

14

강

컴퓨터과학 개론

데이터베이스 (2)

컴퓨터과학과 이관용 교수



KOREA NATIONAL OPEN UNIVERSITY



학습목차

1 관계형 모델

2 데이터베이스 설계

3 SQL

01

관계형 모델

관계형 데이터 모델

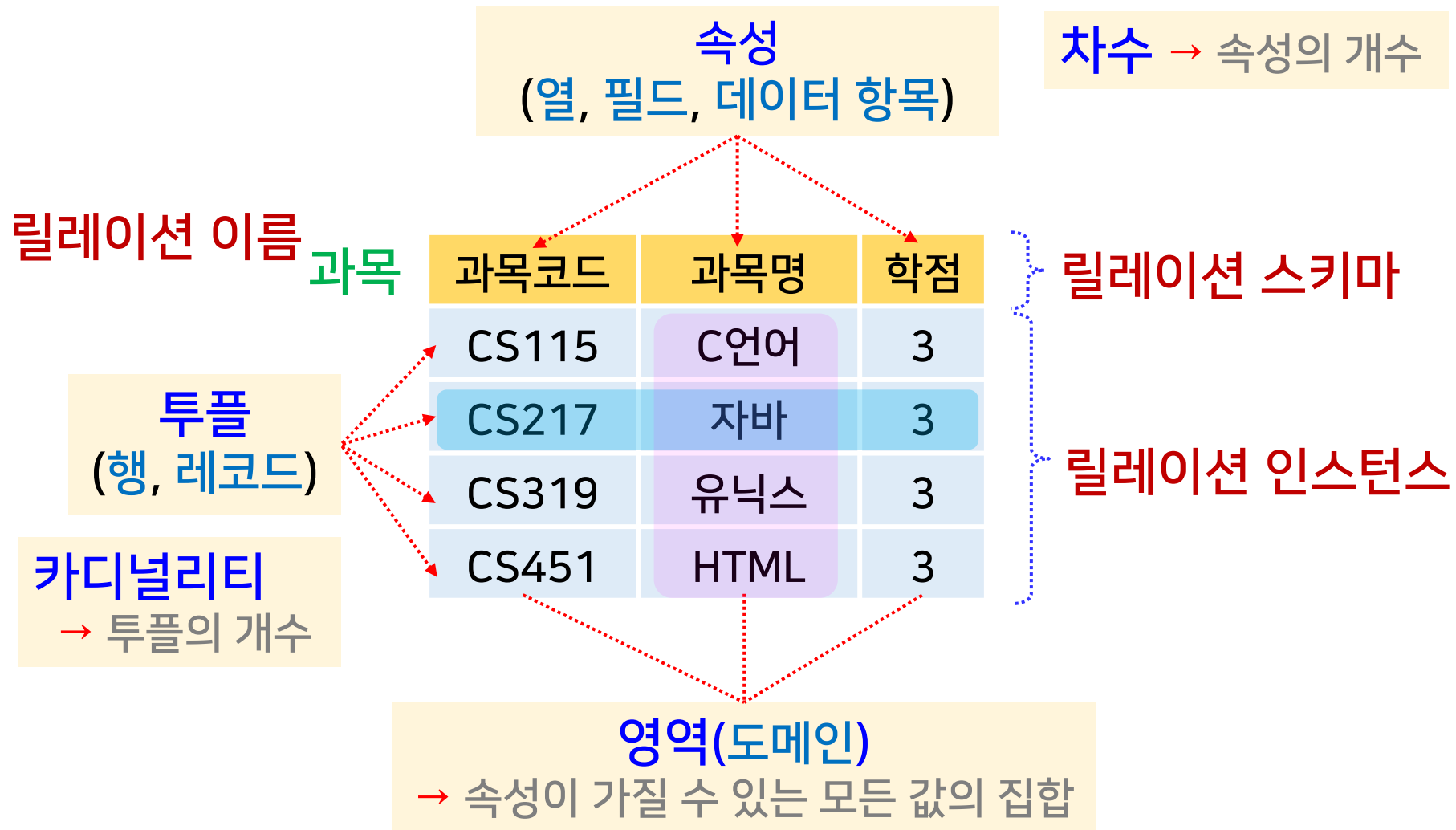
■ 실세계 정보를 **2차원 테이블**(“릴레이션”) 형식으로 표현하고 구현

- 테이블을 이용하여 데이터와 데이터의 관계를 표현
 - “데이터베이스를 2차원 테이블의 집합”으로 간주
 - 실제 데이터가 테이블 형태로 저장되는 것을 의미하지 않음

고객	주민등록번호	이름	주소	신용도
	631212-1234567	홍사장	대구 동구	92
	750101-2345678	김여사	경기도 광명시	45

은행계좌	계좌번호	잔고	주민등록번호
	123-17418	1,003,000	631212-1234567
	102-00419	34,500	750101-2345678
	055-31674	879,000	750101-2345678

