

REPORT

데이터베이스 과제 #1

제출일 : 2019.09.18. ~ 2019.09.25. 23:00



Index_

- 과제 1. create문을 사용해서 테이블을 생성, insert문을 사용해 데이터를 삽입, select문을 사용해 데이터 출력
 - 1.1. Table 생성
 - 1.2. Data 삽입
 - 1.3. Data 확인
- 과제 2. 적절한 디렉토리에 tablespace 생성하기
- 과제 3. 사용자 계정 생성해서 권한부여한뒤 접속해보기
- 과제 4. ERwin Data Modeler로 E/R(Entity/Relationship) Diagram 표현하기
- 과제 0. Code
- 과제 5. 소감

이름 : 구명회

학번 : 20154215

분반 : 01분반

과제 1.

create문을 사용해서 테이블을 생성, insert문을 사용해 데이터를 삽입, select문을 사용해 데이터 출력

※ S, P, SP, J, SPJ, Dept, Emp# 테이블의 이름앞에 Homework_를 붙였습니다.

0. Code에서 주석 달겠습니다.

1.1. Table 생성

S, P, SP, J, SPJ, Dept, Emp# 테이블을 생성했습니다.

```
SQL>
SQL> Create Table Homework_S(
2 S# Char(5),
3 SNAME Char(20),
4 STATUS Numeric(15),
5 CITY Char(15),
6 Primary Key(S#));
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_P(
2 P# Char(6),
3 PNAME Char(20),
4 COLOR Char(6),
5 WEIGHT numeric(5,1),
6 CITY Char(15),
7 Primary Key(P#));
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_SP(
2 S# Char(5),
3 P# Char(6),
4 QTY Numeric(9),
5 Primary Key(S#, P#),
6 Foreign Key(S#) References Homework_S,
7 Foreign Key(P#) References Homework_P);
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_J(
2 J# Char(5),
3 JNAME Char(20),
4 CITY Char(15),
5 Primary Key(J#));
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_SPJ(
2 S# Char(5),
3 P# Char(6),
4 J# Char(5),
5 QTY Numeric(9),
6 Primary Key(S#, P#, J#),
7 Foreign Key(S#) References Homework_S,
8 Foreign Key(P#) References Homework_P,
9 Foreign Key(J#) References Homework_J);
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_Dept(
2 DEPT# Char(5),
3 DNAME Char(20),
4 BUDGET Char(6),
5 Primary key(DEPT#));
```

테이블이 생성되었습니다.

```
SQL>
SQL> Create Table Homework_Emp#(
2 EMP# Char(5),
3 ENAME Char(20),
4 DEPT# Char(5),
5 SALARY Char(5),
6 Primary key(EMP#),
7 Foreign Key(DEPT#) References Homework_Dept);
```

테이블이 생성되었습니다.

1.2. Data 삽입

S, P, SP, J, SPJ, Dept, Emp# 테이블에 데이터를 삽입했습니다.

```
SQL>
SQL> Insert Into Homework_S
2 Values('S1', '홍길동', 20, '서울');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_S
2 Values('S2', '손오공', 10, '부산');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_S
2 Values('S3', '박문수', 30, '부산');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_S
2 Values('S4', '신수동', 20, '서울');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_S
2 Values('S5', '방월동', 30, '경주');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P1', '너트', '빨강', 12, '서울');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P2', '볼트', '초록', 17, '부산');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P3', '스크류', '파랑', 17, '광주');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P4', '스크류', '빨강', 14, '서울');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P5', '캠', '파랑', 12, '부산');
```

1 개의 행이 만들어졌습니다.

```
SQL>
SQL> Insert Into Homework_P
2 Values('P6', '꼭', '빨강', 19, '서울');
```

1 개의 행이 만들어졌습니다.

1.3. Data 확인

S, P, SP, J, SPJ, Dept, Emp# 테이블의 데이터를 조회했습니다.

SQL>
SQL> Select * From Homework_S;

S#	SNAME	STATUS	CITY
S1	홍길동	20	서울
S2	홍길동	10	부산
S3	박수홍	30	부산
S4	신수홍	20	서울
S5	방현영	30	경주

6 개의 행이 선택되었습니다.

SQL>
SQL> Select * From Homework_P;

P#	PNAME	COLOR	WEIGHT	CITY
P1	너트	빨간색	12	서울
P2	볼트	파란색	17	부산
P3	스crew	파란색	17	부산
P4	스crew	파란색	14	서울
P5	캡	파란색	12	부산
P6	캡	파란색	19	서울

6 개의 행이 선택되었습니다.

SQL>
SQL> Select * From Homework_J;

J#	JNAME	CITY
J1	정렬기	부산
J2	정렬기	부산
J3	정렬기	부산
J4	정렬기	부산
J5	정렬기	부산
J6	정렬기	부산
J7	정렬기	부산

7 개의 행이 선택되었습니다.

SQL>
SQL> Select * From Homework_SP;

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

12 개의 행이 선택되었습니다.

SQL>
SQL> Select * From Homework_SPJ;

S#	P#	J#	QTY
S3	P3	J3	300
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

25 개의 행이 선택되었습니다.

