

**SQLD**

# **1. ENTITIY**

# SQLD

## 엔터티(Entity)

### 엔터티(Entity) :

Thomas Bruce : 정보가 저장될 수 있는 장소, 사람, 사건, 개념, 물건 등이다.

**엔터티의 특징 :** 엔터티는 반드시 속성을 가지고 있다.  
두 개 이상의 인스턴스가 있어야 한다.  
엔터티는 인스턴스를 식별할 수 있는 식별자가 있다.  
엔터티는 다른 엔터티와 최소 한 개 이상 관계가 있어야 한다.

# SQLD

## 엔터티-인스턴스의 예

엔터티	인스턴스
과 목	수 학
	영 어
강 사	이춘식
	조시형
사 건	2010-001
	2010-002

## 엔터티와 인스턴스

# SQLD

## 엔터티 분류

### 1) 유/무형

**유형엔터티** : 물리적 형태가 존재하는 것  
ex) 고객, 강사, 사원

**무형엔터티** : 물리적 형태가 존재하지 않는, 개념적으로 사용되는 것  
ex) 거래소 종목, 코스닥 종목 등등

**사건엔터티** : 비즈니스 프로세스 과정에서 생성되는 엔터티  
ex) 환불 건, 수수료 등등

# SQLD

## 엔터티 분류

### 2) 발생 시점

**기본(키)엔터티** : 독립적으로 생성되는 것.

ex) 고객, 상품, 부서

**중심(메인)엔터티** : 기본엔터티로 부터 발생되고 행위엔터티를 생성하는 것

ex) 계좌, 고객이 계좌를 개설한다. 주문, 고객이 상품을 주문한다.

**행위(Activity)엔터티** : 중심엔터티에 의해 생성되는 것,

ex) 주문 이력, 체결 이력 등등

# SQLD

## 속성 : 엔터티의 항목

속성은 더 이상 분리되지 않는 단위이다.

인스턴스의 구성요소이다. (인스턴스는 속성들의 값의 집합이다.)

## 속성의 종류

### 1. 분해여부(의미,속성)

단일 속성 : 하나의 의미로 구성된 것으로 회원ID, 이름 등이다.

복합 속성 : 여러 개의 의미가 있는 속성. 주소(도,시,군 등으로 분해가능)

다중 값 속성 : 하나의 속성이 여러 개의 값을 가지는 경우. 분해해야 한다.

### 2. 특성에 따른 분류

기본 속성 : 비즈니스 프로세스에서 도출되는 본래의 속성

설계 속성 : 데이터모델링 과정에서 발생하는 속성(유일한 값을 부여한다) ex) 상품코드, 지점코드

파생 속성 : 다른 속성에 의해 만들어지는 속성 ex) 평균, 합계 등등

# SQLD

**식별자 : 유일성을 만족하는 속성 (중복 안됨)**

1. 슈퍼키 : 유일성은 만족하지만 최소성은 만족 안 함
2. 후보키 : 유일성과 최소성 만족
3. 기본키 : 후보키들 중에서 엔터티를 대표하는 키(테이블당 하나만 존재)
4. 대체키 : 후보키 중에서 기본키 빼고 나머지들.



# SQLD

## 식별자 분류

### 1. 대표성 여부

주식별자 : 기본키

보조식별자 : 대체키

### 2. 생성여부

내부식별자 : 엔터티 내부에서 생성되는 식별자

외부식별자 : 관계에 의해 만들어지는 식별자

### 3. 속성의 수

단일식별자 : 속성이 하나

복합식별자 : 두개 이상의 속성

### 4. 대체 여부

본질식별자 : 비즈니스프로세스 상에서 만들어지는 식별자

인조식별자 : 인위적으로 만들어내는 식별자(상품코드)

## **2. RELATION**

# SQLD

## 관계

엔터티 간의 관련성을 의미  
존재관계와 행위관계가 존재한다.

