

〈3주차 실습〉

SQL 실습

Database Programming



정규화 예제 실습

- $1NF \rightarrow 2NF$
- $2NF \rightarrow 3NF$
- $3NF \rightarrow BCNF$
- $BCNF \rightarrow 4NF$



실습 과제

1> 1NF → 2NF 변환

- 릴레이션(Relation)

- 사원의 프로젝트를 관리하는 릴레이션이다.

EMP_PROJ(Ssn, Pnumber, Hours, Pname, Plocation)

- 함수 종속(FD, Functional Dependency)

- FD1 : {Ssn, Pnumber} → Hours, Pname, Plocation
- FD2 : Pnumber → Pname, Plocation (부분 함수 종속)

- 변환 결과

- EMP_PROJ1 (Ssn, Pnumber, Hours)
- PROJECT (Pnumber, Pname, Plocation)

- ▶ 부분 함수 종속이 존재하지 않도록 릴레이션 EMP_PROJ을 릴레이션 EMP_PROJ1과 PROJECT로 분해

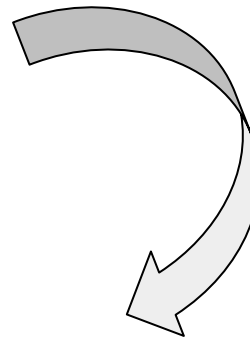
■ SQL 실습 – 변환 결과

• 1NF

```
SQL> SELECT * FROM EMP_PROJ;
```

SSN	PNUMBER	HOURS	PNAME	PLOCATION
123456789	2	7.5	ProductY	Sugarland
123456789	1	32.5	ProductX	Bellaire
833445555	10	10	Computerization	Bellaire
833445555	3	10	ProductZ	Bellaire
833445555	2	10	ProductY	Sugarland
453453453	2	20	ProductY	Sugarland
453453453	1	20	ProductX	Bellaire
666884444	3	40	ProductZ	Bellaire
888665555	20	10	Reorganization	Houston
987654321	30	20	NewBenefits	Stafford
987987987	30	5	NewBenefits	Stafford
999887777	30	30	NewBenefits	Stafford

12 개의 행이 선택되었습니다.



• 2NF

```
SQL> SELECT * FROM EMP_PROJ1;
```

SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
833445555	2	10
833445555	3	10
833445555	10	10
453453453	1	20
453453453	2	20
666884444	3	40
888665555	20	10
987654321	30	20
987987987	30	5
999887777	30	30

12 개의 행이 선택되었습니다.

```
SQL> SELECT * FROM PROJECT;
```

PNUMBER	PNAME	PLOCATION
1	ProductX	Bellaire
2	ProductY	Sugarland
3	ProductZ	Bellaire
10	Computerization	Bellaire
20	Reorganization	Houston
30	NewBenefits	Stafford

6 개의 행이 선택되었습니다.

- 참조 무결성 제약 조건 확인

```
SELECT Pnumber
FROM EMP_PROJ1
WHERE Pnumber NOT IN (SELECT EP1.Pnumber
                      FROM EMP_PROJ1 EP1, PROJECT P
                      WHERE EP1.Pnumber = P.Pnumber);
```

```
SQL> SELECT Pnumber
2    FROM EMP_PROJ1
3    WHERE Pnumber NOT IN (SELECT EP1.Pnumber
4                          FROM EMP_PROJ1 EP1, PROJECT P
5                          WHERE EP1.Pnumber = P.Pnumber);
```

선택된 레코드가 없습니다.

• 변환 전 후 비교

- 정규화 후(2NF)

```
SQL> SELECT Ssn, Pnumber, Hours, Pname, Plocation
2 FROM EMP_PROJ1 NATURAL JOIN PROJECT
3 ORDER BY Ssn asc, Pnumber desc;
```

SSN	PNUMBER	HOURS	PNAME	PLOCATION
123456789	2	7.5	ProductY	Sugarland
123456789	1	32.5	ProductX	Bellaire
333445555	10	10	Computerization	Bellaire
333445555	3	10	ProductZ	Bellaire
333445555	2	10	ProductY	Sugarland
453453453	2	20	ProductY	Sugarland
453453453	1	20	ProductX	Bellaire
666884444	3	40	ProductZ	Bellaire
888665555	20	10	Reorganization	Houston
987654321	30	20	NewBenefits	Stafford
987987987	30	5	NewBenefits	Stafford
999887777	30	30	NewBenefits	Stafford

12 개의 행이 선택되었습니다.