과제 2, 3

적절한 디렉토리에 tablespace 생성하고, 사용자 계정 생성해서 권한부여한뒤 접속해보기

※ 과제2와 과제3의 내용이 너무 적어 사진 한 장으로 마무리 하겠습니다.

코드는 0. Code에서 주석 달겠습니다.

SQL Plus

SQL>
SQL> Conn Sys As Sysdba.
비밀번호 입력;
연결되었습니다.
SQL>
SQL> Create Tablespace Homework1
2 Datafile 'C:\database\Homework1\Homework01_data01.dbf' Size 20M;
테이블스페이스가 생성되었습니다.
SQL>
SQL> Alter Tablespace Homework1
2 Add Datafile 'C:\database\Homework1\Homework01_data02.dbf' Size 20M;
테이블스페이스가 변경되었습니다.
SQL>
SQL> Select tablespace_name From dba_data_files;
TABLESPACE_NAME

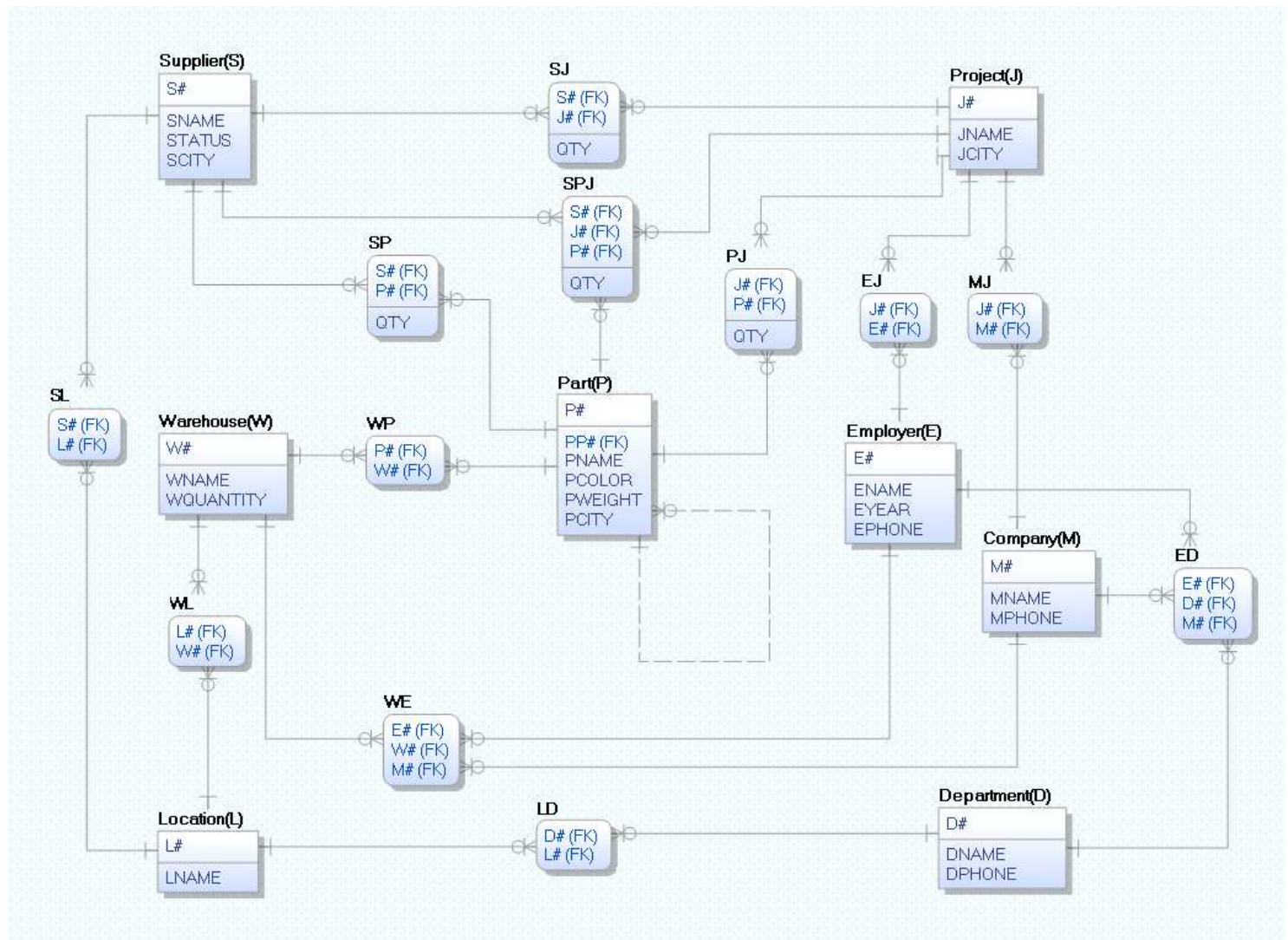
USERS
UNDOTBS1
SYSAUX
SYSTEM
EXAMPLE
HOMEWORK1
HOMEWORK1
7 개의 행이 선택되었습니다.
SQL>
SQL> Create user MH
2 Identified by 1234
3 Default Tablespace Homework1
4 Quota Unlimited on Homework1;
사용자가 생성되었습니다.
SQL>
SQL> Grant Connect, Resource to MH;
권한이 부여되었습니다.
SQL>
SQL>

