2022-2 데이터베이스시스템 Lab02

학번	2020136129	이름	최수연

Task -1

```
USE DreamHomeDB
go
DROP TABLE IF EXISTS Viewing
go

DROP TABLE IF EXISTS Registration
go

DROP TABLE IF EXISTS PropertyForRent
go

DROP TABLE IF EXISTS Staff
go

DROP TABLE IF EXISTS Privateowner
go

DROP TABLE IF EXISTS Clients
go

DROP TABLE IF EXISTS Branch
go

CREATE TABLE Branch(
branchNo VARCHAR(4) NOT NULL,
street VARCHAR(30) NOT NULL,
city VARCHAR(25) NOT NULL,
postcode VARCHAR(10) NOT NULL,
constraint PK_Branch PRIMARY KEY (branchNo),
constraint CH_bno CHECK (branchNo LIKE 'B%')
);
go
```

```
CREATE TABLE   Clients(
clientNo VARCHAR(6) NOT NULL,
fName VARCHAR(20) NULL,
lName VARCHAR(20) NULL,
telNo VARCHAR(22) NULL,
prefType CHAR(6),
MaxRent INT,
eMail VARCHAR(30) NULL,
constraint   PK_cno PRIMARY KEY (clientNo),
constraint CH_mr CHECK (maxRent>=350)
);
go

CREATE TABLE Privateowner(
ownerNo VARCHAR(6) NOT NULL,
fName VARCHAR(20) NOT NULL,
lName VARCHAR(20) NOT NULL,
addr VARCHAR(40) DEFAULT NULL,
telNo VARCHAR(22) NULL,
eMail VARCHAR(30) NULL,
constraint PK_ono PRIMARY KEY (ownerNo)
);
go

CREATE TABLE Staff(
staffNo VARCHAR(4) NOT NULL,
fName VARCHAR(20) NOT NULL,
lName VARCHAR(20) NOT NULL,
position VARCHAR(10) NULL,
gender CHAR(1) NOT NULL,
DOB DATE NOT NULL,
salary INT DEFAULT 9000,
branchNo VARCHAR(4),
constraint CH_salary CHECK (salary>=9000),
constraint PK_sno PRIMARY KEY (staffNo),
constraint FK_bno foreign key (branchNo) REFERENCES Branch(branchNo),
constraint CH_gender CHECK (gender IN('F', 'M'))
);
```

```
CREATE TABLE PropertyForRent(
propertyNo VARCHAR(10) NOT NULL,
street VARCHAR(30) NOT NULL,
city VARCHAR(25) NOT NULL,
postcode VARCHAR(10) NOT NULL,
room_type VARCHAR(10) NOT NULL,
rooms INT NOT NULL DEFAULT 4,
rent INT NOT NULL DEFAULT 600,
ownerNo VARCHAR(4) NOT NULL,
staffNo VARCHAR(4) NULL,
branchNo VARCHAR(4),
constraint PK_pno PRIMARY KEY (propertyNo),
constraint FK_sno foreign key (staffNo) REFERENCES Staff (staffNo),
constraint FK_bno2 foreign key (branchNo) REFERENCES Branch (branchNo),
constraint ch_rent CHECK (rent>=300)
);
go

CREATE TABLE Registration(
clientNo VARCHAR(6) NOT NULL,
branchNo VARCHAR(4) NOT NULL,
staffNo VARCHAR(4) NOT NULL,
dateJoined VARCHAR(20) NOT NULL,
constraint PK_cbsdNo PRIMARY KEY(clientNo, branchNo, staffNo, dateJoined),
constraint FK_cnoR foreign key (clientNo) REFERENCES Clients (clientNo),
constraint FK_bnoR foreign key (branchNo) REFERENCES Branch (branchNo),
constraint FK_snoR foreign key (staffNo) REFERENCES Staff (staffNo)
);
go

CREATE TABLE Viewing(
clientNo VARCHAR(6) NOT NULL,
propertyNo VARCHAR(10) NOT NULL,
viewDate VARCHAR(12) NOT NULL,
comments VARCHAR(20) NULL,
constraint PK_cpv PRIMARY KEY (clientNo,propertyNo, viewDate),
constraint FK_pnoV   FOREIGN KEY (propertyNo) REFERENCES PropertyForRent (propertyNo),
constraint FK_cnoV FOREIGN KEY (clientNo) REFERENCES Clients (clientNo)
```

```
);
go
```

```
use DreamHomeDB
go

INSERT INTO Branch VALUES ('B005', '22 Deer Rd', 'London', 'SW1 4EH');
INSERT INTO Branch VALUES ('B007', '16 Argyll St', 'Aberdeen', 'AB2 3SU');
INSERT INTO Branch VALUES ('B003', '163 Main St', 'Glasgow', 'G11 9QX');
INSERT INTO Branch VALUES ('B004', '32 Manse Rd', 'Bristol', 'BS99 1NZ');
INSERT INTO Branch VALUES ('B002', '56 Clover Dr', 'London', 'NW10 6EU');
INSERT INTO Branch VALUES ('B102', '561 Clover Dr', 'London', 'NW110 6EU');

select * from Branch

