SQL for PostgreSQL

황종필

정규화란?

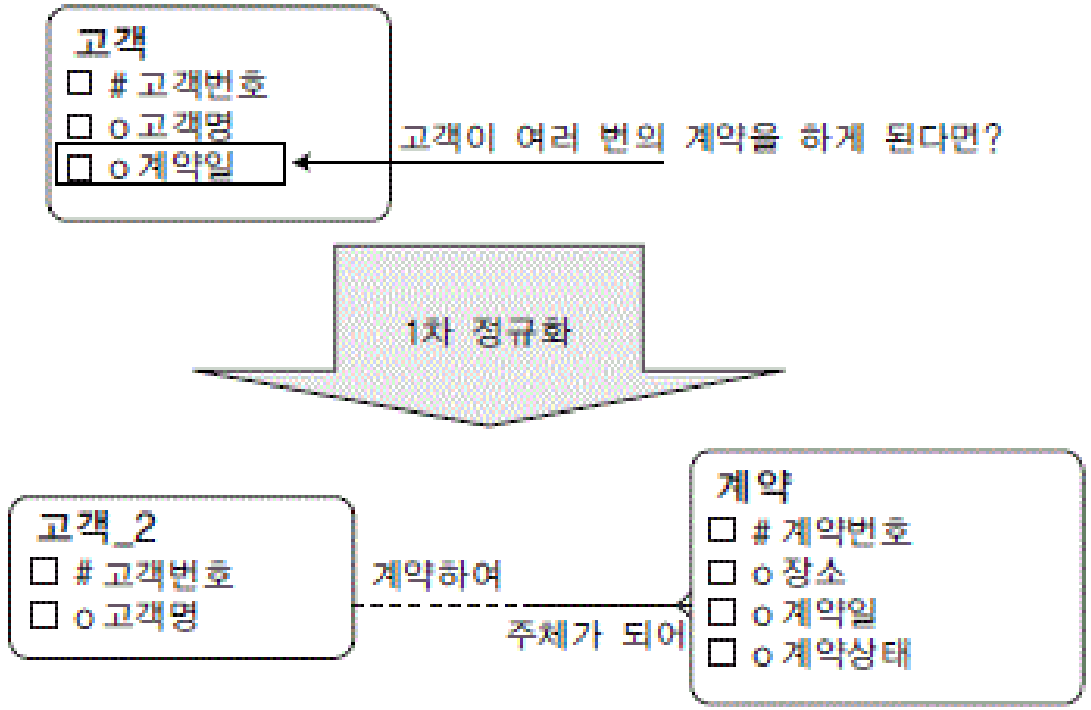
- 제 1 정규형(1NF : First Normal Form)
  - 모든 속성은 반드시 하나의 값을 가져야 한다.
    (반복형태가 있어서는 안됨)
- 제 2 정규형(2NF : Second Normal Form)
  - 모든 속성은 반드시 유일식별자(UID : Unique Identifier) 전부에 종속되어야 한다.(UID 일부에만 종속되어서는 안됨)
- 제 3 정규형(3NF : Third Normal Form)
  - 유일식별자가 아닌 모든 속성 간에는 서로 종속될 수 없다.
    (속성간 종속성 배제)
- 간단히 말하면 속성을 있어야 할 엔터티에 위치시키는 것.

정규화의 장점

- 중복값이 줄어든다.
- NULL 값이 줄어든다.
- 복잡한 코드로 데이터 모델을 보완할 필요가 없다.(무결성)
- 새로운 요구 사항의 발견 과정을 돕니다.
- 업무 규칙의 정밀한 포착을 보증한다.
- 데이터 구조의 안정성을 최대화한다.