관계형 모델과 관련된 용어



릴레이션의 개념

■ DB는 여러 개의 릴레이션으로 구성

- 릴레이션이 가질 수 있는 모든 속성값의 모든 경우의 수에 대한 부분집합

■ 릴레이션 = 릴레이션 스키마 + 릴레이션 인스턴스

- 릴레이션 스키마 relation schema, 릴레이션 내포 intension
 - 릴레이션의 논리적 구조 → 릴레이션 이름과 속성으로 구성
 - 시간에 무관하며 단지 속성에 대한 타입 지정
- 릴레이션 인스턴스 relation instance, 릴레이션 외포 extension
 - 어느 한 시점에 릴레이션이 가지고 있는 튜플의 집합
 - 삽입, 삭제, 갱신 등을 통해 시간에 따라 변하는 릴레이션의 값

릴레이션의 특징

■ 튜플의 유일성

- 하나의 릴레이션에는 중복된 튜플이 없음 → 튜플의 유일한 식별이 가능

■ 튜플의 무순서성

- 한 릴레이션에 포함된 튜플들은 순서를 가지고 있지 않음

■ 속성의 무순서성

- 하나의 릴레이션을 구성하는 속성들 사이에는 순서가 없음

■ 속성값의 원자성

- 모든 속성의 값은 더 이상 분해가 불가능한 하나의 값인 원자값을 가짐

키

■ 튜플을 유일하게 구별하기 위한 속성 또는 속성의 집합

- 유일성
 - 키값으로 하나의 튜플을 유일하게 식별함
- 최소성
 - 키는 모든 튜플을 유일하게 식별할 수 있는 최소의 속성들로 구성됨

■ 종류

- 슈퍼키 super key, 후보키 candidata key,
기본키 primary key, 대체키 alternate key, 외래키 foreign key

키의 종류

- **슈퍼키** → 유일성을 만족하는 속성 또는 속성의 집합
- **후보키** → 슈퍼키 중에서 최소성을 만족하는 것
- **기본키** → 후보키 중에서 기본적으로 사용할 키로 선택된 것
- **대체키** → 후보키 중에서 기본키로 선택되지 않은 것
- **외래키**
 - 다른 릴레이션의 기본키를 그대로 참조하는 속성 또는 속성의 집합
 - 상호 관련 있는 릴레이션 사이에서 데이터 일관성을 유지하는 수단