- 정규화 전(1NF)

```
SQL> SELECT * FROM EMP_PROJ;
```

SSN	PNUMBER	HOURS	PNAME	PLOCATION
123456789	2	7.5	ProductY	Sugarland
123456789	1	32.5	ProductX	Bellaire
333445555	10	10	Computerization	Bellaire
333445555	3	10	ProductZ	Bellaire
333445555	2	10	ProductY	Sugarland
453453453	2	20	ProductY	Sugarland
453453453	1	20	ProductX	Bellaire
666884444	3	40	ProductZ	Bellaire
888665555	20	10	Reorganization	Houston
987654321	30	20	NewBenefits	Stafford
987987987	30	5	NewBenefits	Stafford
999887777	30	30	NewBenefits	Stafford

12 개의 행이 선택되었습니다.

2> 2NF → 3NF 변환

- 릴레이션(Relation)

- 사원과 사원이 속한 부서의 정보를 관리하는 릴레이션이다. Dnumber는 후보키이다.

EMP_DEPT(Ssn, Ename, Bdate, Address, Dnumber, Dname, Dmgr_ssn)

- 함수 종속(FD, Functional Dependency)

- FD1 : $Ssn \rightarrow Ename, Bdate, Address, Dnumber, Dname, Dmgr_ssn$
- FD2 : $Dnumber \rightarrow Dname, Dmgr_ssn$ (이행 함수 종속)

- 변환 결과

- **EMP_DEPT1 (Ssn, Ename, Bdate, Address, Dnumber)**
- **DEPT (Dnumber, Dname, Dmgr_ssn)**

- ▶ 이행 함수 종속이 존재하지 않도록 릴레이션 EMP_DEPT을 릴레이션 EMP_DEPT1과 DEPT로 분해

■ SQL 실습 – 변환 결과

• 2NF

```
SQL> select * from emp_dept;
```

SSN	ENAME	BDATE	ADDRESS	DNUMBER	DNAME	DMGR_SSN
123456789	John	65/01/09	731 Fondren, Houston, TX	5	Research	333445555
333445555	Franklin	55/12/08	683 Voccs, Houston, TX	5	Research	333445555
453453453	Joyce	72/07/31	5631 Rice, Houston, TX	5	Research	333445555
666884444	Ramesh	62/09/15	975 Fire Oak, Humble, TX	5	Research	333445555
888665555	James	37/11/10	450 Stone, Houston, TX	1	Headquarters	888665555
987654321	Jennifer	41/06/20	291 Berry, Bellaire, TX	4	Administration	987654321
987987987	Ahmad	69/03/29	980 Dallas, Houston, TX	4	Administration	987654321
999887777	Alicia	68/01/19	3321 Castle, Spring, TX	4	Administration	987654321

8 개의 행이 선택되었습니다.

• 3NF

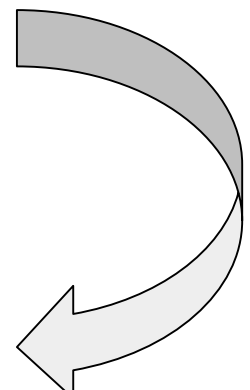
```
SQL> SELECT * FROM EMP_DEPT1;
```

SSN	ENAME	BDATE	ADDRESS	DNUMBER
123456789	John	65/01/09	731 Fondren, Houston, TX	5
333445555	Franklin	55/12/08	683 Voccs, Houston, TX	5
453453453	Joyce	72/07/31	5631 Rice, Houston, TX	5
666884444	Ramesh	62/09/15	975 Fire Oak, Humble, TX	5
888665555	James	37/11/10	450 Stone, Houston, TX	1
987654321	Jennifer	41/06/20	291 Berry, Bellaire, TX	4
987987987	Ahmad	69/03/29	980 Dallas, Houston, TX	4
999887777	Alicia	68/01/19	3321 Castle, Spring, TX	4