제 1 정규형

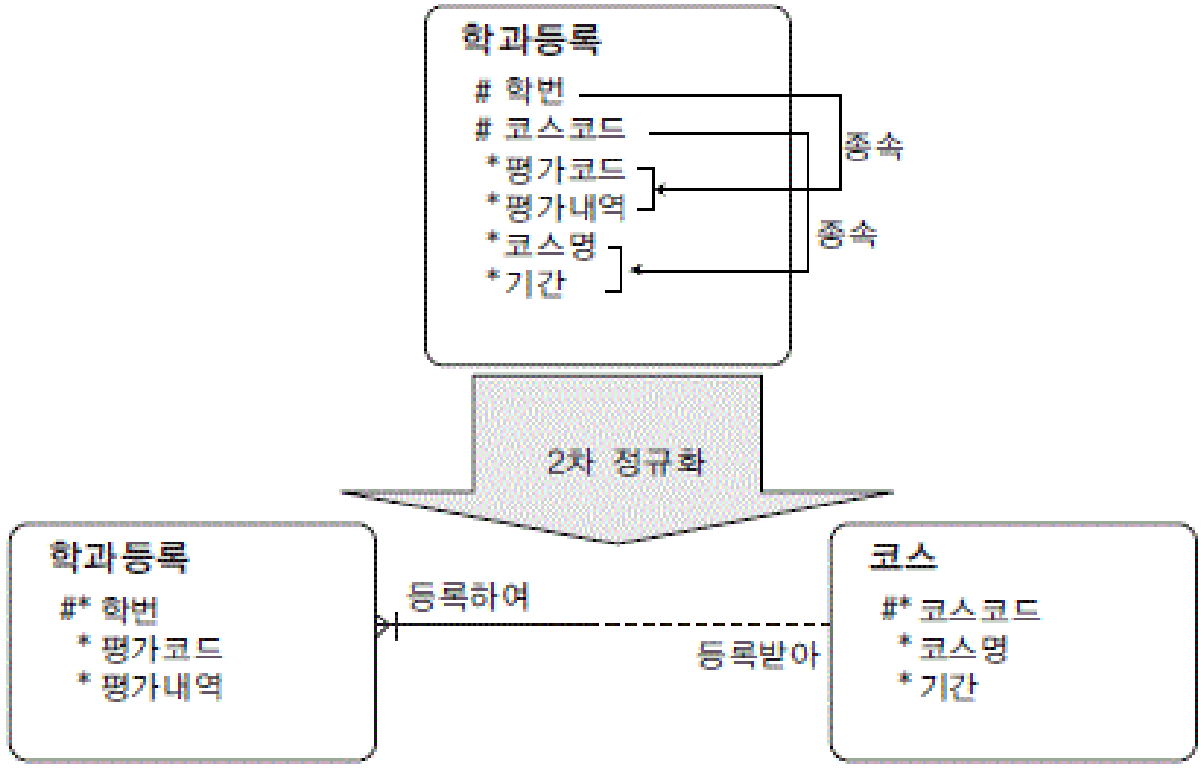
모든 속성은 반드시 하나의 값을 가져야 한다.
(반복형태가 있어서는 안됨)