제약 조건

■ 모든 릴레이션 인스턴스가 만족해야 하는 조건

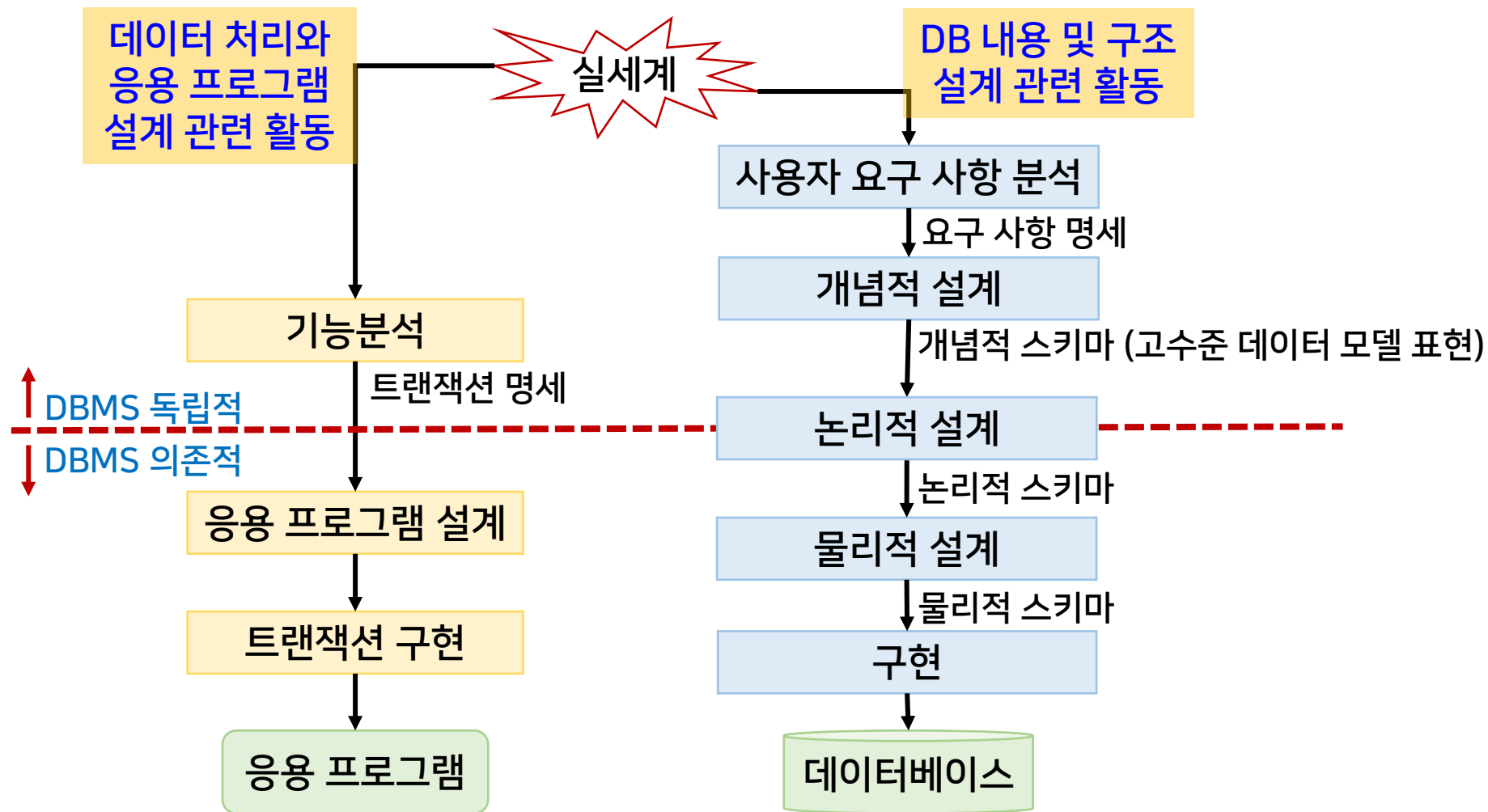
- 영역 제약 조건 domain constraint
 - 각 속성의 값은 반드시 해당 영역에 포함되며, 원자값이어야 함
- 키 제약 조건 key constraint
 - 서로 다른 두 튜플도 모든 속성에 대해서 같은 속성값의 조합을 가질 수 없음
→ 하나의 키가 모든 튜플을 유일하게 식별할 수 있어야 함
- 개체 무결성 제약 조건 entity integrity constraint
 - 튜플을 유일하게 식별하기 위해서 어떠한 기본키값도 널값이 될 수 없음
- 참조 무결성 제약 조건 referential integrity constraint
 - 다른 튜플에 의해 참조되는 튜플은 반드시 해당 릴레이션 내에 존재해야 함

02

데이터베이스 설계

데이터베이스 설계

■ 사용자의 요구 조건에서부터 DB 구조를 도출하고 구축하는 과정



설계 과정

■ ① 사용자 요구 사항 분석

- 사용자가 의도하는 데이터베이스의 목적과 용도 파악
- 잠정적인 주요 사용자의 범위 식별
- 공식적인 요구 조건 명세 정의
 - 정적 정보 구조(DB 내용/구조 설계)에 대한 요구 조건
 - 동적 DB 처리(데이터 처리, 응용 프로그램) 요구 조건
 - 범기관적 제약 조건
- 요구 사항 명세서, 요구 사항 정의서 등의 문서화 작업까지 포함

설계 과정

■ ② 개념적 DB 설계

- 개체 타입과 개체 타입 간의 관계를 나타내기 위해
특정 DBMS와는 무관한 개념적 데이터 모델을 사용하여
데이터베이스에 대한 개념적 구조를 생성하는 과정
 - 개념적 구조 → 사용자의 요구 사항을 간단히 기술한 것으로,
데이터 타입, 관계, 제약 조건을 설명
- E-R 모델 → 관계형 데이터베이스를 위한 대표적 개념적 모델
- 트랜잭션(응용 프로그램) 모델링