**존재관계** : 두 엔터티가 존재여부(속함)의 관계에 있는 것.  
ex) 고객-관리점, 학생-반, 상품-카테고리

**행위관계** : 두 엔터티간의 행위가 있는 것  
ex) 고객 - 계좌  
고객은 계좌를 만들지, 계좌에 속하지 않는다.

# SQLD

## 관계 표시

### 필수 여부

**필수적 관계** : 반드시 하나는 존재해야 하는 관계 |로 표시

**선택적 관계** : 없을 수도 있는 관계 ○로 표시

### 식별여부

**식별관계** : 부모테이블의 키를 자식테이블에서도 키로 사용하는 경우  
부모테이블의 키를 통해 자식테이블을 특정할 수 있다. 실선으로 표현

**비식별관계** : 부모테이블의 키를 자식테이블이 일반 컬럼으로 사용하는 경우 :  
부모테이블의 키를 통해 자식테이블을 특정할 수 없다. 점선으로 표현

# SQLD

## 관계차수

### 1:1관계

완전 1:1관계 : 엔터티에 행이 하나 있을 때 다른 엔터티에 관계맺는 행이 반드시 하나 존재하는 관계

### 선택적 1:1관계 :

엔터티에 행이 하나 있을 때 다른 엔터티에 관계 맺는 행이 하나이거나 없을 수도 있다.

#### [IE 표기법]

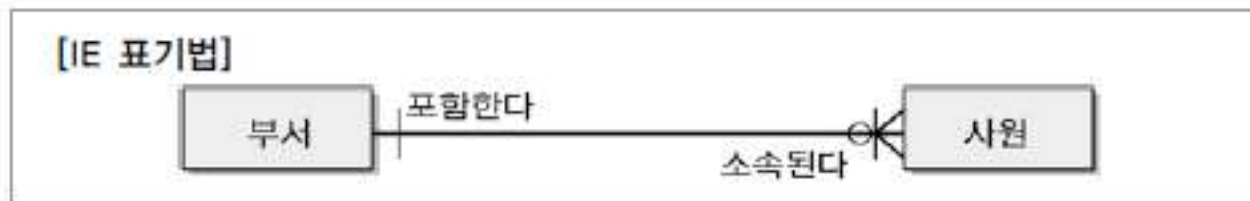


# SQLD

## 관계차수

### 1:N관계 :

엔터티에 행이 하나 있을 때 다른 엔터티에 관계 맺는 행이 여러 있는 관계. 고객-아이디 (넥슨 아이디 3개까지 가능)

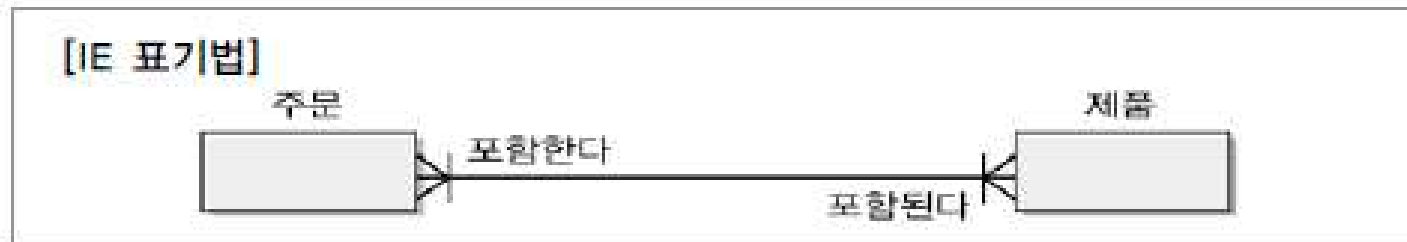


# SQLD

## 관계차수

### M:N관계 :

두개의 엔터티가 서로 여러개의 관계를 가지고 있는 경우  
학생은 여러 과목을 수강하고 한 과목은 여러 학생이 듣는다.  
JOIN시 카텐시아곱이 발생함으로 이 경우 1:N로 해소해 줘야 한다.