SQL Plus

SQL*Plus: Release 11.2.0.1.0 Production on 일 9월 22 14:35:44 2019
Copyright (c) 1982, 2010, Oracle. All rights reserved.
사용자명 입력: MH
비밀번호 입력:
다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL>

과제 4

ERwin Data Modeler로 E/R(Entity/Relationship) Diagram 표현하기



Entity List

Supplier(S), Project(J), Part(P), Warehouse(W), Location(L), Department(D), Employer(E), Company(M)

Relationship List

SJ, SP, SL, SPJ, PJ, PP, EJ, MJ, WP, WL, WE, LD, ED

Supplier(S)	SJ	SP	SL	SPJ	//
Project(J)	SJ	PJ	EJ	MJ	SPJ
Part(P)	SP	WP	PP	PJ	SPJ
Warehouse(W)	WP	WL	WE	//	
Location(L)	SL	WL	LD	//	
Department(D)	LD	ED	//		
Employer(E)	EJ	WE	ED	//	
Company(M)	MJ	WE	ED	//	

Table Attribute 설정

- 편의를 위해 대부분의 자료형을 Char으로 설정했습니다.

Identifying relationship / non-Identifying relationship 설정

- 관계를 설정할 때 PP를 제외한 모든 관계를 Identifying relationship으로 설정했습니다.
왜냐하면 과제 1에서 설정한 모든 테이블에서, 부모 테이블의 기본키와 자식 테이블의 기본키가 같기 때문입니다.
예를들어 SJ관계에서, S테이블과 J테이블의 기본키가 SJ테이블의 기본키이므로 Identifying-relationship입니다.
(S#) (J#) (S#, J#)
- PP relationship은 non-Identifying relationship으로 설정했습니다.
재귀관계는 위의 그림처럼 non-Identifying relationship으로 설정해야합니다.
책 15p 그림 1.6에서 PP는 재귀관계입니다. 관계를 설정하고 속성에 PP#을 외래키로 설정했습니다.
- Many to Many relationship은 고려하지 않았습니다.
- 0. Code에서 각 테이블에 대해서 주석을 달았습니다.

과제 0. Code

※ 과제 1 ~ 4의 코드를 주석 추가해서 순서대로 첨부했습니다.

SQL*Plus: Release 11.2.0.1.0 Production on 수 9월 25 01:14:33 2019

Copyright (c) 1982, 2010, Oracle. All rights reserved.

//

사용자명 입력: **scott** // user 계정, scott으로 로그인했습니다.

비밀번호 입력:

다음에 접속됨:

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

//

[Create Table]

SQL> **Create Table Homework_S**(// Homework_S라는 테이블을 만들었습니다.

2 S# Char(5), // 고유번호를 나타낼 Char형 S#

3 SNAME Char(20), // 이름을 나타낼 Char형 SNAME

4 STATUS Numeric(15), // 상태를 나타낼 Numeric형 STATUS

5 SCITY Char(15), // 도시 이름을 나타낼 Char형 SCITY

6 Primary Key(S#)); // 고유번호를 기본키로 설정

테이블이 생성되었습니다.

SQL> **Create Table Homework_P**(// Homework_P라는 테이블을 만들었습니다.

