

제05장

# 정규화

MySQL

# 학습목표

1. 이상 현상에 대해서 이해할 수 있다.
2. 함수적 종속성에 대해서 이해할 수 있다.
3. 정규화에 대해서 이해할 수 있다.

```
each: function(e, t, n) {  
  var r, i = 0,  
      o = e.length,  
      a = M(e);  
  if (n) {  
    if (a) {  
      for (; o > i; i++)  
        if (r = t.apply(e[i], n), r ===  
    } else  
      for (i in e)  
        if (r = t.apply(e[i], n), r ===  
  } else if (a) {  
    for (; o > i; i++)  
      if (r = t.call(e[i], i, e[i])  
  } else  
    for (i in e)  
      if (r = t.call(e[i], i, e[i])  
  return e  
},  
trim: b && !b.call("\uffff\u00a0") ?  
  return null == e ? "" : b.call(  
} : function(e) {  
  return null == e ? "" : (e + "  
},  
makeArray: function(e, t) {  
  var n = t || [];  
  return null != e && (M(Obj  
},  
isArray: function(e, t, n) {  
  var r;  
  if (t) {  
    if (n) return n.c  
    for (n = t.length;  
      if (n in t  
  }  
}
```

# 목차

1. 이상 현상
2. 함수적 종속성
3. 정규화

```

each: function(e, t, n) {
    var r, i = 0,
        o = e.length,
        a = M(e);
    if (n) {
        if (a) {
            for (; o > i; i++)
                if (r = t.apply(e[i], n), r === !1) break;
        } else
            for (i in e)
                if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], e[i]), r === !1) break;
    } else
        for (i in e)
            if (r = t.call(e[i], e[i]), r === !1) break;
    return e;
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e);
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string"
    ),
inArray: function(e, t, n) {
    var r;
    if (t) {
        if (m) return m.call(t, e, n);
        for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n) : n; n < r; n++)
            if (n in t && t[n] === e) return n;
    }
}

```

# 1. 이상 현상

# 이상 현상

## ■ 이상현상 (Anomaly)

- 잘못된 데이터베이스 설계의 결과물
- 불필요한 데이터의 중복으로 인한 공간 낭비와 부작용 초래
- 실제 데이터와 데이터베이스에 저장된 데이터가 일치하지 않는 문제 발생

## ■ 이상현상 종류

- 삽입 이상 (Insertion Anomaly)
- 갱신 이상 (Update Anomaly)
- 삭제 이상 (Deletion Anomaly)

# 삽입 이상

<u>학번 (PK)</u>	성명	학과코드	학과	학과장
10101	이민지	111	기계공학	박철수
10102	공영숙	111	기계공학	박철수
10103	최지수	111	기계공학	박철수
10104	박영희	222	호텔경영	김진주
10105	김가을	333	섬유공학	이숙희
		555	컴퓨터공학	최진영

## 문제

새로운 학과를 삽입하려고 하였으나  
학번 없이는 삽입할 수 없는 문제가 발생함  
(학번은 PK이므로 NOT NULL임)

## 삽입이상

데이터 삽입을 위해서  
불필요한 데이터를 추가로  
삽입해야 함

# 갱신 이상

학번 (PK)	성명	학과코드	학과	학과장
10101	이민지	111	기계공학	김영환
10102	공영숙	111	기계공학	김영환
10103	최지수	111	기계공학	박철수
10104	박영희	222	호텔경영	김진주
10105	김가을	333	섬유공학	이숙희
10106	김선우	555	컴퓨터공학	최진영

## 문제

기계공학의 학과장을 '박철수'에서 '김영환'으로 수정하다가 일부 데이터가 수정되지 않아 데이터의 신뢰성이 훼손

## 갱신이상

데이터의 일부만 갱신되면서 동일한 데이터가 다른 값을 가지게 됨

# 삭제 이상

학번 (PK)	성명	학과코드	학과	학과장
10101	이민지	111	기계공학	박철수
10102	공영숙	111	기계공학	박철수
10103	최지수	111	기계공학	박철수
10104	박영희	222	호텔경영	김진주
10105	김가을	333	섬유공학	이숙희
10106	김선우	555	컴퓨터공학	최진영

## 문제

'김선우' 학생 데이터를 삭제하면  
'컴퓨터공학' 학과 정보도 함께 삭제됨

## 삭제이상

특정 데이터 삭제로 인해서  
의도하지 않은 다른 데이터가  
함께 삭제됨



```

each: function(e, t, n) {
    var r, i = 0,
        o = e.length,
        a = M(e);
    if (n) {
        if (a) {
            for (; o > i; i++)
                if (r = t.apply(e[i], n), r === !1) break;
        } else
            for (i in e)
                if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], e[i]), r === !1) break;
    } else
        for (i in e)
            if (r = t.call(e[i], e[i]), r === !1) break;
    return e;
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e);
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string"
),
isArray: function(e, t, n) {
    var r;
    if (t) {
        if (n) return m.call(t, e, n);
        for (r = t.length, r = r ? 0 > n ? Math.max(0, r + n) : n; n in t && t[n] === e) return n;
    }
}