# **3. MODELING**



# SQLD

## 데이터 모델링의 특징

**추상화:** 현실세계를 일정한 공통적인 특징을 뽑아 간략하게 표현

**단순화:** 현실세계를 약속된 규약이나 제한된 표기법과 언어로 표현

**명확화:** 누구나 이해하기 쉽게 애매모호함을 제거하고 정확하게 현상을 기술

## 데이터 모델링의 관점

**데이터 :** 비즈니스 프로세스에서 사용되는 데이터

**프로세스 :** 비즈니스 프로세스에서 수행되는 작업

**데이터와 프로세스(상관관계) :** 데이터와 프로세스간의 관계.

로그인 프로세스를 위해서 회원 데이터가 필요하다.

회원정보수정 프로세스가 일어나면 회원 데이터가 변경된다.

# SQLD

## 데이터 독립성

### 데이터 독립성 :

데이터베이스의 구조 변화로 인한 영향을  
프로그램에 미치지 않도록 하는 것  
DBMS의 궁극적인 목표로 응용프로그램이  
데이터에 종속되지 않는 것을 의미한다

**논리적 독립성** : 개념스키마가 변경되어도 외부스키마  
즉 어플리케이션에는 영향이 없다.

**물리적 독립성** : 내부스키마가 변경되어도 개념스키마나 외부스키마에는  
영향이 없다.

독립성을 확보하는 방법 : **3층스키마, 정규화**

# SQLD

## 데이터 모델링의 단계

**개념적 모델링** : 기업의 비즈니스 프로세스를 분석하고 기업 전체에 대한 데이터 모델링 수행 업무 관점 모델링, Entity와 속성을 도출 개념적 ERD 작성

**논리적 모델링** : 식별자를 도출하고 필요한 모든 관계(릴레이션) 정의 정규화를 수행하여 데이터 모델의 독립성 확보

**물리적 모델링** : 데이터베이스를 실제로 구축

# SQLD

**스키마란 ? :**

데이터베이스에서 자료의 구조, 자료의 표현 방법, 자료 간의 관계를 정의한 것

**외부스키마 :** 사용자 관점 업무상 관련이 있는 데이터

관련 데이터베이스의 VIEW를 표시(업무에 따라 볼 수 있는 데이터가 다름)

응용 프로그램이 접근하는 데이터베이스를 정의한다.

**개념 스키마 :** 설계자 관점, 사용자 전체 집단의 데이터베이스 구조

전체 데이터베이스 내의 규칙과 구조를 표현

EX) 데이터 접근 권한, 엔터티 관계, 보안 등등

**내부 스키마 :** 개발자 관점, 데이터베이스의 물리적 저장구조.

# SQLD



# SQLD

**정규화** : 데이터 중복을 최소화 하고 이상현상이 발생하지 않도록 데이터베이스를 설계하는 과정

## 이상현상(Anomaly)

삽입이상(Insertion Anomaly)

- 새 데이터를 추가하기 위해 불필요한 데이터도 함께 추가해야 하는 경우

갱신이상(Update Anomaly)

- 중복된 데이터들 중 일부만 변경할 경우 데이터가 불일치가 발생

삭제이상(Deletion Anomaly)

- 데이터를 삭제할 경우 반드시 필요한 데이터가 함께 삭제되는 경우

# SQLD

**정규화** : 데이터 중복을 최소화 하고 이상현상이 발생하지 않도록 데이터베이스를 설계하는 과정

## 정규화 하는 법

부분함수적 종속을 제거해 완전함수적 종속이 되도록 수정하고  
완전함수적 종속일 때 이행적 함수 종속이 발생하지 않도록 하며  
모든 결정자가 후보키 집합에 속하도록 한다.

# SQLD

## 함수적 종속

A(결정자)  $\rightarrow$  B(종속자)

B는 A에 함수적 종속이다. A가 B를 결정한다.