2 P# Char(6), // 고유번호를 나타낼 Char형 P#

3 PNAME Char(20), // 이름을 나타낼 Char형 PNAME

4 PCOLOR Char(6), // 색을 나타낼 Char형 PCOLOR

5 PWEIGHT numeric(5,1), // 무게를 나타낼 Numeric형 PWEIGHT

6 PCITY Char(15), // 도시 이름을 나타낼 Char형 PCITY

7 Primary Key(P#)); // 고유번호를 기본키로 설정

테이블이 생성되었습니다.

```
SQL> Create Table Homework_SP(           // Homework_SP라는 테이블을 만들었습니다.
2  S#          Char(5),                  //
3  P#          Char(6),                  // S#과 P# >> 5 ~ 7LINE
4  QTY         Numeric(9),              // 뭘 나타내는지 모르겠지만 Numeric형 QTY
5  Primary Key(S#, P#),                 // S#과 P#을 기본키로 설정
6  Foreign Key(S#) References Homework_S, // Homework_S의 기본키 S#를 외래키로 설정
7  Foreign Key(P#) References Homework_P); // Homework_P의 기본키 P#를 외래키로 설정
테이블이 생성되었습니다.
```

```
SQL> Create Table Homework_J(           // Homework_J라는 테이블을 만들었습니다.
2  J#          Char(5),                  // 고유번호를 나타낼 Char형 J#
3  JNAME       Char(20),                 // 이름을 나타낼 Char형 JNAME
4  JCITY       Char(15),                 // 도시 이름을 나타낼 Char형 JCITY
5  Primary Key(J#));                    // 고유번호를 기본키로 설정
테이블이 생성되었습니다.
```

```
SQL> Create Table Homework_SPJ(         // Homework_SPJ라는 테이블을 만들었습니다.
2  S#          Char(5),                  //
3  P#          Char(6),                  //
4  J#          Char(5),                  // S#과 P#과 J# >>> 6 ~ 9LINE
5  QTY         Numeric(9),              // 뭘 나타내는지 모르겠지만 Numeric형 QTY
6  Primary Key(S#, P#, J#),             // S#과 P#과 J#을 기본키로 설정
7  Foreign Key(S#) References Homework_S, // Homework_S의 기본키 S#를 외래키로 설정
8  Foreign Key(P#) References Homework_P, // Homework_P의 기본키 P#를 외래키로 설정
9  Foreign Key(J#) References Homework_J); // Homework_J의 기본키 J#를 외래키로 설정
테이블이 생성되었습니다.
```

```
SQL> Create Table Homework_Dept(        // Homework_Dept라는 테이블을 만들었습니다.
2  DEPT#       Char(5),                  // 고유번호를 나타낼 Char형 D#
3  DNAME       Char(20),                 // 이름을 나타낼 Char형 DNAME
4  BUDGET      Char(6),                  // 예산을 나타낼 Char형 BUDGET
5  Primary key(DEPT#));                  // 고유번호를 기본키로 설정
테이블이 생성되었습니다.
```

```
SQL> Create Table Homework_Emp#(        // Homework_Dept라는 테이블을 만들었습니다.
2  EMP#        Char(5),                  // 고유번호를 나타낼 Char형 D#
3  ENAME       Char(20),                 // 이름을 나타낼 Char형 DNAME
4  DEPT#       Char(5),                  // DEPT# >> 7 LINE
5  SALARY      Char(5),                  // 월급을 나타낼 Char형 SALARY
6  Primary key(EMP#),                    // EMP#을 기본키로 설정
7  Foreign Key(DEPT#) References Homework_Dept); // Homework_Dept의 기본키 DEPT#를 외래키로 설정
테이블이 생성되었습니다.
```

////////////////////////////////////

[Insert Data, Select Data]

※ 데이터의 삽입과 검색에 대해서는 중복되는 SQL문이 많으므로 하나하나 설명하지 않고 맨 위에 Insert into~와 Select from~을 어떻게 사용하는지만 적었습니다.
그리고 삽입하는 SQL문이 많아서 처음과 끝부분만 남기고 제거했습니다.
처음과 끝부분이 포함된 코드를 txt파일에 정리해서 제출하겠습니다.

```
// Insert into [Table]    // data를 삽입하고싶은 Table
// Values([column]);    // column 형식에 맞는 값들
SQL> Insert Into Homework_S Values('S1', '홍길동', 20, '서울');
1 개의 행이 만들어졌습니다.
```

```
...
SQL> Insert Into Homework_S Values('S5', '방월동', 30, '경주');
1 개의 행이 만들어졌습니다.
```

```
// Select [column]      // 출력할 column
// From [Table];        // 출력하고싶은 Table
SQL> Select * From Homework_S;
S#      SNAME                      STATUS SCITY
-----
S1      홍길동                      20 서울
S2      손오공                      10 부산
S3      박문수                      30 부산
S4      신수동                      20 서울
S5      방월동                      30 경주
```

```
SQL> Insert Into Homework_P Values('P1', '너트', '빨강', 12, '서울');
1 개의 행이 만들어졌습니다.
```

```
...
SQL> Insert Into Homework_P Values('P6', '콧', '빨강', 19, '서울');
1 개의 행이 만들어졌습니다.
```

```
SQL> Select * From Homework_P;
P#      PNAME                      PCOLOR    PWEIGHT PCITY
-----
P1      너트                      빨강      12 서울
P2      볼트                      초록      17 부산
P3      스크류                    파랑      17 광주
P4      스크류                    빨강      14 서울
P5      캡                        파랑      12 부산
P6      콧                        빨강      19 서울
6 개의 행이 선택되었습니다.
```

```
SQL> Insert Into Homework_SP Values('S1', 'P1', 300);
1 개의 행이 만들어졌습니다.
```

```
...
SQL> Insert Into Homework_SP Values('S4', 'P5', 400);
1 개의 행이 만들어졌습니다.
```


SQL> Select * From Homework_SP;

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

12 개의 행이 선택되었습니다.

SQL> Insert Into Homework_J Values('J1', '정렬기', '부산');

1 개의 행이 만들어졌습니다.

...

SQL> Insert Into Homework_J Values('J7', '테이블', '서울');

1 개의 행이 만들어졌습니다.

SQL> Select * From Homework_J;

J#	JNAME	JCITY
J1	정렬기	부산
J2	천공기	광주
J3	판독기	경주
J4	콘솔	경주
J5	인쇄기	서울
J6	터미널	인천
J7	테이블	서울

7 개의 행이 선택되었습니다.

SQL> Insert Into Homework_SPJ Values('S3', 'P3', 'J3', 300);

1 개의 행이 만들어졌습니다.

...

SQL> Insert Into Homework_SPJ Values('S5', 'P6', 'J4', 500);

1 개의 행이 만들어졌습니다.

SQL> Select * From Homework_SPJ;

S#	P#	J#	QTY
S3	P3	J3	300
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200

S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

25 개의 행이 선택되었습니다.

SQL> Insert Into Homework_Dept Values('D1', '마케팅부', '10M');

1 개의 행이 만들어졌습니다.

...

SQL> Insert Into Homework_Dept Values('D3', '연구부', '5M');

1 개의 행이 만들어졌습니다.