```

## 2. 함수적 종속성

# 함수적 종속성

## ■ 함수적 종속성

- Functional Dependency
- X가 Y를 함수적으로 결정하는 경우 X를 결정자, Y를 종속자라고 함
- 표현 방법 :  $X \rightarrow Y$
- 이런 경우 "Y가 X에 함수적으로 종속되어 있다." 라고 함
- X를 알면 Y가 무엇인지 바로 알 수 있는 경우를 의미함

## ■ 종류

- 부분 함수적 종속성 (Partial Functional Dependency)
- 완전 함수적 종속성 (Full Functional Dependency)
- 이행 함수적 종속성 (Transitive Functional Dependency)

# 완전 함수적 종속성

## ■ Case1. 종속자가 기본키에만 종속된 경우

학번	성명	성별	연락처	주소
10101	김철수	남	010-1111-1111	서울시 마포구
10102	최영희	여	010-2222-2222	서울시 강남구

- 기본키 : 학번
- 학번을 알면 성명, 성별, 연락처, 주소 정보를 알 수 있으므로  
성명, 성별, 연락처, 주소는 학번에 완전 함수적 종속된 관계이다.

# 완전 함수적 종속성

## ■ Case2. 종속자가 기본키를 구성하는 모든 칼럼에 종속된 경우

학번	과목코드	과목명	성적
10101	T1000	빅데이터	A+
10101	U1000	AI	B+

- 기본키 : 학번 + 과목코드
- 학번과 과목코드를 모두 알면 성적 정보를 알 수 있으므로 성적은 학번과 과목코드에 완전 함수적 종속된 관계이다.

# 부분 함수적 종속성

## ■ Case1. 종속자가 기본키가 아닌 칼럼에 종속된 경우

학번	성명	연락처	주소	지역번호
10101	김철수	010-1111-1111	서울시 마포구	02
10102	최영희	010-2222-2222	서울시 강남구	02

- 기본키 : 학번
- 주소를 알면 지역번호 정보를 알 수 있으므로 (주소는 PK가 아님)  
지역번호는 주소에 부분 함수적 종속된 관계이다.

# 부분 함수적 종속성

## ■ Case2. 종속자가 기본키를 구성하는 어느 한 칼럼에 종속된 경우

학번	과목코드	과목명	성적
10101	T1000	빅데이터	A+
10101	U1000	AI	B+

- 기본키 : 학번 + 과목코드
- 과목코드를 알면 과목명 정보를 알 수 있으므로 (과목코드는 PK를 구성하는 칼럼 중 하나) 과목명은 과목코드에 부분 함수적 종속된 관계이다.

```

each: function(e, t, n) {
    var r, i = 0,
        o = e.length,
        a = M(e);
    if (n) {
        if (a) {
            for (; o > i; i++)
                if (r = t.apply(e[i], n), r === !1) break;
        } else
            for (i in e)
                if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], e[i]), r === !1) break;
    } else
        for (i in e)
            if (r = t.call(e[i], e[i]), r === !1) break;
    return e;
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e);
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string"
),
isArray: function(e, t, n) {
    var r;
    if (t) {
        if (n) return m.call(t, e, n);
        for (r = t.length, r = n ? 0 > n ? Math.max(0, r + n) : r; r--;)
            if (n in t && t[n] === e) return n;
    }
}

```

### 3. 정규화

# 정규화

## ■ 정규화 (Normalization)

- 데이터베이스의 이상현상(Anomaly)을 방지하기 위한 방법
- 테이블에 존재하는 함수적 종속 문제를 해결하기 위해 데이터베이스 설계를 재구성하는 것

## ■ 정규화 법칙 (Rule)

- 제1정규형
- 제2정규형
- 제3정규형
- BCNF
- 제4정규형
- 제5정규형



# 제1정규형 - Step1

- 모든 칼럼(Column)은 원자값(Atomic Value)을 가져야 한다.

학번	성명	학과	과목
10101	이유림	영문학	Reading, Grammar
10102	이채림	국문학	고전문학
10103	이효림	체육학	재활

\* 원자성(Atomicity)  
: 칼럼의 데이터는 하나여야 한다.

\* 원자성 위배  
: 2개 이상의 값이 하나의 칼럼에  
포함되었음



학번	성명	학과	과목1	과목2
10101	이유림	영문학	Reading	Grammar
10102	이채림	국문학	고전문학	null
10103	이효림	체육학	재활	null

# 제1정규형 - Step2

- 모든 칼럼(Column)은 반복되는 그룹이 나타나지 않아야 한다.

