123456789’, ’Michael’, ’M’, ’1988-01-04’, ’SON’),
    (’123456789’, ’Alice’, ’F’, ’1988-12-30’, ’DAUGHTER’),
    (’123456789’, ’Elizabeth’, ’F’, ’1967-05-05’, ’SPOUSE’);
```

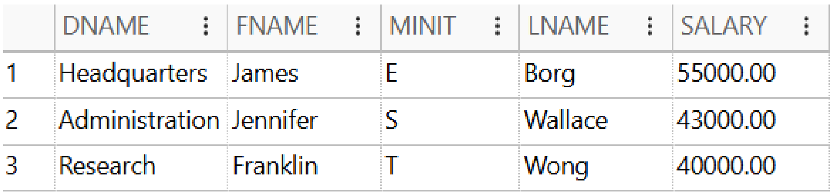
Listing 2: Populate database

1.1 VIEW

```
USE COMPANY;
GO

-- a. A view that has the department name, manager name, and manager salary for
    every department.
CREATE VIEW VIEW_A
AS
    SELECT DNAME, FNAME, MINIT, LNAME, SALARY
    FROM DEPARTMENT JOIN EMPLOYEE ON MGR_SSN = SSN;
GO

-- b. A view that has the employee name, supervisor name, and employee salary for
    each employee who works in the ’Research’ department.
CREATE VIEW VIEW_B
AS
    SELECT
        E1.FNAME AS EMPLOYEE_FNAME,
        E1.MINIT AS EMPLOYEE_MINIT,
        E1.LNAME AS EMPLOYEE_LNAME,
        E2.FNAME AS SUPER_FNAME,
        E2.MINIT AS SUPER_MINIT,
        E2.LNAME AS SUPER_LNAME,
        E1.SALARY
    FROM (EMPLOYEE E1 JOIN EMPLOYEE E2 ON E1.SUPER_SSN = E2.SSN) JOIN DEPARTMENT
    D1 ON E1.DNO = D1.DNUMBER
    WHERE D1.DNAME = ’Research’;
GO

-- c. A view that has the project name, controlling department name, number of
    employees, and total hours worked per week on the project for each project.
CREATE VIEW VIEW_C
AS
    SELECT
        PNAME,
        DNAME, COUNT(ESSN) AS NUM_EMPLOYEE,
        SUM([HOURS]) AS TOTAL_HOUR
    FROM
        (PROJECT JOIN DEPARTMENT ON DNUM = DNUMBER) JOIN WORKS_ON ON PNUMBER = PNO
    GROUP BY PNAME, DNAME;
GO

-- d. A view that has the project name, controlling department name, number of
    employees, and total hours worked per week on the project for each project
    with more than two employees working on it.
CREATE VIEW VIEW_D
AS
    SELECT
        PNAME,
        DNAME, COUNT(ESSN) AS NUM_EMPLOYEE,
        SUM([HOURS]) AS TOTAL_HOUR
    FROM
        (PROJECT JOIN DEPARTMENT ON DNUM = DNUMBER) JOIN WORKS_ON ON PNUMBER = PNO
    GROUP BY PNAME, DNAME
    HAVING COUNT(ESSN) > 2;
GO

-- e. A view (SSN, Full Name of employee, Number of dependents) that includes
    information about employees who have the number of dependents greater than 2.
CREATE VIEW VIEW_E
AS
    SELECT
        SSN,
        CONCAT(FNAME, ’ ’, MINIT, ’ ’, LNAME) AS FULL_NAME,
        COUNT(DEPENDENT_NAME) AS NUM_DEPENDENT
    FROM EMPLOYEE JOIN DEPENDENT ON SSN = ESSN
    GROUP BY SSN, FNAME, MINIT, LNAME
    HAVING COUNT(DEPENDENT_NAME) > 2;
GO

-- f. A view (Full Name of employee, date of birth, gender) for those employees
    who have their birthdate in July.
CREATE VIEW VIEW_F
AS
    SELECT
        CONCAT(FNAME, ’ ’, MINIT, ’ ’, LNAME) AS FULL_NAME,
        BDATE,
        SEX
    FROM EMPLOYEE
    WHERE MONTH(BDATE) = 7;
GO