SQL> Select * From Homework_Dept;

DEPT#	DNAME	BUDGET
-----	-----	-----
D1	마케팅부	10M
D2	개발부	12M
D3	연구부	5M

SQL> Insert Into Homework_Emp# Values('E1', '이몽룡', 'D1', '40K');

1 개의 행이 만들어졌습니다.

...

SQL> Insert Into Homework_Emp# Values('E4', '성춘향', 'D2', '35K');

1 개의 행이 만들어졌습니다.

SQL> Select * From Homework_Emp#;

EMP#	ENAME	DEPT#	SALAR
-----	-----	-----	-----
E1	이몽룡	D1	40K
E2	장길산	D1	42K
E3	허참봉	D2	30K
E4	성춘향	D2	35K

////////////////////////////////////

[과제2, 과제3 Create Tablespace, Create User]

SQL> show user // 현재 접속중인 계정을 출력합니다.

USER은 "SCOTT"입니다

SQL> conn sys as sysdba // 관리자 계정으로 접속합니다.

비밀번호 입력: // sys가 system보다 권한이 높습니다.

연결되었습니다.

// Homework1이라는 이름의 Tablespace를 생성합니다.

// Tablespace를 생성하기 전 C:\Wdatabase 디렉터리에 Homework1 디렉터리를 생성했습니다.

// Homework1 디렉터리에 크기가 20M인 데이터파일 Homework01_data01.dbf를 생성합니다.

SQL> Create Tablespace Homework1

2 Datafile 'C:\Wdatabase\Homework1\Homework01_data01.dbf' Size 20M;

테이블스페이스가 생성되었습니다.

// 데이터파일의 이름을 변경합니다.

// Homework1 디렉터리에 크기가 20M인 데이터파일 Homework01_data02.dbf로 변경합니다.

SQL> Alter Tablespace Homework1

2 Add Datafile 'C:\Wdatabase\Homework1\Homework01_data02.dbf' Size 20M;

테이블스페이스가 변경되었습니다.

SQL> Select tablespace_name From dba_data_files; // 테이블 스페이스에 저장된 테이블을 조회합니다.

TABLESPACE_NAME

USERS

UNDOTBS1

SYSAUX

SYSTEM

EXAMPLE

HOMEWORK1

HOMEWORK1

7 개의 행이 선택되었습니다.

SQL> Create user MH // MH라는 이름의 사용자 계정을 생성합니다.

2 Identified by 1234 // 비밀번호는 1234입니다.

3 Default Tablespace Homework1 // 사용할 Tablespace는 2에서 만든 Homework1입니다.

4 Quota Unlimited on Homework1; // Homework1의 용량은 Unlimited입니다.

사용자가 생성되었습니다.

SQL> Grant Connect, Resource to MH; // 만든 사용자 계정에 권한을 부여했습니다.

권한이 부여되었습니다.

SQL> connect MH/1234 // 만들었던 사용자계정으로 로그인합니다.

연결되었습니다.

SQL> show user; // 현재 접속중인 사용자를 표시합니다.