INSERT INTO Staff VALUES ('SL21', 'John', 'White','Manager', 'M', '1945-10-01', 30000, 'B005');
INSERT INTO Staff VALUES ('SG37', 'Ann', 'Beech','Assistant', 'F', '1960-10-11', 12000, 'B003');
INSERT INTO Staff VALUES ('SG14', 'David','Ford', 'Supervisor','M', '1958-11-24', 18000, 'B003');
INSERT INTO Staff VALUES ('SA9', 'Mary', 'Howe', 'Assistant', 'F', '1970-02-19', 9000, 'B007');
INSERT INTO Staff VALUES ('SG5', 'Susan','Brand','Manager', 'F', '1940-06-03', 24000, 'B003');
INSERT INTO Staff VALUES ('SL41', 'Julie','Lee', 'Assistant', 'F', '1965-06-13', 9000, 'B005');
INSERT INTO Staff VALUES ('SL45', 'Julie','Lee', 'Assistant', 'F', '1965-06-13', 15000, 'B005');
select * from Staff

INSERT INTO PrivateOwner VALUES ('CO46', 'Joe', 'Keogh', '2 Fergus Dr, Aberdeen AB2
7SX','01224-861212', 'abc@gmail.com');
INSERT INTO PrivateOwner VALUES ('CO87', 'Carol','Farrel','6 Achray St, Glasgow G32 9DX',
'0141-357-7419', 'abc@gmail.com');
INSERT INTO PrivateOwner VALUES ('CO40', 'Tina', 'Murphy','63 Well St, Glasgow G42', '0141-
943-1728', 'abc@gmail.com');
INSERT INTO PrivateOwner VALUES ('CO93', 'Tony', 'Shaw', '12 Park Pl, Glasgow G4 0QR', '0141-
225-7025', 'abc@gmail.com');
INSERT INTO PrivateOwner VALUES ('CO99', 'Rin', 'Puth', '17 Park Pl, Glasgow G51', '0141-123-
4326', 'rin@gmail.com');

select * from Privateowner

INSERT INTO PropertyForRent VALUES ('PA14', '16 Holhead', 'Aberdeen', 'AB7 5SU', 'House',6,
650, 'CO46', 'SA9', 'B007');
INSERT INTO PropertyForRent VALUES ('PL94', '6 Argyll St', 'London', 'NW2', 'Flat', 4, 400,
```

```
'CO87', 'SL41', 'B005');
INSERT INTO PropertyForRent VALUES ('PG4', '6 Lawrence St','Glasgow', 'G11 9QX', 'Flat', 3, 350,
'CO40', NULL, 'B003');
INSERT INTO PropertyForRent VALUES ('PG36', '2 Manor Rd', 'Glasgow', 'G32 4QX', 'Flat', 3, 375,
'CO93', 'SG37', 'B003');
INSERT INTO PropertyForRent VALUES ('PG21', '18 Dale Rd', 'Glasgow', 'G12', 'House',5, 600,
'CO87', 'SG37', 'B003');
INSERT INTO PropertyForRent VALUES ('PG16', '5 Novar Dr', 'Glasgow', 'G12 9AX', 'Flat', 4, 450,
'CO93', 'SG14', 'B003');

select * from PropertyForRent

INSERT INTO Clients VALUES ('CR76', 'John', 'Kay', '0207-774-5632', 'Flat', 425,
'abc@gmail.com');
INSERT INTO Clients VALUES ('CR56', 'Aline','Stewart', '0141-848-1825', 'Flat', 350,
'abc@gmail.com');
INSERT INTO Clients VALUES ('CR74', 'Mike', 'Ritchie', '01475-392178', 'House', 750,
'abc@gmail.com');
INSERT INTO Clients VALUES ('CR62', 'Mary', 'Tregar', '01224-196720', 'Flat', 600,
'abc@gmail.com');
INSERT INTO Clients VALUES ('CR51', 'Mike', 'Trout', '01272-172245', 'House', 900,
'mt@gmail.com');

select * from Clients

INSERT INTO Viewing VALUES ('CR56', 'PA14', '2001-05-24', 'too small');
INSERT INTO Viewing VALUES ('CR76', 'PG4', '2001-04-20', 'too --ote');
INSERT INTO Viewing VALUES ('CR56', 'PG4', '2001-05-26', NULL);
INSERT INTO Viewing VALUES ('CR62', 'PA14', '2001-05-14', 'no dining room');
INSERT INTO Viewing VALUES ('CR56', 'PG36', '2001-04-28', NULL);

select * from Viewing

INSERT INTO Registration VALUES ('CR76', 'B005', 'SL41', '2001-01-02');
INSERT INTO Registration VALUES ('CR56', 'B003', 'SG37', '2000-04-11');
INSERT INTO Registration VALUES ('CR74', 'B003', 'SG37', '1999-11-16');
INSERT INTO Registration VALUES ('CR62', 'B007', 'SA9', '2000-03-07');
INSERT INTO Registration VALUES ('CR51', 'B007', 'SA9', '2001-09-16');
select * from Registration
```