-- g. A view (Name of dependent, SSN of employee, date of birth of dependent) that
```

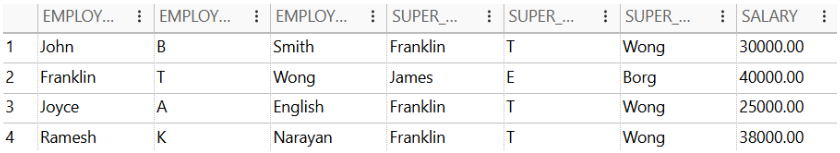
```
     includes information on all dependents who are less than 18 years old.
CREATE VIEW VIEW_G
AS
    SELECT DEPENDENT_NAME, ESSN, BDATE
    FROM DEPENDENT
    WHERE BDATE > DATEADD(YEAR, -18, GETDATE());
GO
```

Output results: See figure 1, 2, 3, 4, 5, 6, 7.



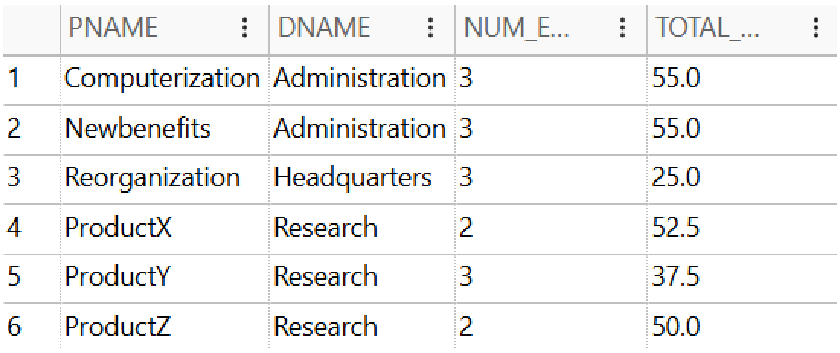
	DNAME	FNAME	MINIT	LNAME	SALARY
1	Headquarters	James	E	Borg	55000.00
2	Administration	Jennifer	S	Wallace	43000.00
3	Research	Franklin	T	Wong	40000.00

Figure 1: Result of VIEW_A



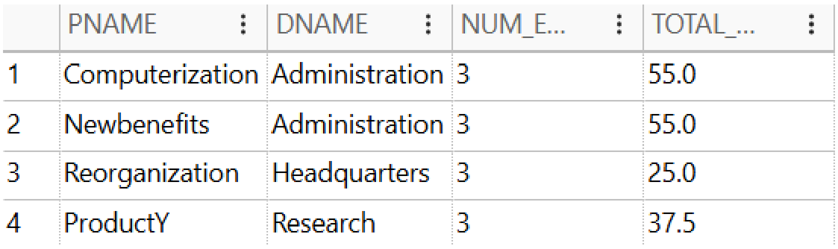
	EMPLOY...	EMPLOY...	EMPLOY...	SUPER_...	SUPER_...	SUPER_...	SALARY
1	John	B	Smith	Franklin	T	Wong	30000.00
2	Franklin	T	Wong	James	E	Borg	40000.00
3	Joyce	A	English	Franklin	T	Wong	25000.00
4	Ramesh	K	Narayan	Franklin	T	Wong	38000.00

Figure 2: Result of VIEW_B



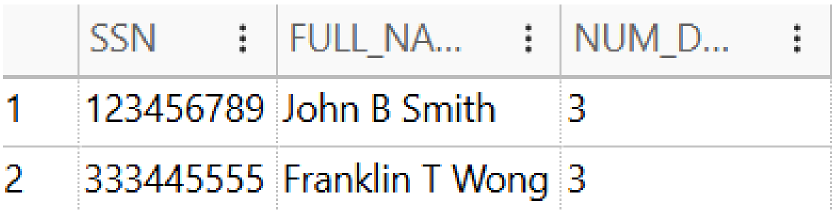
	PNAME	DNAME	NUM_E...	TOTAL_...
1	Computerization	Administration	3	55.0
2	Newbenefits	Administration	3	55.0
3	Reorganization	Headquarters	3	25.0
4	ProductX	Research	2	52.5
5	ProductY	Research	3	37.5
6	ProductZ	Research	2	50.0

Figure 3: Result of VIEW_C



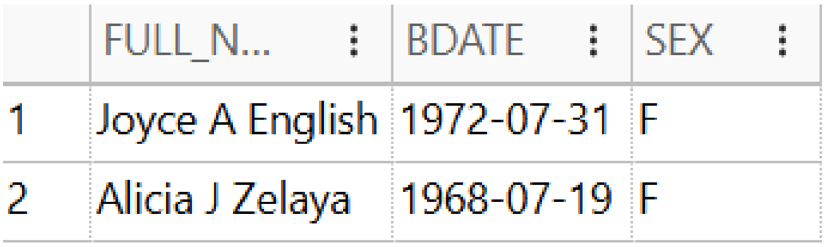
	PNAME	DNAME	NUM_E...	TOTAL_...
1	Computerization	Administration	3	55.0
2	Newbenefits	Administration	3	55.0
3	Reorganization	Headquarters	3	25.0
4	ProductY	Research	3	37.5

Figure 4: Result of VIEW_D



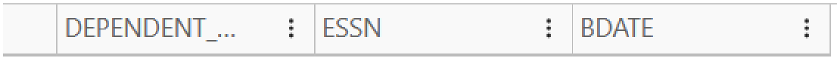
	SSN	FULL_NA...	NUM_D...
1	123456789	John B Smith	3
2	333445555	Franklin T Wong	3

Figure 5: Result of VIEW_E



	FULL_N...	BDATE	SEX
1	Joyce A English	1972-07-31	F
2	Alicia J Zelaya	1968-07-19	F

Figure 6: Result of VIEW_F



	DEPENDENT_...	ESSN	BDATE

Figure 7: Result of VIEW_G

2 Exercise 2