출처 : www.dbguide.net

제 2 정규형

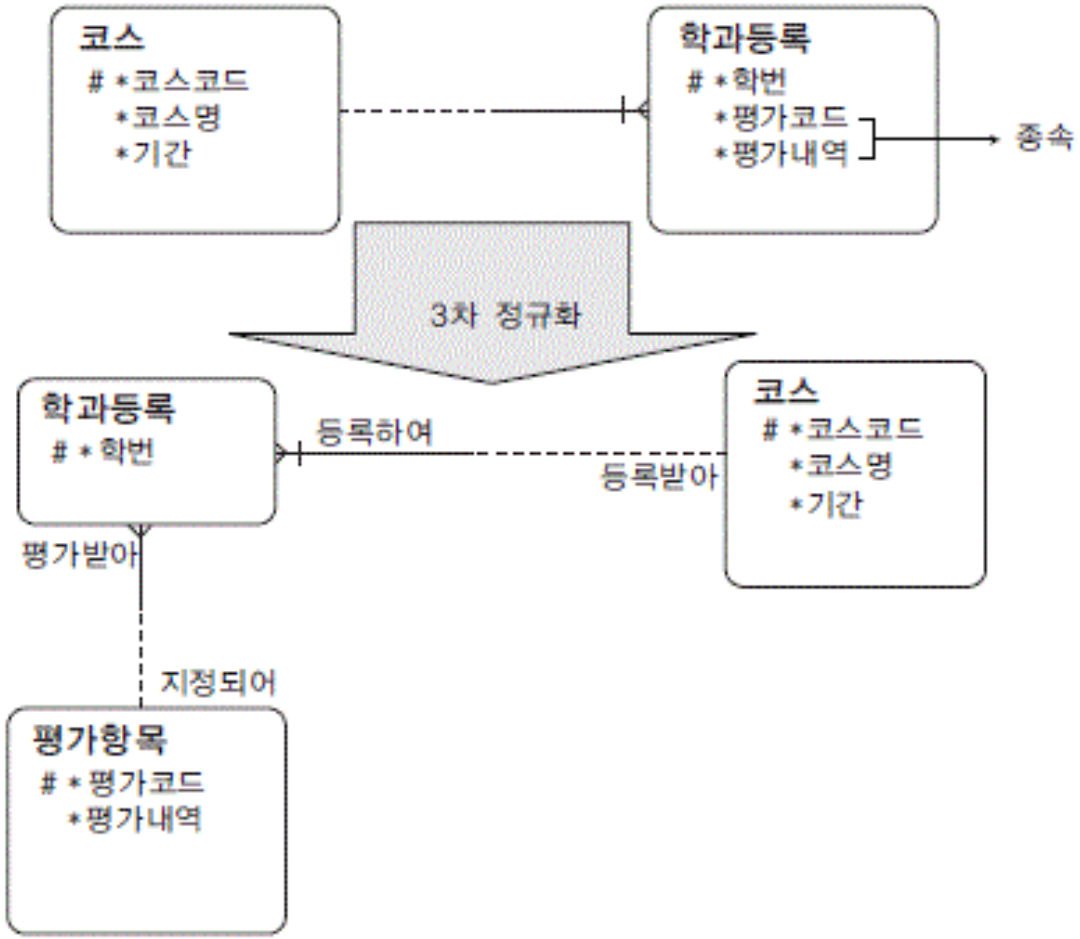
모든 속성은 반드시 유일식별자(UID : Unique Identifier) 전부에 종속되어야 한다.(UID 일부에만 종속되어서는 안됨)



출처 : www.dbguide.net

제 3 정규형

유일식별자가 아닌 모든 속성 간에는 서로 종속될 수 없다.
(속성간 종속성 배제)



출처 : www.dbguide.net

정규형 예시 - 1

- 몇 정규형 위반?
- 정규화를 하면?

자격증 보유 사항(PK : 회원번호)

회원번호	이름	주소	핸드폰번호	주민번호	자격증	취득일
1	홍길동	서울시 구로구	(111) 111 -1111	111111-1111111	운전면허1종	19980203
1	홍길동	서울시 구로구	(111) 111 -1111	111111-1111111	MCSE	19990630
1	홍길동	서울시 구로구	(111) 111 -1111	111111-1111111	정보처리기사	20111113

정규형 예시 - 1 (제 1 정규형 위반)

- 위의 테이블은 회원이 보유한 자격증 정보를 관리하기 위한 테이블이며 중복되는 정보가 저장되어 있음.
- 홍길동이 갖고 있는 자격증 정보를 관리하기 위해 홍길동에 대한 정보, 즉 회원번호, 이름, 주소 핸드폰번호, 주민번호 등의 정보가 중복해서 저장되고 있으므로 쓸데 없는 공간의 낭비가 있음.
- 만약 홍길동의 주소나 핸드폰 번호가 바뀌게 되면 3개의 레코드 모두를 변경해야 함.
- 이를 해결하기 위해 반복 되어지는 그룹 속성(=같은 성격과 내용의 컬럼이 연속적으로 나타나는 컬럼)을 제거한 뒤 새로운 테이블을 생성하고 기존의 테이블과 1:N의 관계를 형성해야 함.