학번	성명	학과	과목1	과목2
10101	이유림	영문학	Reading	Grammar
10102	이채림	국문학	고전문학	null
10103	이효림	체육학	재활	null

과목1, 과목2 칼럼은  
사실상 같은 칼럼이므로  
2개로 나누면 안 된다.



학번	성명	학과	과목1
10101	이유림	영문학	Reading
10101	이유림	영문학	Grammar
10102	이채림	국문학	고전문학
10103	이효림	체육학	재활

사실 아직은  
학번에 중복이 발생하여  
기본키 역할을 못한다.

# 제1정규형 - Step3

- 기본키(PK)와 특정 칼럼간의 함수적 종속 관계를 제거해야 한다.

학번	성명	학과	과목1
10101	이유림	영문학	Reading
10101	이유림	영문학	Grammar
10102	이채림	국문학	고전문학
10103	이효림	체육학	재활

학번을 알면 성명과 학과를 알 수 있는  
함수적 종속 관계가 있다.

종속 관계에 있는 학번, 성명, 학과를  
별도 테이블로 구성해야 한다.



학번	성명	학과
10101	이유림	영문학
10101	이유림	영문학
10102	이채림	국문학
10103	이효림	체육학

학번	과목1
10101	Reading
10101	Grammar
10102	고전문학
10103	재활

# 제1정규형 - Step4

- 기본키(PK)와 외래키를 부여하고 관계를 맺는다.

학번	성명	학과
10101	이유림	영문학
10102	이채림	국문학
10103	이효림	체육학

번호	학번	과목1
1	10101	Reading
2	10101	Grammar
3	10102	고전문학
4	10103	재활

각 테이블에 적절한 기본키를 부여하고,  
학번을 기준으로 관계를 맺는다.

## 제2정규형 - Step1

- 2개 이상의 칼럼으로 만든 기본키를 사용할 때만 발생한다.
- 1개의 칼럼으로 만든 기본키를 사용하면 제2정규형은 만족한다.
- 기본키를 구성한 칼럼 중에서 일부 칼럼에만 종속된 칼럼(부분 함수적 종속)이 존재하는 경우 이를 없앤다.

성명	학과		학과장	과목
이유림	영문학과	→	김과장	Reading
이채림	국문학과	→	최과장	고전문학
이효림	체육학과	→	송과장	재활

기본키(PK)를 성명과 학과  
2개의 칼럼으로 구성

학과와 학과장 사이에는  
부분 함수적 종속이 존재함  
(학과 데이터에 따라 학과장 데이터가  
결정된다.)

학과와 학과장 칼럼을  
따로 분리해야 함

# 제2정규형 - Step2

## ■ 부분 함수적 종속 제거

성명	학과	학과장	과목
이유림	영문학과	김과장	Reading
이채림	국문학과	최과장	고전문학
이효림	체육학과	송과장	재활



학과	학과장
영문학과	김과장
국문학과	최과장
체육학과	송과장

학과	성명	과목
영문학과	이유림	Reading
국문학과	이채림	고전문학
체육학과	이효림	재활

종속 관계에 있는 학과와 학과장 칼럼  
을 별도의 테이블로 분리 완료

# 제3정규형 - Step1

- 기본키가 아닌 일반 칼럼들 간의 종속 관계를 제거해야 한다.

<u>동아리회원번호</u>	성명	학교코드	학교전화
1	강백호	10	02-111-1111
2	김영호	20	032-222-2222
3	정일호	30	042-333-3333
4	최백호	40	051-444-4444

학교코드와 학교전화 사이에는  
종속 관계가 있음

학교코드와 학교전화 칼럼을  
따로 분리해야 함

# 제3정규형 - Step2

- 기본키가 아닌 일반 칼럼들 간의 종속 관계를 제거해야 한다.

<u>동아리회원번호</u>	성명	학교코드	학교전화
1	강백호	10	02-111-1111
2	김영호	20	032-222-2222
3	정일호	30	042-333-3333
4	최백호	40	051-444-4444



<u>동아리회원번호</u>	성명	학교코드
1	강백호	10
2	김영호	10
3	정일호	20
4	최백호	20

<u>학교코드</u>	학교전화
10	02-111-1111
20	02-222-2222

종속 관계의 학교코드와 학교전화  
칼럼을 서로 다른 테이블로 분리



# 정규형 실습 - 1NF

## ■ 다음 테이블을 제1정규화하시오.

- 모든 Domain은 원자값을 가져야 한다.
- 모든 칼럼에 반복 그룹(Repeating Group)이 없어야 한다.
- 기본키(PK)와 특정 칼럼간의 종속 관계를 제거해야 한다.
- 모든 테이블은 데이터 식별을 위해 기본키를 사용해야 한다.