설계 과정

■ ③ 논리적 DB 설계

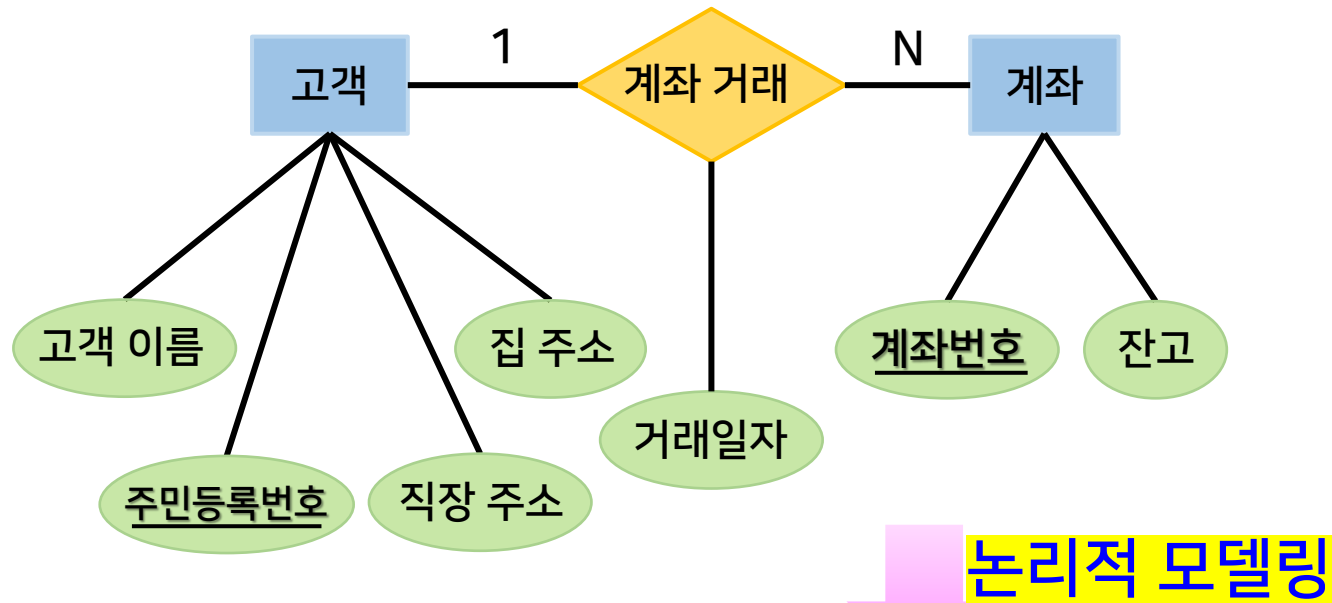
- 상용 DBMS를 사용해서 고수준의 개념적 구조를 목표 DBMS의 논리적 데이터 모델로 변환하는 과정
 - “논리적 데이터 모델링 ”
 - 결과 → 목표 DBMS의 데이터 정의어로 기술된 스키마
- 입출력과 기능적 형태로만 정의된 트랜잭션에 대한 인터페이스 설계

설계 과정_논리적 설계

■ 논리적 모델링

- 개념적 구조를 목표 DBMS의 구조로 변환하는 과정
- E-R 다이어그램으로부터 관계형 데이터 모델로의 변환 과정
 - 사각형의 개체 타입 → 개체 릴레이션으로 표현
 - 개체 타입에 속한 속성은 해당 개체 릴레이션의 속성이 됨
 - 마름모의 관계 타입 → 관계 릴레이션으로 표현
 - 관계 타입에 속한 속성은 해당 관계 릴레이션의 속성이 됨
 - 연관된 개체 타입의 키 속성을 관계 릴레이션의 속성으로 포함시킴