회원(PK : 회원번호)

회원번호	이름	주소	핸드폰번호	주민번호
1	홍길동	서울시 구로구	(111) 111 -1111	111111-1111111

자격증 보유(PK : 회원번호, FK : [회원].회원번호)

자격증번호	회원번호	자격증	취득일
1	1	운전면허1종	19980203
2	1	MCSE	19990630
3	1	정보처리기사	20111113

정규형 예시 - 2

- 몇 정규형 위반?
- 정규화를 하면?

과정(PK : 과정코드)

과정코드	과정명	교육기간	강의시간	수강료	교재1	교재2	교재3	교재4	교재5
A0001	웹프로그래밍	5개월	8시간	1800000	SQL Server	HTML 기초	ASP활용	JSP활용	실무구축
A0002	웹마스터	4개월	4시간	700000	Windows 2000	SQL Server			
A0003	ASP프로그래밍	3개월	8시간	1400000	HTML 활용	ASP기초			
J0001	JAVA프로그래밍	9개월	8시간	2000000	SQL Server	JAVA기초	JSP		

정규형 예시 - 2 (제 1 정규형 위반)

- 위의 테이블은 교재 컬럼의 계수는 가장 많은 교재를 사용하는 과정을 기준으로 생성되어 있음.
- 교재가 5권인 웹 프로그램밍의 컬럼들은 모두 채워지지만 그 외 과목은 최소 1개 이상의 Null 값이 들어가게 됨.
- 또 교재가 늘어 나게 된다면 컬럼이 계속 추가해야 한다.
- 해결 방법은 교재 컬럼처럼 반복되는 속성을 갖는 컬럼의 경우에는 테이블에 Null이 입력되는 수가 있으므로 이러한 경우의 수를 제거하기 위해 테이블을 타라 분리해서 기존의 테이블과 1:N의 관계로 형성해야 함.

과정(PK : 과정코드)

과정코드	과정명	교육기간	강의시간	수강료
A0001	웹프로그래밍	5개월	8시간	1800000
A0002	웹마스터	4개월	4시간	700000
A0003	ASP프로그래밍	3개월	8시간	1400000
J0001	JAVA프로그래밍	9개월	8시간	2000000

교재(PK : [과정].과정코드+교재순서, FK : [과정].과정코드)

과정코드	교재순서	교재
A0001	1	SQL Server
A0001	2	HTML 기초
A0001	3	ASP활용
A0001	4	JSP활용
A0001	5	실무구축
A0002	1	Windows 2000
A0002	2	SQL Server
A0003	1	HTML 활용
A0003	2	ASP기초
J0001	1	SQL Server
J0001	3	JAVA기초
J0001	4	JSP

하지만 답안도 틀림

정규형 예시 - 2 (제 3 정규형 위반)

과정(PK : 과정코드)

과정코드	과정명	교육기간	강의시간	수강료
A0001	웹프로그래밍	5개월	8시간	1800000
A0002	웹마스터	4개월	4시간	700000
A0003	ASP프로그래밍	3개월	8시간	1400000
J0001	JAVA프로그래밍	9개월	8시간	2000000

과정별교재(PK : 과정코드+교재순서, FK : [과정].과정코드, FK : [교재].교재번호)

과정코드	교재순서	교재번호
A0001	1	1
A0001	2	2
A0001	3	3
A0001	4	4
A0001	5	5
A0002	1	6
A0002	2	1
A0003	1	7
A0003	2	8
J0001	1	1
J0001	3	9
J0001	4	10

교재(PK : 교재번호)

교재번호	교재
1	SQL Server
2	HTML 기초
3	ASP활용
4	JSP활용
5	실무구축
6	Windows 2000
7	HTML 활용
8	ASP기초
9	JAVA기초
10	JSP

정규형 예시 - 3

- 몇 정규형 위반?
- 정규화를 하면?