고객번호	고객명	고객포인트	고객연락처
123	김다은	0	555-4444, 010-2222-3333
234	이다은	1000	505-3333, 010-3333-4444
345	정다은	500	2626-3333, 010-9999-2222
456	최다은	0	010-4444-5555

# 정규형 실습 - 1NF

원자값

고객번호	고객명	고객포인트	고객연락처1	고객연락처2
123	김다은	0	555-4444	010-2222-3333
234	이다은	1000	505-3333	010-3333-4444
345	정다은	500	2626-3333	010-9999-2222
456	최다은	0	010-4444-5555	

반복그룹 제거

고객번호	고객명	고객포인트	고객연락처
123	김다은	0	555-4444
123	김다은	0	010-2222-3333
234	이다은	1000	505-3333
234	이다은	1000	010-3333-4444
345	정다은	500	2626-3333
345	정다은	500	010-9999-2222
456	최다은	0	010-4444-5555

중속 제거 후  
최종 결과

고객번호	고객명	고객포인트	번호	고객번호	고객연락처
123	김다은	0	1	123	555-4444
234	이다은	1000	2	123	010-2222-3333
345	정다은	500	3	234	505-3333
456	최다은	0	4	234	010-3333-4444
			5	345	2626-3333
			6	345	010-9999-2222
			7	456	010-4444-5555

# 정규형 실습 - 2NF

- 다음 테이블을 제2정규화하시오.
  - 모든 칼럼이 완전 함수적 종속을 만족해야 한다.

<u>제조사</u>	<u>모델</u>	세부 모델명	제조국가
Benz	C-Class	C200	Germany
Benz	E-Class	E300	Germany
Hyundai	Genesis	G70	Korea
Hyundai	Genesis	G80	Korea
BMW	3-series	320D	Germany
BMW	5-series	520D	Germany

# 정규형 실습 - 2NF

- 기본키 중에서 제조사에만 종속 관계가 있어 이를 제거함
- 제조사와 제조국가의 부분 함수적 종속 제거
  - Benz -> Germany
  - Hyundai -> Korea
  - BMW -> Germany

제조사	제조국가
Benz	Germany
Hyundai	Korea
BMW	Germany

제조사	모델	세부 모델명
Benz	C-Class	C200
Benz	E-Class	E300
Hyundai	Genesis	G70
Hyundai	Genesis	G80
BMW	3-series	320D
BMW	5-series	520D

# 정규형 실습 - 3NF

- 다음 테이블을 제3정규화하시오.
  - 기본키가 아닌 칼럼 사이에 함수 종속이 없어야 한다.

대회	년도	우승자	우승자 국가
US 오픈	2010	제이슨 본	미국
프랑스 오픈	2015	에단 헌트	미국
웬블덴	2016	셜록 홈즈	영국
호주 오픈	2017	최배달	한국

# 정규형 실습 - 3NF

- 기본키가 아닌 우승자와 우승자 국가 사이에 종속 관계가 있어 이를 제거
- 우승자와 우승자 국가의 이행적 종속 제거
  - 제이슨 본 -> 미국
  - 에단 헌트 -> 미국
  - 셸록 홈즈 -> 영국
  - 최배달 -> 한국

대회	년도	우승자
US 오픈	2010	제이슨 본
프랑스 오픈	2015	에단 헌트
윌블던	2016	셸록 홈즈
호주 오픈	2017	최배달

우승자	우승자 국가
제이슨 본	미국
에단 헌트	미국
셸록 홈즈	영국
최배달	한국

# 반정규화란?

## ■ 반정규화

- DeNormalization
- 정규화된 릴레이션, 칼럼, 관계를 다시 조정하는 과정
- 릴레이션에 중복을 허용하고 분리된 릴레이션을 다시 통합함
- 반정규화로 인해 데이터 무결성이 깨질 수 있음

# 반정규화 필요성

## ■ 반정규화를 해야 하는 경우

- 데이터를 조회할 때 디스크 I/O량이 많아서 성능이 저하되는 경우
- 두 릴레이션의 경로가 너무 멀어 조인으로 인한 성능 저하가 예상되는 경우
- 칼럼을 계산하여 읽을 때 성능이 저하될 것이 예상되는 경우

## ■ 반정규화 목적

- 시스템의 IO 성능 향상
- 시스템 개발(Development)과 운영(Maintenance)의 단순화

## ■ 반정규화 종류

- 릴레이션 반정규화 : 릴레이션을 병합하거나 분할함
- 속성 반정규화 : 속성 또는 파생속성을 추가함