설계 과정_논리적 설계

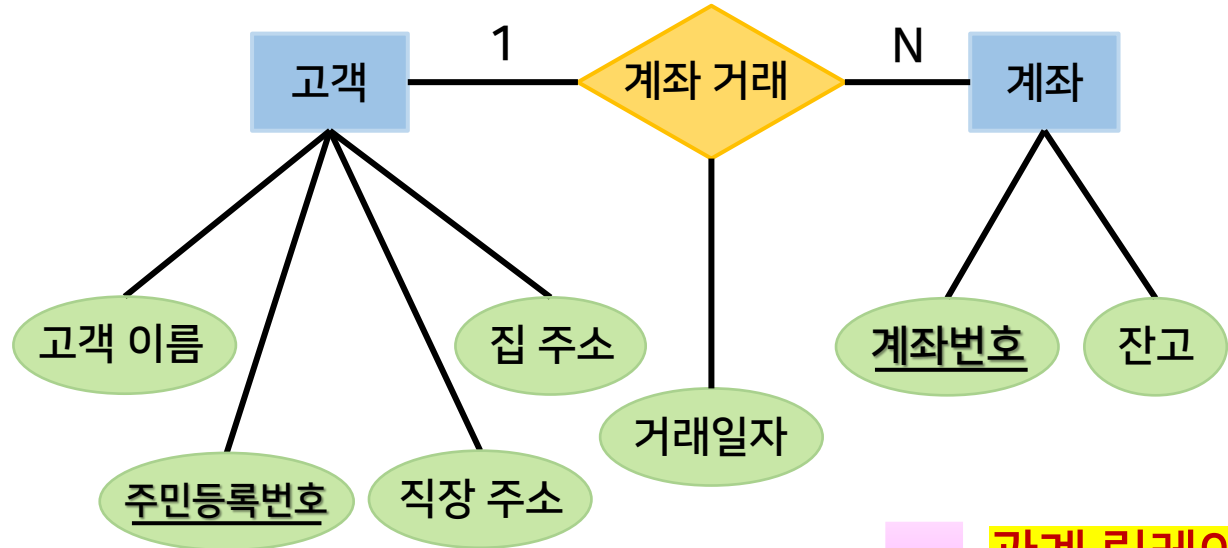


고객 주민등록번호 고객 이름 직장 주소 집 주소

계좌 계좌번호 잔고

계좌 거래 주민등록번호 계좌번호 거래일자

설계 과정_논리적 설계



관계 릴레이션을 사용하지 않고
묵시적으로 관계를 표현하는 경우

고객	<u>주민등록번호</u>	고객 이름	직장 주소	집 주소
계좌	<u>계좌번호</u>	잔고	주민등록번호 (외래키)	거래 일자

설계 과정

■ ④ 물리적 DB 설계

- 논리적 구조로부터 효율적이고 구현 가능한 물리적 데이터베이스 구조를 설계하는 것
 - 물리적 DB 구조 → 데이터베이스에 포함될 여러 파일 타입에 대한 저장 레코드의 양식, 순서, 접근 경로, 저장 공간의 할당 등을 표현한 것
- 트랜잭션에 대한 세부적인 설계

설계 과정

■ ⑤ 구현

- 목표 DBMS의 데이터 정의어로 기술된 명령문을 컴파일하고 실행해서
- 데이터베이스 스키마와 빈 데이터베이스 파일 생성
 - 실제 데이터 적재 → 데이터베이스 실행 및 운영
- 트랜잭션의 구현

03

SQL

SQL

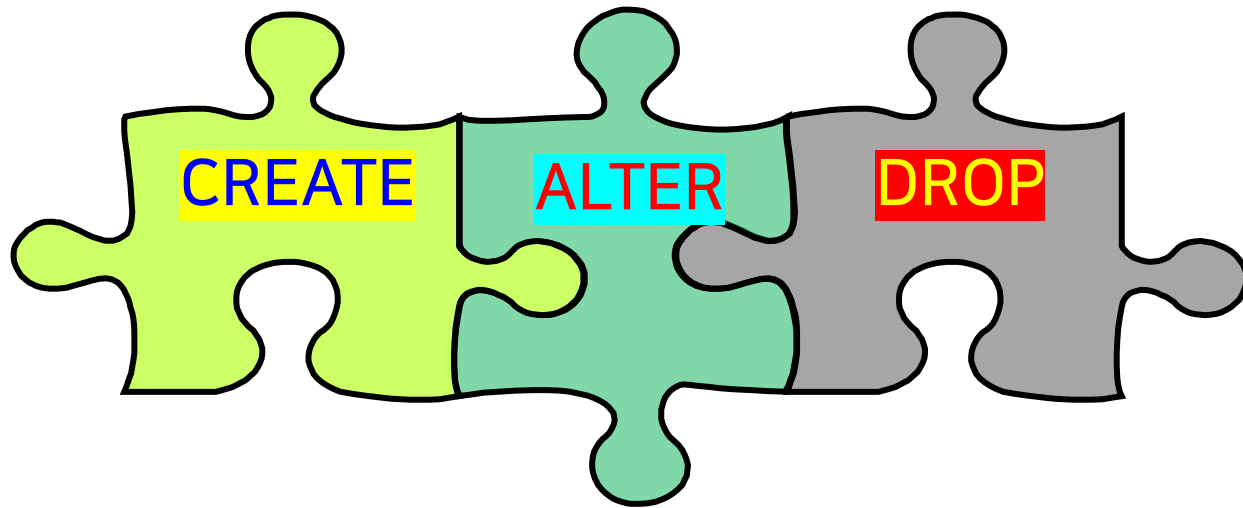
■ Structured Query Language

- “구조화된 질의어”
 - IBM 관계 DB 시스템인 SYSTEM R을 위해 설계
 - 관계형 모델의 DB에서 널리 사용
- 검색을 위한 단순한 질의어가 아닌 “종합적인 DB 언어”
 - 데이터의 정의, 조작, 제어 기능을 모두 제공
- 온라인 단말기를 통한 대화식 사용 + 부속어(삽입어) 사용
- 릴레이션, 튜플, 속성 등보다는 테이블, 행, 열과 같은 용어 표현을 선호
- 레코드 집합 단위의 연산을 수행하는 비절차적 언어

데이터 정의를

- 스키마 **SCHEMA**, 도메인 **DOMAIN**, 테이블 **TABLE**, 뷰 **VIEW**, 인덱스 **INDEX**를 정의하거나 수정 및 제거하는 문장으로 구성

- 명령문의 종류



데이터 정의어_기본적인 데이터 타입

■ 숫자

- 정수 → TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
- 실수 → FLOAT, DOUBLE
- 정형 숫자 → DECIMAL(p,s), NUMERIC