A이면 B이고 동시에 A이면 C일 수 없지만, B이면 반드시 A인 것은 아니다.

EX) 아이디가 DE이면 일반회원이고, 아이디가 DE이면 우수회원일 수는 없지만 일반회원이면 반드시 아이디가 DE인 것은 아니다.

## 함수적 종속 분류

완전함수적 종속 : 종속자가 기본키에만 종속되는 경우.

기본키가 여러 개의 속성으로 이루어져 있다면 모든 속성에 종속되는 경우

부분함수적 종속 : 기본키를 구성하는 여러 속성 중 일부 속성에 종속되는 경우

이행적 함수 종속 :  $X \rightarrow Y, Y \rightarrow Z$  :

X를 통해 Y를 알 수 있고, Y를 통해 Z를 알 수 있는 경우



# SQLD

## 정규화 할 테이블

학생번호	이름	학과	학과전화번호	과목번호	강사	학점
11002	김애란	문예창작학과	02-1111-1111	A101, B201	하명도, 이동헌	A,B
97654	헤밍웨이	문예창작학과	02-1111-1111	B201	이동헌	A
24516	빌게이츠	컴퓨터공학과	02-2222-2222	A101	하명도	B
42555	프레디머큐리	실용음악학과	02-3233-3633	D441	이창진	B

# SQLD

## 정규화

제 1정규화 : 속성의 원자성을 확보한다. 기본키를 설정한다.

<u>학생번호</u>	이름	학과	학과전화번호	<u>과목번호</u>	강사	학점
11002	김애란	문예창작학과	02-1111-1111	A101	하명도	A
11002	김애란	문예창작학과	02-1111-1111	B201	이동헌	B
97654	헤밍웨이	문예창작학과	02-1111-1111	B201	이동헌	A
24516	빌게이츠	컴퓨터공학과	02-2222-2222	A101	하명도	B
42555	프레디머큐리	실용음악학과	02-3233-3633	D441	이창진	B

# SQLD

## 정규화

제 2정규화 : 기본키가 2개 이상의 속성으로 구성된 경우 부분함수종속성을 제거한다.

<u>학생번호</u>	이름	학과	학과전화번호	<u>과목번호</u>	강사	학점
11002	김애란	문예창작학과	02-1111-1111	A101	하명도	A
11002	김애란	문예창작학과	02-1111-1111	B201	이동헌	B
97654	헤밍웨이	문예창작학과	02-1111-1111	B201	이동헌	A
24516	빌게이츠	컴퓨터공학과	02-2222-2222	A101	하명도	B
42555	프레디머큐리	실용음악학과	02-3233-3633	D441	이창진	B

# SQLD

## 정규화

제 2정규화 : 기본키가 2개 이상의 속성으로 이루어진 경우 부분함수종속성을 제거한다.

<u>학생번호</u>	<u>과목번호</u>	학점	<u>과목번호</u>	강사	<u>학생번호</u>	이름	학과	학과전화번호
11002	A101	A	A101	하명도	11002	김애란	문예창작학과	02-1111-1111
11002	B201	A	B201	이동헌	97654	헤밍웨이	문예창작학과	02-1111-1111
24516	A101	B	D441	이창진	24516	빌게이츠	컴퓨터공학과	02-2222-2222
42555	D441	B			42555	프레디머큐리	실용음악학과	02-3233-3633

# SQLD

## 정규화

제 3정규화 : 기본키를 제외한 컬럼 간의 종속성을 제거한다. 즉 이행함수종속성을 제거한다.