8 행이 선택되었습니다.

```
SQL> SELECT * FROM DEPT;
```

DNUMBER	DNAME	DMGR_SSN
1	Headquarters	888665555
4	Administration	987654321
5	Research	333445555



- 참조 무결성 제약 조건 확인

```
SELECT Dnumber
FROM EMP_DEPT1
WHERE Dnumber NOT IN (SELECT ED1.Dnumber
                      FROM EMP_DEPT1 ED1, DEPT D
                      WHERE ED1.Dnumber = D.Dnumber);
```

```
SQL> SELECT Dnumber
2  FROM EMP_DEPT1
3  WHERE Dnumber NOT IN (SELECT ED1.Dnumber
4                        FROM EMP_DEPT1 ED1, DEPT D
5                        WHERE ED1.Dnumber = D.Dnumber);
```

선택된 레코드가 없습니다.

• 변환 전 후 비교

- 정규화 후(3NF)

```
SQL> SELECT Ssn, Ename, Bdate, Address, Dnumber, Dname, Dmgr_ssn
2 FROM EMP_DEPT1 NATURAL JOIN DEPT
3 ORDER BY Ssn;
```

SSN	ENAME	BDATE	ADDRESS	DNUMBER	DNAME	DMGR_SSN
123456789	John	65/01/09	731 Fondren, Houston, TX	5	Research	333445555
333445555	Franklin	55/12/08	683 Vocs, Houston, TX	5	Research	333445555
453453453	Joyce	72/07/31	5631 Rice, Houston, TX	5	Research	333445555
666884444	Ramesh	62/09/15	975 Fire Oak, Humble, TX	5	Research	333445555
888665555	James	37/11/10	450 Stone, Houston, TX	1	Headquarters	888665555
987654321	Jennifer	41/06/20	291 Berry, Bellaire, TX	4	Administration	987654321
987987987	Ahmad	69/03/29	980 Dallas, Houston, TX	4	Administration	987654321
999887777	Alicia	68/01/19	3321 Castle, Spring, TX	4	Administration	987654321

8 개의 행이 선택되었습니다.

- 정규화 전(2NF)

```
SQL> select * from emp_dept;
```

SSN	ENAME	BDATE	ADDRESS	DNUMBER	DNAME	DMGR_SSN
123456789	John	65/01/09	731 Fondren, Houston, TX	5	Research	333445555
333445555	Franklin	55/12/08	683 Vocs, Houston, TX	5	Research	333445555
453453453	Joyce	72/07/31	5631 Rice, Houston, TX	5	Research	333445555
666884444	Ramesh	62/09/15	975 Fire Oak, Humble, TX	5	Research	333445555
888665555	James	37/11/10	450 Stone, Houston, TX	1	Headquarters	888665555
987654321	Jennifer	41/06/20	291 Berry, Bellaire, TX	4	Administration	987654321
987987987	Ahmad	69/03/29	980 Dallas, Houston, TX	4	Administration	987654321
999887777	Alicia	68/01/19	3321 Castle, Spring, TX	4	Administration	987654321

8 개의 행이 선택되었습니다.

3> 3NF → BCNF 변환

- 릴레이션(Relation)

- 학생의 수강 정보를 관리하는 릴레이션이다. 학번, 과목이 강사를 결정하고, 강사는 하나의 과목만 담당한다.

TEACH(Student, Course, Instructor)

- 함수 종속(FD, Functional Dependency)

- FD1 : {Student, Course} → Instructor
 - FD2 : Instructor → Course (결정자가 슈퍼키(후보키)가 아님)

- 변환 결과

- TEACH1 (Student, Instructor)
 - TEACH2 (Instructor, Course)

- ▶ 후보키가 아닌 결정자가 존재하는 릴레이션 TEACH에서 릴레이션 TEACH1과 릴레이션 TEACH2로 분해
- ▶ 릴레이션 TEACH1에서 Student와 Instructor를 복합키로 설정
- ▶ 릴레이션 TEACH2에서 Instructor를 기본키로 설정

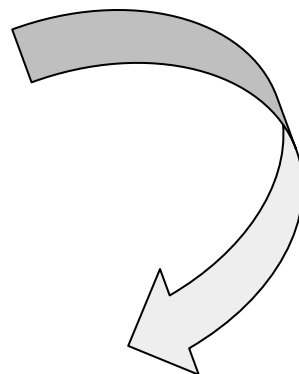
■ SQL 실습 – 변환 결과

• 3NF

```
SQL> SELECT * FROM TEACH;
```

STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omicinski
Zelaya	Database	Navathe
Narayan	Operating Systems	Ammar

9 개의 행이 선택되었습니다.