■ 문자

- CHAR(n), VARCHAR(n), TEXT

■ 날짜/시간

- DATE, TIME, DATETIME

데이터 정의어_테이블 생성

공급자

공급자번호

공급자이름

거래포인트

도시

```
CREATE TABLE 공급자 (  
  공급자번호 CHAR(5) NOT NULL,  
  공급자이름 VARCHAR(20),  
  거래포인트 SMALLINT,  
  도시 VARCHAR(20),  
  PRIMARY KEY(공급자번호)  
);
```

공급자 번호	공급자 이름	거래 포인트	도시
S1	이승주	20	서울
S2	이동일	10	대전
S3	이상현	30	대전
S4	소병식	20	서울
S5	황대섭	30	광주

데이터 정의어-테이블 생성

부품 부품번호 부품이름 색상 무게 창고소재지

```
CREATE TABLE 부품 (
  부품번호 CHAR(5) NOT NULL,
  부품이름 VARCHAR(20),
  색상 CHAR(6),
  무게 SMALLINT,
  창고소재지 VARCHAR(20)
  PRIMARY KEY(부품번호)
);
```

부품 번호	부품 이름	색상	무 게	창고 소재지
P1	너트	적	12	서울
P2	볼트	녹	17	대전
P3	스크루	청	17	대구
P4	스크루	적	14	서울
P5	캠	청	12	대전
P6	콕	적	19	서울

데이터 정의를 테이블 생성

납품 납품번호 공급자번호 부품번호 수량

```
CREATE TABLE 납품 (
  납품번호 CHAR(5) NOT NULL,
  공급자번호 CHAR(5),
  부품번호 CHAR(5),
  수량 INT,
  PRIMARY KEY(납품번호),
  FOREIGN KEY(공급자번호)
    REFERENCES 공급자(공급자번호),
  FOREIGN KEY(부품번호)
    REFERENCES 부품(부품번호)
);
```

납품 번호	공급자 번호	부품 번호	수량
D1	S1	P1	300
D2	S1	P2	200
D3	S1	P3	400
D4	S1	P4	200
D5	S1	P5	100
D6	S1	P6	100
D7	S2	P1	300
D8	S2	P2	400
D9	S3	P2	200
D10	S4	P2	200
D11	S4	P4	300
D12	S4	P5	400

데이터 정의어_테이블 변경

■ 새로운 열 추가

ALTER TABLE 테이블이름

ADD 열이름 데이터타입 [NOT NULL] [DEFAULT 기본값];

ALTER TABLE 납품 ADD 납품일 DATETIME;

■ 기존 열 삭제

ALTER TABLE 테이블이름

DROP 열이름 (CASCADE | RESTRICT);

ALTER TABLE 공급자 DROP 도시 CASCADE;

■ 기존 열에 대한 기본값 지정 및 삭제

ALTER TABLE 테이블이름

ALTER 열이름 (DROP DEFAULT | SET DEFAULT 기본값);

ALTER TABLE 납품 ALTER 수량 SET DEFAULT 100;

데이터 정의어_테이블 삭제

■ 명시된 기본 테이블을 시스템으로부터 제거

DROP TABLE 테이블이름 [CASCADE | RESTRICT];

DROP TABLE 공급자 RESTRICT;

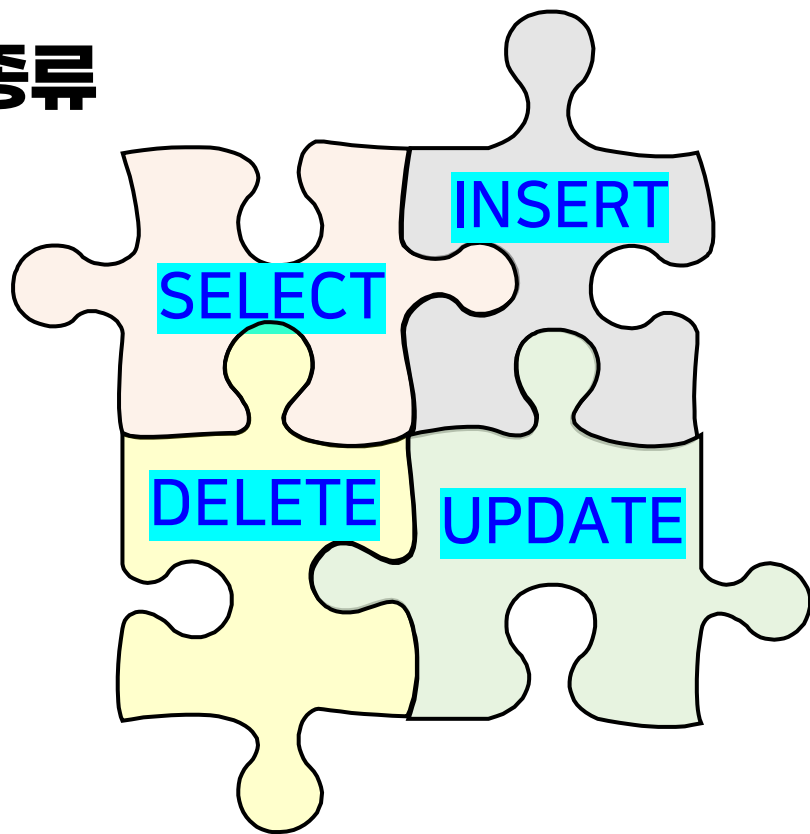
- 테이블에 대한 설명이 카탈로그에서 제거되므로
기본 테이블에서 정의된 모든 인덱스와 뷰도 자동적으로 삭제됨