Task -2

1. List all book titles.

RA: $\Pi_{title}(Book)$

select title from Book

2. List all borrower details.

RA: Borrower

select * from Borrower

3. List all books titles published between 2010 and 2014.

RA: $\Pi_{title}(\sigma_{year>='2010' \wedge year<='2014'}(Book))$

select title from Book

where year BETWEEN 2010 AND 2014

4. Remove all books published before 1950 from the database.

RA: $\sigma_{year<'1950'}(Book)$

delete from Book

where year < 1950

5. List all book titles that have never been borrowed by any borrower.

RA: $\Pi_{title}\left(\sigma_{count(copyNo)=0}\left(BookLoan \bowtie_{BookLoan.copyNo = copyNo}\left(Book \bowtie_{Book.ISBN = BookCopy.ISBN} BookCopy\right)\right)\right)$

select b.title

from Book b, BookCopy bc, BookLoan bl

where b.ISBN = bc.ISBN

and select ((count(*) FROM BookLoan WHERE copyNo = bc.copyNo) = 0)

6. List all book titles that contain the word 'database' and are available for loan.

RA: $\Pi_{title}\left(\sigma_{title\ LIKE\ '\%database\%'\ and\ available = '1'}\left(Book \bowtie_{Book.ISBN = BookCopy.ISBN} BookCopy\right)\right)$

select b.title from Book b, BookCopy c

where b.ISBN = c.ISBN

and b.title like '%database%' and c.available = '1'

7. List the names of borrowers with overdue books.

RA: $\Pi_{borrowerName}(Borrower \bowtie_{Borrower.borrowerNo = borrowerNo} (\sigma_{dateDue>getdate}(BookLoan))$

SELECT borrowerName

From Borrower bw, BookLoan bl

Where bw.borrowerNo = b1.borrowerNo and dateDue > getdate

8. How many copies of each book title are there?

RA: $\Pi_{ISBN,\ COUNT(ISBN)\ \rightarrow\ cnt}\ \ \gamma_{ISBN,\ COUNT(ISBN)}\ BookCopy$

```
select ISBN, count(ISBN) as cnt from BookCopy
group by ISBN
```

9. How many copies of ISBN "0-321-52306-7" are currently available?

RA: $\Pi_{COUNT\ copyNO}\ (\ \sigma_{ISBN\ =\ '0-321-52306-7'\ and\ available\ =\ '1'}(BookCopy))$

```
select count(copyNo) from BookCopy
where ISBN = '0-321-52306-7'
and available = '1'
```