• BCNF

```
SQL> SELECT * FROM TEACH1;
```

STUDENT	INSTRUCTOR
Narayan	Ammar
Narayan	Mark
Smith	Ammar
Smith	Navathe
Smith	Schulman
Wallace	Ammar
Wallace	Mark
Wong	Omicinski
Zelaya	Navathe

9 개의 행이 선택되었습니다.

```
SQL> SELECT * FROM TEACH2;
```

INSTRUCTOR	COURSE
Ahamad	Operating System
Ammar	Operating System
Mark	Database
Navathe	Database
Omicinski	Database
Schulman	Theory

6 개의 행이 선택되었습니다.

- 참조 무결성 제약 조건 확인

```
SELECT Instructor
FROM TEACH1
WHERE Instructor NOT IN (SELECT T1.Instructor
                        FROM TEACH1 T1, TEACH2 T2
                        WHERE T1.Instructor = T2.Instructor);
```

```
SQL> SELECT Instructor
2  FROM TEACH1
3  WHERE Instructor NOT IN (SELECT T1.Instructor
4                          FROM TEACH1 T1, TEACH2 T2
5                          WHERE T1.Instructor = T2.Instructor);
```

선택된 레코드가 없습니다.

- 변환 전 후 비교

- 정규화 후(BCNF)

```
SQL> SELECT Student, Course, Instructor
2 FROM TEACH1 NATURAL JOIN TEACH2;
```

STUDENT	COURSE	INSTRUCTOR
Narayan	Operating System	Ammar
Narayan	Database	Mark
Smith	Operating System	Ammar
Smith	Database	Navathe
Smith	Theory	Schulman
Wallace	Operating System	Ammar
Wallace	Database	Mark
Wong	Database	Omicinski
Zelaya	Database	Navathe

9 개의 행이 선택되었습니다.

- 정규화 전(3NF)

```
SQL> SELECT * FROM TEACH;
```

STUDENT	COURSE	INSTRUCTOR
Narayan	Operating Systems	Ammar
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omicinski
Zelaya	Database	Navathe

9 개의 행이 선택되었습니다.

4> BCNF → 4NF 변환

- 릴레이션(Relation)

- 사원의 프로젝트 정보와 부양가족 정보를 관리하는 릴레이션이다. 사원은 여러 개의 프로젝트를 참여할 수 있고, 여러 명의 부양가족을 등록할 수 있다.

EMP(Ename, Pname, Dname)

- 다치 종속(MVD, Multivalued Dependency)

- MVD1 : Ename →→ Pname (다치 종속)
- MVD2 : Ename →→ Dname (다치 종속)

- 변환 결과

- EMP_PROJECTS (Ename, Pname)
- EMP_DEPENDENTS (Ename, Dname)

- ▶ 다치 종속이 존재하는 릴레이션 EMP를 릴레이션 EMP_PROJECTS와 릴레이션 EMP_DEPENDENTS로 분해

■ SQL 실습 – 변환 결과

• BCNF

```
SQL> SELECT * FROM EMP;
```

ENAME	PNAME	DNAME
Smith	X	Anna
Smith	X	John
Smith	Y	Anna
Smith	Y	John

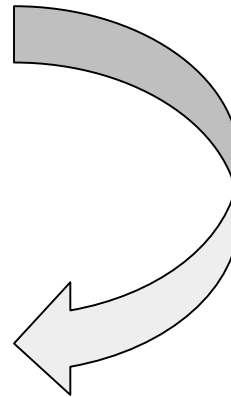
• 4NF

```
SQL> SELECT * FROM EMP_PROJECTS;
```