```
USE master;
GO

CREATE DATABASE HotelManagementDB;
GO


USE HotelManagementDB;
GO

CREATE TABLE Hotel(
    hotelNo VARCHAR(10) PRIMARY KEY,
    hotelName NVARCHAR(100),
    city NVARCHAR(50),
);
```

```
GO


CREATE TABLE Room(
    roomNo VARCHAR(10),
    hotelNo VARCHAR(10),
    type VARCHAR(20),
    price DECIMAL(18,2),
    NumAdultMax INT,
    PRIMARY KEY (roomNo, hotelNo),
    FOREIGN KEY(hotelNo) REFERENCES Hotel(hotelNo)
);
GO

CREATE TABLE Guest(
    guestNo VARCHAR(10) PRIMARY KEY,
    guestName NVARCHAR(100),
    guestAddress NVARCHAR(200),
    TotalAmount DECIMAL(18,2) DEFAULT 0
);
GO

CREATE TABLE Booking(
    hotelNo VARCHAR(10),
    dateFrom DATE,
    roomNo VARCHAR(10),
    guestNo VARCHAR(10),
    dateTo DATE,
    NumOfAdult INT,
    PRIMARY KEY (hotelNo, dateFrom, roomNo),
    FOREIGN KEY (roomNo, hotelNo) REFERENCES Room(roomNo, hotelNo),
    FOREIGN KEY (guestNo) REFERENCES Guest(guestNo)
);
GO
```

Listing 3: Create database

2.1 2a

```
ALTER TABLE Room
ADD CONSTRAINT CHK_DoubleRoomPrice
CHECK ((type = ’Double’ AND price > 100) OR type <> ’Double’)
GO
```

2.2 2b

```
CREATE TRIGGER trg_CheckDoublePriceVsSingle
on Room
after INSERT, UPDATE
AS
BEGIN
    IF exists (
        SELECT 1
        FROM inserted i
        JOIN Room r ON i.hotelNo = r.hotelNo
        WHERE i.type = ’Double’
        AND r.type = ’Single’
        AND i.price <= r.price
    )
    BEGIN
        RAISERROR("Error: Double Price greater than Single Price", 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO
```

2.3 2c

```
CREATE TRIGGER trg_CheckGuessOverlap
ON Booking
after UPDATE, INSERT
AS
BEGIN
    IF EXISTS(
        SELECT 1
        FROM inserted i
        JOIN Booking b ON i.guestNo = b.guestNo
        WHERE
            NOT (i.hotelNo = b.hotelNo AND i.roomNo = b.roomNo AND i.dateFrom = b.
    dateFrom)
            AND (i.dateFrom < b.dateTo AND i.dateTo > b.dateFrom)
    )
    BEGIN
        RAISERROR('Error: Overlap time customers', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
Go
```

2.4 2d

```
CREATE TRIGGER trg_CheckMaxAdults
ON Booking
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS(
        SELECT 1
        from inserted i
        JOIN Room r on i.hotelNo = r.hotelNo and i.roomNo = r.roomNo
        WHERE i.NumOfAdult > r.NumAdultMax
    )
    BEGIN
        RAISERROR('Error: Num of Adult greater than max capacity', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
GO
```

2.5 2e

```
CREATE TRIGGER trg_UpdateGuestTotalAmount
ON Booking
after INSERT, UPDATE, DELETE
```

```
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @AffectedGuests Table (guestNo VARCHAR(10));

    INSERT into @AffectedGuests
    SELECT guestNo from inserted
    UNION
    SELECT guestNo from deleted;

    UPDATE g
    SET TotalAmount = ISNULL((
        SElect sum(r.price * DATEDIFF(DAY, b.dateFrom, b.dateTo))
        From Booking b
        Join Room r on b.hotelNo = r.hotelNo and b.roomNo = r.roomNo
        where b.guestNo = g.guestNo
    ), 0)
    From Guest g
    WHERE g.guestNo IN (SELECT guestNo from @AffectedGuests);
END;
GO
```

2.6 2f

```
CREATE TRIGGER trg_InsertLondonHotelRoom
ON LondonHotelRoom
INSTEAD OF INSERT
AS
BEGIN
    set NOCOUNT ON;

    insert into Hotel(hotelNo, hotelName, city)
    SELECT distinct i.hotelNo, i.hotelName, i.city
    From inserted i
    WHERE not exists (select 1 from Hotel h where h.hotelNo = i.hotelNo);


    INSERT into Room(roomNo, hotelNo, type, price, NumAdultMax)
    SELECT i.roomNo, i.hotelNo, i.type, i.price, NULL
    FROM inserted i;
END;
GO
```