교육평가 (PK : 학번 + 과정코드)

학번	과정코드	평가	과정명	기간
100	A01	A	JAVA프로그래밍	4개월
101	B01	D+	웹마스터	6개월
101	B03	A	DBMS전문가	5개월
100	B01	B	웹마스터	6개월
102	A01	B	JAVA프로그래밍	4개월
102	B03	A	DBMS전문가	5개월

정규형 예시 - 3 (제 2 정규형 위반)

- 평가는 복합키인 학번 + 과정코드에 대해서 의존적이므로 이를 “완전 함수의 종속”이라고 한다.
- 과정명, 기간은 복합키의 일부분인 과정코드에 의해서 의존적이므로 “부분 함수 종속”이라고 한다.
- “JAVA프로그래밍” 과정명이 “java”로 수정 되었을 경우 그에 해당하는 모든 데이터를 수정해야 한다.
- 교육 평가 테이블에서 과정명, 기간에 중복되는 데이터가 발생 → 데이터 무결성 유지 어려움.
- 부분 함수 종속성을 일으키는 요인을 제거(테이블에서 분리하여 키가 아닌 모든 속성들이 기본키에 완전 함수 종속이 되도록 함.

교육평가 (PK : 학번 + 과정코드, FK : [교육과목].과정코드)

학번	과정코드	평가
100	A01	A
101	B01	D+
101	B03	A
100	B01	B
102	A01	B
102	B03	A

교육과목 (PK : 과정코드)

과정코드	과정명	기간
A01	JAVA프로그래밍	4개월
B01	웹마스터	6개월
B03	DBMS전문가	5개월

정규형 예시 - 4

- 몇 정규형 위반?
- 정규화를 하면?

과목수강(PK : 학번)

학번	과목	수강료
100	JAVA	70000
101	C	50000
102	ASP	60000
103	JAVA	70000
104	ASP	60000

정규형 예시 - 4 (제 3 정규형 위반)

- 과목 수강 테이블에 C++ 수강료가 60000원이라는 내용을 삽입하려고 할 때 이 과목을 수강하는 학생이 존재하지 않더라도 수강신청 한 것처럼 해야 삽입이 가능함.
- 101번 학생이 수강을 취소하면 C과목이 사라짐
- JAVA의 수강료를 80000원으로 변경하려고 할 경우 데이터의 불일치가 발생할 수 있음.
- 특정 학생이 수강 할 과목을 선정하게 되면 그 과목에 의하여 수강료가 결정됨. 키가 아닌 속성인 과목에 의해서 수강료가 결정되게 되며 학번으로 수강료를 검색할 수 있음. 이를 이행적 함수 종속성이라함.
- 함수 종속성을 일으키는 요인을 제거(테이블 분리)

과목수강(PK : 학번, FK : [과목].과목)

학번	과목
100	JAVA
101	C
102	ASP
103	JAVA
104	ASP

과목(PK : 과목)

과목	수강료
JAVA	70000
C	50000
ASP	60000

완전한 상태가 아님

정규형 예시 - 4 (제 3 정규형 위반)

과목수강(PK : 학번, FK : [과목].과목코드)

학번	과목코드
100	1
101	2
102	3
103	1
104	3

과목(PK : 과목코드)

과목코드	과목	수강료
1	JAVA	70000
2	C	50000
3	ASP	60000

Table Re-create

- Customers
- Orders

Table Script

Windows Functions

- WINDOW_FUNCTION([arguments])
  OVER (PARTITION BY … [ORDER BY …])

1. 그룹 내 순서 획득
   - RANK, DENSE_RANK, ROW_NUMBER
2. 그룹 내 특정위치 컬럼 획득
   - FIRST_VALUE, LAST_VALUE, LAG, LEAD, NTH_VALUE
3. 그룹 내 집계정보 획득
   - SUM, MAX, MIN, AVG, COUNT
4. 기타
   - CUME_DIST, NTILE

SQL - 1

- 회원별로 주문의 순서를 알고 싶다.(중복 배제)