10. How many times has the book title with ISBN "0-321-52306-7" been borrowed?

RA: $\Pi_{COUNT\ copyNo}\ (BookLoan \bowtie_{BookLoan.copyNo\ =\ BookCopy.copyNo\ and\ BookCopy.ISBN\ =\ '0-321-52306-7'}\ BookCopy)$

```
select count(l.copyNo) from BookLoan l, BookCopy c
where l.copyNo = c.copyNo
and c.ISBN = '0-321-52306-7'
```

11. Produce a report of book titles that have been borrowed by "Peter Bloomfield."

RA: $\Pi_{title}\left(\sigma_{r.BorrowNo\ =\ l.BorrowNo\ and\ l.copyNo\ =\ c.copyNo\ and\ c.ISBN\ =\ b.ISBN\ and\ r.borrowerName\ =\ 'Peter\ Bloomfield'}(\rho_b Book \times \rho_r Borrower \times \rho_c BookCopy \times \rho_l BookLoan)\right)$

```
select b.title from Book b, Borrower r, BookCopy c, BookLoan l
where r.borrowerNo = l.borrowerNo
and l.copyNo = c.copyNo
and c.ISBN = b.ISBN
and r.borrowerName = 'Peter Bloomfield'
```

12. For each book title with more than three copies, list the names of library members who have borrowed them.

RA: $(\Pi_{b.title,r.borrowerName}\sigma_{r.BorrowNo\ =\ l.BorrowNo\ and\ l.copyNo\ =\ c.copyNo\ and\ c.ISBN\ =\ b.ISBN}\ (\rho_b Book \times \rho_l BookCopy \times \rho_c BookCopy \times \rho_l BookLoan)) \cap (\Pi_{ISBN,COUNT(c1.copyNo)}\sigma_{COUNT(c1.copyNo)\ >=\ 3}\gamma_{ISBN,COUNT(copyNo)}\sigma_{c.ISBN\ =\ c1.ISBN}(\rho_{c1} BookCopy))$

```
select b.title, r.borrowerName
from Borrower r, Book b, BookCopy c, BookLoan l
where r.borrowerNo = l.borrowerNo
and l.copyNo = c.CopyNo
and c.ISBN = b.ISBN
and exists (select ISBN, count(c1.CopyNo) from BookCopy c1
where c1.ISBN = c.ISBN
group by c1.ISBN
having count (c1.CopyNo) >= 3)
```

13. Produce a report with the details of borrowers who currently have books overdue.

RA:

$\Pi_{r.borrowerNo,r.borrwerName,r.borrowerAddress}\ \sigma_{r.borrowerNo = l.borrowerNo\ and\ l.copyNo = c.copyNo\ and\ l.dateDue < getdate\ and\ c.available = '0'}\ (\rho_r Borrower \times \rho_c BookCopy \times \rho_l BookLoan)$

```
select r.borrowerNo, r.borrowerName, r.borrowerAddress from Borrower r, BookLoan l,
BookCopy c
where r.borrowerNo = l.borrowerNo
and l.copyNo = c.copyNo
and l.dateDue < getdate
and c.available = '0'
```

14. Produce a report detailing how many times each book title has been borrowed.

RA: $\Pi_{b.title,c.ISBN,COUNT(copyNo)}\ \gamma_{ISBN,title,COUNT(copyNo)}\ \sigma_{l.copyNo = c.copyNo\ and\ c.ISBN = b.ISBN}\ (\rho_b Book \times \rho_c BookCopy \times \rho_l BookLoan)$

```
select b.title, c.ISBN, count(l.copyNo) from BookLoan l, BookCopy c, Book b
where l.copyNo = c.CopyNo
and c.ISBN = b.ISBN
group by c.ISBN, b.title
```