USER은 "MH"입니다

//

[Create Table]

```
CREATE TABLE Company(M)(          // Company(M)라는 테이블을 만들었습니다.
M#          CHAR(5) NOT NULL ,      // 회사의 고유번호를 나타낼 Char형 M#
MNAME       CHAR(20) NULL ,         // 회사의 이름을 나타낼 Char형 MNAME
MPHONE      CHAR(15) NULL ,         // 회사의 전화번호를 나타낼 Char형 MPHONE
PRIMARY KEY (M#));                  // 회사의 고유번호를 기본키로 설정
```

```
CREATE TABLE Department(D)(        // Department(D)라는 테이블을 만들었습니다.
D#          CHAR(5) NOT NULL ,      // 부서의 고유번호를 나타낼 Char형 D#
DNAME       CHAR(20) NULL ,         // 부서의 이름을 나타낼 Char형 DNAME
DPHONE      CHAR(15) NULL ,         // 부서의 전화번호를 나타낼 Char형 DPHONE
PRIMARY KEY (D#));                  // 부서의 고유번호를 기본키로 설정
```

```
CREATE TABLE Employer(E)(          // Employer(E)라는 테이블을 만들었습니다.
E#          CHAR(5) NOT NULL ,      // 고용인의 고유번호를 나타낼 Char형 E#
ENAME       CHAR(20) NULL ,         // 고용인의 이름을 나타낼 Char형 ENAME
EYEAR       CHAR(5) NULL ,          // 고용인의 나이를 나타낼 Char형 EPHONE
EPHONE      CHAR(15) NULL ,         // 고용인의 전화번호를 나타낼 Char형 EPHONE
PRIMARY KEY (E#));                  // 고용인의 고유번호를 기본키로 설정
```

```
CREATE TABLE Part(P)(              // Part(P)라는 테이블을 만들었습니다.
P#          CHAR(6) NOT NULL ,      // 부품의 고유번호를 나타낼 Char형 P#
PP#         CHAR(6) NOT NULL ,      // 재귀관계로, 부품의 고유번호를 나타낼 Char형 P##
PNAME       CHAR(20) NULL ,         // 부품의 이름을 나타낼 Char형 PNAME
PCOLOR      CHAR(6) NULL ,          // 부품의 색을 나타낼 Char형 PCOLOR
PWEIGHT     NUMERIC(5,1) NULL ,     // 부품의 무게를 나타낼 Numeric형 PWEIGHT
PCITY       CHAR(15) NULL ,         // 부품의 원산지(?)를 나타낼 Char형 PCITY
PRIMARY KEY (P#),                  // 부품의 고유번호를 기본키로 설정
FOREIGN KEY (PP#) REFERENCES Part(P) (P#)); // Part_P(P)의 기본키 P#를 외래키로 설정
```

```
CREATE TABLE Warehouse(W)(         // Warehouse(W)라는 테이블을 만들었습니다.
W#          CHAR(5) NOT NULL ,      // 창고의 고유번호를 나타낼 Char형 W#
WNAME       CHAR(20) NULL ,         // 창고의 이름을 나타낼 Char형 WNAME
WQUANTITY   NUMERIC(7) NULL ,       // 창고에 있는 부품들의 수량을 나타낼 NUMERIC형 WQUANTITY
PRIMARY KEY (W#));                  // 창고의 고유번호를 기본키로 설정
```

```
CREATE TABLE Supplier(S)(          // Supplier(S)라는 테이블을 만들었습니다.
S#          CHAR(5) NOT NULL ,      // 공급자의 고유번호를 나타낼 Char형 S#
SNAME       CHAR(20) NULL ,         // 공급자의 이름을 나타낼 Char형 SNAME
STATUS      NUMERIC(15) NULL ,      // 공급자의 상태를 나타낼 NUMERIC형 STATUS
SCITY       CHAR(15) NULL ,         // 공급자가 현재 있는 도시를 나타낼 Char형 SCITY
PRIMARY KEY (S#));                  // 공급자의 고유번호를 기본키로 설정
```



```

CREATE TABLE Project(J)(          // Project(J)라는 테이블을 만들었습니다.
J#          CHAR(5) NOT NULL ,    // 프로젝트의 고유번호를 나타낼 Char형 J#
JNAME       CHAR(20) NULL ,       // 프로젝트의 이름을 나타낼 Char형 JNAME
JCITY       CHAR(15) NULL ,       // 프로젝트를 수행할 도시를 나타낼 Char형 JCITY
PRIMARY KEY (J#));                // 프로젝트의 고유번호를 기본키로 설정

```

```

CREATE TABLE Location(L)(        // Location(L)라는 테이블을 만들었습니다.
L#          CHAR(5) NOT NULL ,    // 위치의 고유번호를 나타낼 Char형 L#
LNAME       CHAR(20) NULL ,       // 위치의 이름을 나타낼 Char형 LNAME
PRIMARY KEY (L#));                // 위치의 고유번호를 기본키로 설정

```

```

CREATE TABLE SPJ(                // SPJ라는 테이블을 만들었습니다.
S#          CHAR(5) NOT NULL,     //
J#          CHAR(5) NOT NULL,     //
P#          CHAR(6) NOT NULL,     //
QTY         NUMBER(7) NULL,       // 월 나타내는지 모르겠지만 NUMBER형 QTY
PRIMARY KEY (S#,J#,P#),           // S#과 J#과 P#을 기본키로 설정
FOREIGN KEY (S#) REFERENCES Supplier(S) (S#), // Supplier(S)의 기본키 S#을 외래키로 설정
FOREIGN KEY (J#) REFERENCES Project(J) (J#),  // Project(J)의 기본키 J#을 외래키로 설정
FOREIGN KEY (P#) REFERENCES Part(P) (P#));    // Part(P)의 기본키 P#을 외래키로 설정

```

```

CREATE TABLE WP(                 // WP라는 테이블을 만들었습니다.
P#          CHAR(6) NOT NULL ,    //
W#          CHAR(5) NOT NULL ,    //
PRIMARY KEY (P#,W#),              // P#과 W#을 기본키로 설정
FOREIGN KEY (P#) REFERENCES Part(P) (P#),    // Part(P)의 기본키 P#을 외래키로 설정
FOREIGN KEY (W#) REFERENCES Warehouse(W) (W#)); // Warehouse(W)의 기본키 W#을 외래키로 설정

```

```

CREATE TABLE WL(                 // WL라는 테이블을 만들었습니다.
L#          CHAR(5) NOT NULL ,    //
W#          CHAR(5) NOT NULL ,    //
PRIMARY KEY (L#,W#),              // L#과 W#을 기본키로 설정
FOREIGN KEY (L#) REFERENCES Location(L) (L#), // Location(L)의 기본키 L#을 외래키로 설정
FOREIGN KEY (W#) REFERENCES Warehouse(W) (W#)); // Warehouse(W)의 기본키 W#을 외래키로 설정

```

```

CREATE TABLE WE(                 // WE라는 테이블을 만들었습니다.
E#          CHAR(5) NOT NULL ,    //
W#          CHAR(5) NOT NULL ,    //
M#          CHAR(5) NOT NULL ,    //
PRIMARY KEY (E#,W#,M#),           // E#과 W#과 M#을 기본키로 설정
FOREIGN KEY (E#) REFERENCES Employer(E) (E#), // Employer(E)의 기본키 E#을 외래키로 설정
FOREIGN KEY (W#) REFERENCES Warehouse(W) (W#), // Warehouse(W)의 기본키 W#을 외래키로 설정
FOREIGN KEY (M#) REFERENCES Company(M) (M#)); // Company(M)의 기본키 M#을 외래키로 설정