데이터 조작용어

■ 조작(검색, 갱신, 삭제, 삽입) 대상

- 기본 테이블, 뷰

■ 명령문의 종류



데이터 조작어_데이터 검색 SELECT문

SELECT [ALL | DISTINCT] 열_리스트
FROM 테이블_리스트
 [WHERE 조건]
 [GROUP BY 열
 [HAVING 조건]]
 [ORDER BY 열 [ASC | DESC]];

SELECT 부품번호
 FROM 납품;

SELECT **DISTINCT** 부품번호
 FROM 납품;

SELECT *
 FROM 납품;

납품

납품 번호	공급자 번호	부품 번호	수량
D1	S1	P1	300
D2	S1	P2	200
D3	S1	P3	400
D4	S1	P4	200
D5	S1	P5	100
D6	S1	P6	100
D7	S2	P1	300
D8	S2	P2	400
D9	S3	P2	200
D10	S4	P2	200
D11	S4	P4	300
D12	S4	P5	400

데이터 조작어_데이터 검색 SELECT문

SELECT 공급자번호, 거래포인트 AS 거래점수
FROM 공급자
WHERE 거래포인트>20 AND 도시='대전';

SELECT 공급자번호, 거래포인트
FROM 공급자
WHERE 도시='대전'
ORDER BY 거래포인트 DESC;

SELECT 공급자.*, 부품.*
FROM 공급자, 부품
WHERE 공급자.도시=부품.창고소재지;

공급자

공급자 번호	공급자 이름	거래 포인트	도시
S1	이승주	20	서울
S2	이동일	10	대전
S3	이상현	30	대전
S4	소병식	20	서울
S5	황대섭	30	광주

부품

부품 번호	부품 이름	색상	무게	창고 소재지
P1	너트	적	12	서울
P2	볼트	녹	17	대전
P3	스크루	청	17	대구
P4	스크루	적	14	서울
P5	캠	청	12	대전
P6	콕	적	19	서울

데이터 조작성_데이터 검색 SELECT문

```
SELECT COUNT(*) AS 개수, SUM(수량) AS 수량합계
FROM 납품
WHERE 부품번호='P2';
```

```
SELECT 부품번호
FROM 납품
GROUP BY 부품번호
HAVING COUNT(*)>=2;
```

```
SELECT *
FROM 부품
WHERE 부품이름 LIKE '스%';
```

부품

부품 번호	부품 이름	색상	무게	창고 소재지
P1	너트	적	12	서울
P2	볼트	녹	17	대전
P3	스크루	청	17	대구
P4	스크루	적	14	서울
P5	캠	청	12	대전
P6	콕	적	19	서울

납품

납품 번호	공급자 번호	부품 번호	수량
D1	S1	P1	300
D2	S1	P2	200
D3	S1	P3	400
D4	S1	P4	200
D5	S1	P5	100
D6	S1	P6	100
D7	S2	P1	300
D8	S2	P2	400
D9	S3	P2	200
D10	S4	P2	200
D11	S4	P4	300
D12	S4	P5	400

데이터 조작용어_데이터 검색 SELECT문

```
SELECT 공급자이름
FROM 공급자
WHERE 공급자번호 IN
(SELECT 공급자번호
FROM 납품
WHERE 부품번호='P2');
```

```
SELECT 공급자이름
FROM 공급자
WHERE EXISTS
(SELECT *
FROM 납품
WHERE 공급자번호=공급자.공급자번호
AND 부품번호='P2');
```

```
SELECT 공급자.공급자이름
FROM 공급자, 납품
WHERE 공급자.공급자번호=납품.공급자번호
AND 납품.부품번호='P2';
```