ENAME	PNAME
Smith	X
Smith	Y

```
SQL> SELECT * FROM EMP_DEPENDENTS;
```

ENAME	DNAME
Smith	Anna
Smith	Jonh



- 참조 무결성 제약 조건 확인

```
SELECT Ename
FROM EMP_PROJECTS
WHERE Ename NOT IN (SELECT EP.Ename
                    FROM EMP_PROJECTS EP, EMP_DEPENDENTS ED
                    WHERE EP.Ename = ED.Ename);
```

```
SQL> SELECT Ename
2  FROM EMP_PROJECTS
3  WHERE Ename NOT IN (SELECT EP.Ename
4                      FROM EMP_PROJECTS EP, EMP_DEPENDENTS ED
5                      WHERE EP.Ename = ED.Ename);
```

선택된 레코드가 없습니다.

- 변환 전 후 비교

- 정규화 후(4NF)

```
SQL> SELECT Ename, Pname, Dname  
2 FROM EMP_PROJECTS NATURAL JOIN EMP_DEPENDENTS  
3 ORDER BY PNAME;
```

ENAME	PNAME	DNAME
Smith	X	Anna
Smith	X	Jonh
Smith	Y	Jonh
Smith	Y	Anna

- 정규화 전(BCNF)

```
SQL> SELECT * FROM EMP;
```

ENAME	PNAME	DNAME
Smith	X	Anna
Smith	X	John
Smith	Y	Anna
Smith	Y	John

- 제공한 자료를 사용하여 다음의 각 문제를 해결한 결과를 제출하시오.
 - 각 문제에서 제시한 설명, 릴레이션 스키마, 종속성을 참고하여 Oracle에서 변환 결과, 참조 무결성 확인, 변환 전 후를 비교하고, 이에 대한 타당성을 주석 처리하여 작성할 것

Q1> 다음은 1NF를 만족하는 릴레이션과 함수 종속이다. 이를 2NF로 변환하라.

- 학생의 성적을 관리하는 릴레이션이다.

STUDENT_GRADE(Sid, Cid, Grade, Cname, Cnit)

FD1 : {Sid, Cid} → Grade, Cname, Cunit

FD2 : Cid → Cname, Cunit

Q2> 다음은 2NF를 만족하는 릴레이션과 함수 종속이다. 3NF로 변환하라.

- 도서 정보를 관리하는 릴레이션이다.
- Book_type은 후보키이며, 한 구역에 여러 권의 도서를 배치할 수 있다.

BOOK(Book_title, Authurname, Book_type, Location, Publisher)

FD1 : Book_title → Authurname, Book_type, Location, Publisher

FD2 : Book_type → Location

Q3> 다음은 3NF를 만족하는 릴레이션과 함수 종속이다. BCNF로 변환하라.

- 프로젝트의 미팅 정보를 관리하는 릴레이션이다.
- 프로젝트와 미팅 일자가 사원을 결정하고, 사원은 하나의 프로젝트만 참여한다.

STAFF_MEETING(ProjectName, MeetingDate, EmployeeName)

FD1 : {ProjectName, MeetingDate} \rightarrow EmployeeName

FD2 : EmployeeName \rightarrow ProjectName

Q4> 다음은 BCNF를 만족하는 릴레이션과 다치 종속이다. 4NF로 변환하라.

- 회원의 정보를 관리하는 릴레이션이다.
- 회원은 여러 명의 형제자매를 등록할 수 있고, 여러 개의 취미를 등록할 수 있다.

MEMBER(Name, Sibling, Hobby)

MVD1 : Name \twoheadrightarrow Sibling

MVD2 : Name \twoheadrightarrow Hobby

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : spool file(Oracle에서 실행한 로그 파일)
- 제출 형식 : 학번_이름_주차
 - Ex) 학번_홍길동_3주차.zip
- 제출 기한 : 수업 시작 시간으로 부터 24시간 이내 제출
 - 제출 기한 위반 시 감점 기준
 - 지각 제출 시 과제 점수에서 40% 감점
 - 1일 초과 당 10% 추가 감점 (단, 7일 이후 제출 불가)