```

```

CREATE TABLE SP(                                // SP라는 테이블을 만들었습니다.
S#      CHAR(5) NOT NULL ,                        //
P#      CHAR(6) NOT NULL ,                        //
QTY     NUMBER(9) NULL ,                          // 뭘 나타내는지 모르겠지만 NUMBER형 QTY
PRIMARY KEY (S#,P#),                             // S#과 P#을 기본키로 설정
FOREIGN KEY (S#) REFERENCES Supplier(S) (S#),     // Supplier(S)의 기본키 S#을 외래키로 설정
FOREIGN KEY (P#) REFERENCES Part(P) (P#));         // Part(P)의 기본키 P#을 외래키로 설정

```

```

CREATE TABLE MJ(                                // MJ라는 테이블을 만들었습니다.
J#      CHAR(5) NOT NULL ,                        //
M#      CHAR(5) NOT NULL ,                        //
PRIMARY KEY (J#,M#),                             // J#과 M#을 기본키로 설정
FOREIGN KEY (J#) REFERENCES Project(J) (J#),       // Project(J)의 기본키 J#을 외래키로 설정
FOREIGN KEY (M#) REFERENCES Company(M) (M#));      // Company(M)의 기본키 M#을 외래키로 설정

```

```

CREATE TABLE SL(                                // SL라는 테이블을 만들었습니다.
S#      CHAR(5) NOT NULL ,                        //
L#      CHAR(5) NOT NULL ,                        //
PRIMARY KEY (S#,L#),                             // S#과 L#을 기본키로 설정
FOREIGN KEY (S#) REFERENCES Supplier(S) (S#),     // Supplier(S)의 기본키 S#을 외래키로 설정
FOREIGN KEY (L#) REFERENCES Location(L) (L#));     // Location(L)의 기본키 L#을 외래키로 설정

```

```

CREATE TABLE SJ(                                // SJ라는 테이블을 만들었습니다.
S#      CHAR(5) NOT NULL ,                        //
J#      CHAR(5) NOT NULL ,                        //
QTY     NUMBER(9) NULL ,                          // 뭘 나타내는지 모르겠지만 NUMBER형 QTY
PRIMARY KEY (S#,J#),                             // S#과 J#을 기본키로 설정
FOREIGN KEY (S#) REFERENCES Supplier(S) (S#),     // Supplier(S)의 기본키 S#을 외래키로 설정
FOREIGN KEY (J#) REFERENCES Project(J) (J#));      // Project(J)의 기본키 J#을 외래키로 설정

```

```

CREATE TABLE PJ(                                // PJ라는 테이블을 만들었습니다.
J#      CHAR(5) NOT NULL ,                        //
P#      CHAR(6) NOT NULL ,                        //
QTY     NUMBER(9) NULL ,                          // 뭘 나타내는지 모르겠지만 NUMBER형 QTY
PRIMARY KEY (J#,P#),                             // J#과 P#을 기본키로 설정
FOREIGN KEY (J#) REFERENCES Project(J) (J#),       // Project(J)의 기본키 J#을 외래키로 설정
FOREIGN KEY (P#) REFERENCES Part(P) (P#));         // Part(P)의 기본키 P#을 외래키로 설정

```

```

CREATE TABLE LD(                                // LD라는 테이블을 만들었습니다.
D#      CHAR(5) NOT NULL ,                        //
L#      CHAR(5) NOT NULL ,                        //
PRIMARY KEY (D#,L#),                             // D#과 L#을 기본키로 설정
FOREIGN KEY (D#) REFERENCES Department(D) (D#),    // Department(D)의 기본키 D#을 외래키로 설정
FOREIGN KEY (L#) REFERENCES Location(L) (L#));     // Location(L)의 기본키 L#을 외래키로 설정

```

```

CREATE TABLE ED(                                // ED라는 테이블을 만들었습니다.
E#      CHAR(5) NOT NULL ,                        //
D#      CHAR(5) NOT NULL ,                        //
M#      CHAR(5) NOT NULL ,                        //
PRIMARY KEY (E#,D#,M#),                          // E#과 D#과 M#을 기본키로 설정
FOREIGN KEY (E#) REFERENCES Employer(E) (E#),    // Employer(E)의 기본키 E#을 외래키로 설정
FOREIGN KEY (D#) REFERENCES Department(D) (D#),   // Department(D)의 기본키 D#을 외래키로 설정
FOREIGN KEY (M#) REFERENCES Company(M) (M#));     // Company(M)의 기본키 M#을 외래키로 설정

```

```

CREATE TABLE EJ(                                // EJ라는 테이블을 만들었습니다.
J#      CHAR(5) NOT NULL ,                        //
E#      CHAR(5) NOT NULL ,                        //
PRIMARY KEY (J#,E#),                            // J#과 E#을 기본키로 설정
FOREIGN KEY (J#) REFERENCES Project(J) (J#),     // Project(J)의 기본키 J#을 외래키로 설정
FOREIGN KEY (E#) REFERENCES Employer(E) (E#));   // Employer(E)의 기본키 E#을 외래키로 설정

```

////////////////////////////////////

5. 소감

사실 저는 이번에 프로그램 설치하는게 역대급으로 오래 걸렸습니다.