<u>학생번호</u>	이름	학과	학과전화번호
11002	김애란	문예창작학과	02-1111-1111
97654	헤밍웨이	문예창작학과	02-1111-1111
24516	빌게이츠	컴퓨터공학과	02-2222-2222
42555	프레디머큐리	실용음악학과	02-3233-3633

# SQLD

## 정규화

**제 3정규화** : 기본키를 제외한 컬럼 간의 종속성을 제거한다. 즉 이행함수종속성을 제거한다.

학생 테이블

<u>학생번호</u>	이름	학과
11002	김애란	문예창작학과
97654	헤밍웨이	문예창작학과
24516	빌게이츠	컴퓨터공학과
42555	프레디머큐리	실용음악학과

학과 테이블

<u>학과</u>	학과전화번호
문예창작학과	02-1111-1111
컴퓨터공학과	02-2222-2222
실용음악학과	02-3233-3633

# SQLD

## 정규화 완료

학생 테이블

학생번호	이름	학과
11002	김애란	문예창작학과
97654	헤밍웨이	문예창작학과
24516	빌게이츠	컴퓨터공학과
42555	프레디머큐리	실용음악학과

학과 테이블

학과	학과전화번호
문예창작학과	02-1111-1111
컴퓨터공학과	02-2222-2222
실용음악학과	02-3233-3633

학점 테이블

학생번호	과목번호	학점
11002	A101	A
11002	B201	A
24516	A101	B
42555	D441	B

과목테이블

과목번호	강사
A101	하명도
B201	이동헌
D441	이창진

# SQLD

## 정규화

**BCNF** : 모든 결정자가 후보키(유일성과 최소성을 만족시켜라)

-> 학과가 학과전화번호를 결정하는데 학과는 유일성과 최소성을 만족하지 않음

<u>학생번호</u>	<u>과목번호</u>	학과	학과전화번호	학점
11002	A101	문예창작학과	02-1111-1111	A
11002	B201	문예창작학과	02-1111-1111	A
24516	A101	컴퓨터공학과	02-2222-2222	B
42555	D441	실용음악학과	02-3233-3633	B



# SQLD

## 정규화

BCNF : 모든 결정자가 후보키

성적 테이블

<u>학생번호</u>	<u>과목번호</u>	학과	학점
11002	A101	문예창작학과	A
24516	B201	컴퓨터공학과	B
42555	D441	실용음악학과	B

과목 테이블

<u>학과</u>	학과전화번호
문예창작학과	02-1111-1111
컴퓨터공학과	02-2222-2222
실용음악학과	02-3233-3633

# SQLD

## 반정규화

데이터베이스 성능향상을 위해서 데이터 중복을 허용하고 조인을 줄이는 방법

### \* 수행하는 경우

- 1) 수행속도가 느린 경우
- 2) 다량의 범위를 자주 처리해야 하는 경우
- 3) 특정 범위의 데이터만 자주 처리하는 경우

반정규화는 최후의 수단이다.

절차 : 대상 조사 및 검토 – 다른 방법 검토 – 반정규화 수행

# SQLD

## 반정규화

데이터베이스 성능향상을 위해서 데이터 중복을 허용하고 조인을 줄이는 방법

1. 계산된 컬럼 추가 : 미리 필요한 연산을 마친 특정 컬럼을 추가한다.
2. 테이블을 수직 분할 한다. 컬럼을 분할하여 새로운 테이블을 만든다.
3. 테이블을 수평 분할 한다. 특정 값을 기준으로 ROW를 분리한다.
4. 테이블 병합 : 정규화를 통해 분리한 테이블을 다시 하나로 합친다.

# SQLD

## 반정규화

데이터베이스 성능향상을 위해서 데이터 중복을 허용하고 조인을 줄이는 방법

### 테이블 병합

1:1 관계의 테이블을 하나의 테이블로 합친다.

1:n 관계의 테이블을 하나의 테이블로 합친다. (다량의 중복 발생)

슈퍼타입과 서브타입 관계가 발생하면 하나의 테이블로 합친다.

\*슈퍼타입 : 고객

\*서브타입 : 개인고객, 법인 고객

\*배타적 관계 : 개인 고객일 수는 있어도 법인 고객일 수는 없다.

\*포괄적 관계 : 개인고객일 수도 법인고객일 수도 있다.