공급자				납품			
공급자 번호	공급자 이름	거래 포인트	도시	납품 번호	공급자 번호	부품 번호	수량
S1	이승주	20	서울	D1	S1	P1	300
S2	이동일	10	대전	D2	S1	P2	200
S3	이상현	30	대전	D3	S1	P3	400
S4	소병식	20	서울	D4	S1	P4	200
S5	황대섭	30	광주	D5	S1	P5	100
				D6	S1	P6	100
				D7	S2	P1	300
				D8	S2	P2	400
				D9	S3	P2	200
				D10	S4	P2	200
				D11	S4	P4	300
				D12	S4	P5	400

데이터 조작어_데이터 삽입 INSERT문

INSERT

INTO 테이블[(열이름{,열이름}*)]

VALUES (열값_리스트);

INSERT

INTO 부품(부품번호, 창고소재지, 무게)

VALUES ('P7', '세종', 24);

부품

부품 번호	부품 이름	색상	무게	창고 소재지
P1	너트	적	12	서울
P2	볼트	녹	17	대전
P3	스크루	청	17	대구
P4	스크루	적	14	서울
P5	캠	청	12	대전
P6	콕	적	19	서울
P7			24	세종

데이터 조작어_DELETE문

DELETE

FROM 테이블
[WHERE 조건];

DELETE
FROM 공급자
WHERE 도시='광주';

DELETE
FROM 공급자;

DROP TABLE 공급자;

공급자

공급자 번호	공급자 이름	거래 포인트	도시
S1	이승주	20	서울
S2	이동일	10	대전
S3	이상현	30	대전
S4	소병식	20	서울
S5	황대섭	30	광주

데이터 조작어_데이터 갱신 UPDATE문

UPDATE 테이블

SET 열이름 = 산술식[, 열이름=산술식]*

[WHERE 조건];

```
UPDATE 부품
SET 색상='황'
WHERE 부품번호='P2';
```

부품

부품 번호	부품 이름	색상	무 게	창고 소재지
P1	너트	적	12	서울
P2	볼트	황	17	대전
P3	스크루	청	17	대구
P4	스크루	적	14	서울
P5	캠	청	12	대전
P6	콕	적	19	서울

뷰

■ 하나 이상의 기본 테이블로부터 유도되어 만들어진 가상 테이블

- 뷰 내용은 물리적으로 구현되어 실제로 존재하는 것이 아니라 뷰에 대한 조작을 요구할 때마다 기본 테이블의 데이터를 이용해서 내용을 만들어서 사용자에게는 있는 것처럼 보여줌
- 데이터 검색은 기본 테이블에서처럼 동등한 연산으로 수행
 - 모든 뷰가 갱신할 수 있는 것은 아님 → 기본 키의 유지 여부에 따라 결정됨

■ 기본 테이블을 들여다보는 “**유리창**”

뷰의 생성과 제거

CREATE VIEW 공급자2

AS SELECT 공급자번호, 거래포인트

FROM 공급자

WHERE 거래포인트 > 15;

→ 열의 부분집합으로 원하는 뷰를 생성

AS SELECT문은 일반 검색문의 사용과 동일
단, ORDER BY 사용 불가

DROP VIEW 공급자2;

기본 테이블이 제거되면

이를 기초로 만들어진 인덱스/뷰도 자동적으로 제거

뷰의 장단점

장점

- DB 재구성(확장, 구조변경)면에서 어느 정도의 논리적 데이터 독립성을 제공
- 동일 데이터에 대해 동시에 여러 사용자에게 다양한 뷰를 제공
- 특정 사용자가 관심있는 데이터에만 초점을 맞추고 나머지는 무시함
 - 사용자의 데이터에 대한 인식과 관리를 단순화시킴
- 감춰진 데이터에 대해 보안이 자동으로 제공됨

단점

- 독자적인 인덱스를 가질 수 없고,
정의 변경 불가, 삽입/삭제/갱신 연산에 제약이 따름



1. 관계형 모델

- 2차원테이블 → 릴레이션, 속성, 튜플, 영역, 카디널리티, 차수
- 릴레이션의 개념과 특징, 키의 개념과 종류, 제약 조건

2. 데이터베이스 설계

- 사용자요구사항분석 - 개념적 설계 - 논리적 설계 - 물리적 설계 - 구현
- 논리적모델링 → ERD로부터 관계 데이터 모델로의 변환 과정

3. SQL

- SQL → 종합적인 언어, 대화식/부속어사용, 테이블/레코드/열 용어선호, 비절차적언어
- 데이터정의어 → 생성 CREATE, 변경 ALTER, 삭제 DROP
- 데이터조작어 → 검색 SELECT, 삽입 INSERT, 삭제 DELETE, 갱신 UPDATE
- 뷰 → 개념, 생성 및 제거, 장단점

15

강

다음시간 안내

컴퓨터 네트워크

Thank You~



KOREA NATIONAL OPEN UNIVERSITY