SQL - 1

- 회원별로 주문의 순서를 알고 싶다.(중복 배제)

```
select ord.userid
     , ord.order_date
     , row_number() over (partition by ord.userid order by order_date) as rk
  from orders ord
```

userid	order_date	rk
1	2015-08-01	1
1	2015-08-03	2
1	2015-08-10	3
1	2015-08-14	4
1	2015-08-25	5
2	2015-08-03	1
2	2015-08-11	2
2	2015-08-12	3
2	2015-08-22	4
2	2015-08-28	5
3	2015-08-07	1
3	2015-08-19	2
3	2015-08-30	3
4	2015-08-05	1
4	2015-08-18	2
5	2015-08-15	1
5	2015-08-17	2
5	2015-08-21	3
5	2015-08-23	4
5	2015-08-29	5

SQL - 2

- 주문의 방법의 TOUCH 인 주문들 중 가장 먼저 주문한 주문을 알고 싶다.(중복 허용)

SQL - 2

- 주문의 방법의 TOUCH 인 주문들 중 가장 먼저 주문한 주문을 알고 싶다.(중복 허용)

```
select ord.userid
     , ord.order_date
     , ord.method
     , ord.rk
  from (select ord.userid
             , ord.order_date
             , ord.method
             , dense_rank() over (order by order_date) as rk -- rank 가능
          from orders ord
         where ord.method = 'TOUCH') ord
 where ord.rk = 1
```

userid	order_date	method	rk
1	2015-08-03	TOUCH	1
2	2015-08-03	TOUCH	1

SQL - 3

- 주문의 방법이 TOUCH인 주문들 중 TOP 3의 주문 정보를 알고 싶다.(중복 허용, 무조건 1,2,3 포함)

SQL - 3

- 주문의 방법이 TOUCH인 주문들 중 TOP 3의 주문 정보를 알고 싶다.(중복 허용, 무조건 1,2,3 포함)

```
select ord.userid
     , ord.order_date
     , ord.method
     , ord.rk
  from (select ord.userid
             , ord.order_date
             , ord.method
             , dense_rank() over (order by order_date) as rk
          from orders ord
         where ord.method = 'TOUCH') ord
 where ord.rk between 1 and 3
```

userid	order_date	method	rk
1	2015-08-03	TOUCH	1
2	2015-08-03	TOUCH	1
1	2015-08-10	TOUCH	2
2	2015-08-11	TOUCH	3

SQL - 4

- 회원 1과 3의 각 주문 정보를 회원의 가장 빠른 주문 일자와 함께 보고 싶다.

SQL - 4

- 회원 1과 3의 각 주문 정보를 회원의 가장 빠른 주문 일자와 함께 보고 싶다.

```
select ord.userid
     , first_value(order_date) over (partition by ord.userid order by order_date) as first_order_date
     , ord.order_date
     , ord.method
  from orders ord
 where ord.userid in (1, 3)
```

userid	first_order_date	order_date	method
1	2015-08-01	2015-08-01	CALL
1	2015-08-01	2015-08-03	TOUCH
1	2015-08-01	2015-08-10	TOUCH
1	2015-08-01	2015-08-14	CALL
1	2015-08-01	2015-08-25	TOUCH
3	2015-08-07	2015-08-07	CALL
3	2015-08-07	2015-08-19	TOUCH
3	2015-08-07	2015-08-30	CALL

SQL - 5

- 회원 4과 5의 각 주문의 주문 금액과 회원의 전체 주문 금액에서의 비중을 보고 싶다.

SQL - 5

- 회원 4과 5의 각 주문의 주문 금액과 회원의 전체 주문 금액에서의 비중을 보고 싶다.

```
select ord.userid
     , ord.order_date
     , ord.amount
     , sum(ord.amount) over (partition by ord.userid)              as total_amount
     , round(ord.amount::numeric / sum(ord.amount) over (partition by ord.userid) * 100, 2) as
amount_rate
  from orders ord
 where ord.userid in (4, 5)
```

userid	order_date	amount	total_amount	amount_rate
4	2015-08-05	20000	50000	40
4	2015-08-18	30000	50000	60
5	2015-08-15	10000	75000	13.33
5	2015-08-17	20000	75000	26.67
5	2015-08-21	5000	75000	6.67
5	2015-08-23	10000	75000	13.33
5	2015-08-29	30000	75000	40