Task -3

```
USE LectureDB;

go
DROP TABLE IF EXISTS Possess
go

DROP TABLE IF EXISTS Take
go

DROP TABLE IF EXISTS DiscountCoupon
go

DROP TABLE IF EXISTS Student
go

DROP TABLE IF EXISTS Lecture
go

DROP TABLE IF EXISTS Instructor
go

DROP TABLE IF EXISTS LectureRoom
go

CREATE TABLE LectureRoom (
LR_id VARCHAR(10) NOT NULL,
LR_location VARCHAR(50),
LR_maximumstudent INT DEFAULT 0,
CONSTRAINT pk_LR_id PRIMARY KEY (LR_id)
);

CREATE TABLE Instructor (
I_id VARCHAR(10) NOT NULL,
I_name VARCHAR(30) NOT NULL,
I_office_num VARCHAR(10) NOT NULL,
I_phone VARCHAR(16) NOT NULL,
I_email VARCHAR(40) NOT NULL,
```

```
I_career VARCHAR(2000) NOT NULL,
I_field VARCHAR(50) NOT NULL,
CONSTRAINT pk_I_id PRIMARY KEY(I_id),
CONSTRAINT unique_I_OPE UNIQUE(I_office_num, I_phone, I_email)
);

CREATE TABLE Lecture (
L_id VARCHAR(10) NOT NULL,
L_name VARCHAR(30) NOT NULL,
L_fee INT DEFAULT 0,
L_startdate DATE NOT NULL,
L_enddate DATE NOT NULL,
L_length INT,
L_date VARCHAR(10) NOT NULL,
L_starttime TIME NOT NULL,
L_endtime TIME NOT NULL,
L_numberofstudents INT DEFAULT 0,
L_contents VARCHAR(2000),
I_id VARCHAR(10) NOT NULL,
LR_id VARCHAR(10) NOT NULL,
SetUp_date DATE NOT NULL,
CONSTRAINT pk_L_id PRIMARY KEY(L_id),
CONSTRAINT fk_I_idL FOREIGN KEY(I_id) REFERENCES Instructor(I_id) ON DELETE CASCADE,
CONSTRAINT fk_LR_idL FOREIGN KEY(LR_id) REFERENCES LectureRoom(LR_id) ON DELETE
CASCADE,
CONSTRAINT check_L_startdate CHECK(L_startdate <= L_enddate),
CONSTRAINT unique_L_name UNIQUE(L_name)
);

CREATE TABLE Student (
S_id VARCHAR(10) NOT NULL,
S_name VARCHAR(30) NOT NULL,
S_dateofbirth DATE,
S_phone VARCHAR(16) NOT NULL,
CONSTRAINT pk_S_id PRIMARY KEY(S_id),
CONSTRAINT unique_S_phone UNIQUE(S_phone)
);

CREATE TABLE DiscountCoupon (
```

```
DC_id INT NOT NULL,
Discount_rate INT NOT NULL,
Validity_date date NOT NULL,
L_id VARCHAR(10) NOT NULL,
Registration_date DATE NOT NULL,
CONSTRAINT pk_DC_id PRIMARY KEY(DC_id),
CONSTRAINT fk_L_idD FOREIGN KEY(L_id) REFERENCES Lecture(L_id) ON DELETE CASCADE,
CONSTRAINT check_Discount_rate CHECK(Discount_rate > 0 AND Discount_rate <= 100)
);

CREATE TABLE Take (
S_id VARCHAR(10) NOT NULL,
L_id VARCHAR(10) NOT NULL,
take_date DATE NOT NULL,
CONSTRAINT pk_SL_id PRIMARY KEY(S_id, L_id),
CONSTRAINT fk_S_idT FOREIGN KEY(S_id) REFERENCES Student(S_id) ON DELETE CASCADE,
CONSTRAINT fk_L_idT FOREIGN KEY(L_id) REFERENCES Lecture(L_id) ON DELETE CASCADE
);

CREATE TABLE Possess (
DC_id INT NOT NULL,
S_id VARCHAR(10) NOT NULL,
Possession_date date NOT NULL,
CONSTRAINT pk_DCS_id PRIMARY KEY(DC_id, S_id),
CONSTRAINT fk_DC_idP FOREIGN KEY(DC_id) REFERENCES DiscountCoupon(DC_id) ON DELETE
CASCADE,
CONSTRAINT fk_S_idP FOREIGN KEY(S_id) REFERENCES Student(S_id) ON DELETE CASCADE
);
```

```
use LectureDB
go

INSERT INTO LectureRoom VALUES ('LR01', '201', '20');
INSERT INTO LectureRoom VALUES ('LR02', '302', '10');
INSERT INTO LectureRoom VALUES ('LR03', '409', '20');
INSERT INTO LectureRoom VALUES ('LR04', '802', '10');
INSERT INTO LectureRoom VALUES ('LR05', '404', '10');

Select * from LectureRoom
```

```
INSERT INTO Instructor VALUES ('I01', 'Jeemin', '101', '010-1234-5678',