Oracle 11g r2과 ERwin 설치하는데 추석이 겹쳐있어서 이주일넘게 지연되었습니다.

제가 ERwin라이센스를 못찾은건지 안온건지 모르겠지만 아무튼 못찾아서 메일을 보냈는데 추석임에도 불구하고 직원분이 친절하게 라이센스를 알려주셔서 다행스럽게 설치를 완료했습니다.

이번에 알게된점은 문의메일을 보내면 친절하게 답을 받을수 있다는 것을 실제로 체험해보았습니다.

과제를 하면서 어려웠던점은 SQL 언어의 문법, 자료형, 데이터를 출력하는법등을 하나하나 찾아봐야하는데 워낙 기초적인것이라 그런건지 제가 못찾아서 그런건지 몰라도 비슷한말에 제가 원하는 정보는 별로 없었고, 전문적인 분야라서 더욱 정보가 적어서 카페나 db사이트까지 뒤져서 찾아보는 것이 힘들었습니다.

제가 이번에 과제를 하면서 개인적으로 해결했던 문제 세 개를 제출합니다!

1) 과제1에서 homework_p 테이블 자료를 출력할 때 소숫점 0이 출력이 안되는 문제가 있었습니다.

교수님에게 여쭙봤더니 float이나 double로 바꾸면 된다고 하셨는데 여전히 출력되지 않았습니다.

문제를 to_char()을 사용해서 강제로 소숫점을 출력할 수 있었습니다.

```

SQL> select * from homework_p;

```

P#	PNAME	PCOLOR	PWEIGHT	PCITY
P1	너트	빨강	12	서울
P2	볼트	초록	17	부산
P3	스크류	파랑	17	광주
P4	스크류	빨강	14	서울
P5	캐폭	파랑	12	부산
P6	목	빨강	19	서울

6 개의 행이 선택되었습니다.

```

SQL> select P#, PNAME, PCOLOR, to_char(PWEIGHT, 'FM99.0'), PCITY from homework_p;

```

P#	PNAME	PCOLOR	TO_CH	PCITY
P1	너트	빨강	12.0	서울
P2	볼트	초록	17.0	부산
P3	스크류	파랑	17.0	광주
P4	스크류	빨강	14.0	서울
P5	캐폭	파랑	12.0	부산
P6	목	빨강	19.0	서울

6 개의 행이 선택되었습니다.

2) 과제량은 상관없었지만 제가 만들었던 테이블들이 뭐가 있었는지 확인하고 싶었습니다.

테이블의 조회는 다음과 같이 조회할 수 있었습니다.

이 명령어는 과제1을 할 때 궁금해서 찾아봤었는데 tablespace만드는 과제에서 테이블 목록을 조회하는 명령어와 비슷했습니다.

(Select tablespace_name From dba_data_files; // 테이블 스페이스에 저장된 테이블을 조회합니다.)

```
SQL> select table_name from tabs;
```

TABLE_NAME
DEPT
EMP
BONUS
SALGRADE
HOMEWORK_DEPT
HOMEWORK_EMP#
HOMEWORK_S
HOMEWORK_P
HOMEWORK_SP
HOMEWORK_J
HOMEWORK_SPJ
MAPLESTORY
TEST1

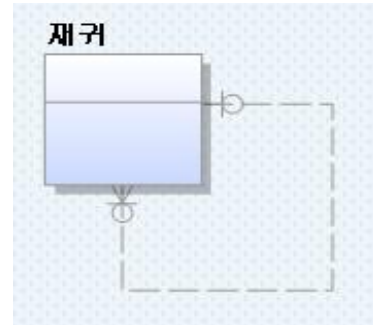
13 개의 행이 선택되었습니다.

3) Identifying-relationship과 non-Identifying-relationship에 대해서입니다.

4번과제에서 PP를 그릴 때 non-relationship인건 알겠는데 이걸 한줄로 연결해야할지 두줄로 연결해야할지 고민했었습니다.

1:1, 1:M, N:M, 정상적 관계나 슈퍼타입-서브타입관계 식별/비식별 관계 등 표현이 여러 가지가 있었습니다.

PP를 그리기 위해 필요했던 병렬관계와 재귀관계의 표현은 다음과 같이 표현합니다.



책에서는 두줄로 표현되어 있었는데, 교수님이 재귀관계라고 하셔서 병렬관계로 표현할지 재귀관계로 표현할지 고민했는데, 교수님이 재귀관계라고 하셔서 PP를 재귀관계로 표현했습니다.

PP 이외에도 나머지 관계들을 잘 그린건지 모르겠어서 계속 인터넷 자료들과 비교해가면서 맞는지 아닌지 생각을 많이 했습니다. 첫 번째 과제가 이렇게 어려울 줄 몰랐고, 다음 과제나 그 이후의 학업이나 취업 등에 대비해서, 앞으로도 공부를 더 열심히 하도록 하겠습니다.