'119cloud@koreatech.ac.kr', 'professor 3 years', 'Korean literature');
INSERT INTO Instructor VALUES ('I02', 'Sooyeon', '102', '010-1004-8282',
'sooddong@koreatech.ac.kr', 'Doctoral degree', 'Mathematics');
INSERT INTO Instructor VALUES ('I03', 'Charin', '104', '010-5858-1234', 'chacha@koreatech.ac.kr',
'High school teacher 2 years', 'Chemical engineering');
INSERT INTO Instructor VALUES ('I04', 'Miri', '105', '010-4242-0000', 'abc@naver.com', '3 years of
teaching experience', 'Biotechnology');
INSERT INTO Instructor VALUES ('I05', 'May', '106', '010-0101-0101', 'haha@gmail.com', 'Master
degree', 'English literature');

Select * from Instructor

INSERT INTO Lecture VALUES ('L01', 'Math', '100000', '2022-10-01', '2022-12-16', '3', 'MON',
'09:00:00', '12:00:00', '10', 'Differential and integral calculus', 'I01', 'LR01', '2022-09-30');
INSERT INTO Lecture VALUES ('L02', 'Korean', '80000', '2022-08-26', '2022-10-01', '2', 'TUE',
'13:30:00', '15:30:00', '8', 'Grammar', 'I02', 'LR02', '2022-08-01');
INSERT INTO Lecture VALUES ('L03', 'Biology', '90000', '2022-10-01', '2022-12-15', '3', 'WED',
'10:00:00', '13:00:00', '10', 'Anatomy', 'I03', 'LR03', '2022-09-30');
INSERT INTO Lecture VALUES ('L04', 'Chemistry', '90000', '2022-10-10', '2022-12-15', '2', 'THU',
'12:00:00', '14:00:00', '9', 'Chemical formula', 'I04', 'LR04', '2022-09-30');
INSERT INTO Lecture VALUES ('L05', 'English', '100000', '2022-8-10', '2022-10-01', '1', 'FRI',
'13:00:00', '14:00:00', '7', 'Conversational English', 'I05', 'LR05', '2022-08-01');

Select * from Lecture

INSERT INTO Student VALUES ('2020126602', 'Yuri', '2001-01-01', '010-4241-1249');
INSERT INTO Student VALUES ('2021341234', 'Youheon', '2002-04-05', '010-1357-1113');
INSERT INTO Student VALUES ('2017332122', 'Rin', '1998-08-12', '010-2468-1012');
INSERT INTO Student VALUES ('2019126223', 'Siyeon', '2000-02-27', '010-1357-9111');
INSERT INTO Student VALUES ('2020341331', 'Rose', '2001-10-01', '010-1119-1119');

Select * from Student

INSERT INTO DiscountCoupon VALUES (10674, 30, '2022-12-31', 'L01', '2022-09-14');
INSERT INTO DiscountCoupon VALUES (86723, 15, '2022-12-31', 'L02', '2022-09-20');
INSERT INTO DiscountCoupon VALUES (65420, 20, '2022-12-31', 'L03', '2022-09-25');
INSERT INTO DiscountCoupon VALUES (97463, 25, '2022-12-31', 'L04', '2022-09-28');
INSERT INTO DiscountCoupon VALUES (36753, 40, '2022-12-31', 'L05', '2022-09-30');
```

```
Select * from DiscountCoupon

INSERT INTO Take VALUES ('2020126602', 'L01', '2022-11-10');
INSERT INTO Take VALUES ('2021341234', 'L02', '2022-09-04');
INSERT INTO Take VALUES ('2017332122', 'L03', '2022-10-27');
INSERT INTO Take VALUES ('2019126223', 'L04', '2022-11-23');
INSERT INTO Take VALUES ('2020341331', 'L05', '2022-09-15');

Select * from Take

INSERT INTO Possess VALUES (10674, '2020126602', '2022-07-28');
INSERT INTO Possess VALUES (86723, '2021341234', '2022-08-25');
INSERT INTO Possess VALUES (65420, '2017332122', '2022-07-15');
INSERT INTO Possess VALUES (97463, '2019126223', '2022-08-03');
INSERT INTO Possess VALUES (36753, '2020341331', '2022-09-16');

Select * from Possess
```