

SQL 첫걸음

- 1강. 데이터베이스와 SQL
- 2강. 테이블에서 데이터 검색
- 3강. 정렬과 연산
- 4강. 데이터 추가, 삭제, 갱신
- 5강. 집계와 서브쿼리
- 6강. 데이터베이스 객체 작성 / 삭제
- 7강. 복수의 테이블 다루기
- 8강. 데이터베이스 설계

1강. 데이터베이스와 SQL

DB 정리되어 저장된 데이터의 집합

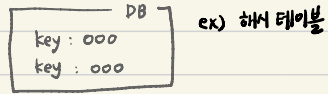
DBMS DB를 관리하는 소프트웨어 → 데이터 검색·추가·삭제·갱신

SQL DB 중, 관계형 DBMS를 조작하는 언어 SQL을 포함 → SQL

- ① DML (Data Manipulation Language) → 데이터 조작 : select, insert, update, delete
- ② DDL (Data Definition Language) → DB 정의 (테이블 구조) : create, alter, drop, rename
- ③ DCL (Data Control Language) → DB 접근권한 제어 : grant, revoke
- ↳ TCL : commit, rollback

DB의 종류

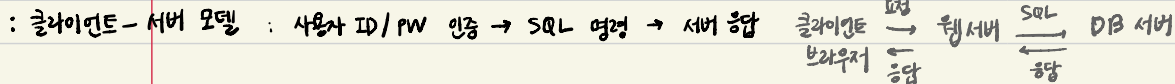
- 1) 계층형 DB 폴더 > 파일 등의 계층구조 (현재 DBMS 구조 채택↓)
- 2) 관계형 DB 행·열의 표 형식 과차원 데이터를 저장하는 형태의 DB
- 3) 객체지향 DB 객체 (object) 그대로를 저장하는 DB - 객체지향 언어 Java, C++
- 4) XML DB XML 형식의 데이터를 저장하는 DB - 전용명령어 XQuery
- 5) key-Value Store 키 : 값 구조 형태의 데이터를 저장하는 DB



RDBMS 제품

- Oracle 오라클 개발, RDBMS의 사실상 표준
- SQL Server MS 개발, 윈도우에서만 동작
- MySQL 오픈소스 커뮤니티 개발, 경량 DB
- ⋮

DB 서버



select 열 from Sample 21 WHERE 조건식 ;

↳ 열을 선택

↳ 행을 선택

2장. 테이블에서 데이터 검색

문자의 자료형 → 자료형에 맞게 표기한 상수값 'literal'

- ① Integer 정수형, 소수점 x 수식형
- ② CHAR 고정된 문자열 → 공백문자로 나머지 채움 문자열형
- ③ VARCHAR 가변 문자열 문자열형
- ④ DATE
- ⑤ TIME

수식형 조건식 WHERE no = 2 ;
 문자열형 조건식 WHERE name = '박준용' ;

NULL 검색 < WHERE no = NULL (X) : 연산자 사용 불가
 WHERE no IS NULL (O)

조건 사용시 유의점

- 1) < WHERE no = 1 OR 2 (X) → 상수 '2'는 언제나 참. 모든 행 반환
 WHERE no = 1 OR no = 2 (O)
- 2) 우선순위 AND > OR

↳ WHERE 행 LIKE '%SQL%';

패턴 매칭 (부호) 검색 : LIKE 연산자

- % SQL % → 중간에 SQL 들어가는 / 시작하는 / 끝나는 문자열
- % SQL → SQL로 끝나는 문자열만
- SQL % → SQL로 시작하는 문자열만

[% : 임의의 문자열 (빈 문자열도 매칭)
 _ : 임의의 문자 1개

이스케이프

WHERE 행 LIKE '% \ % %'

↓
일반문자로 취급



3장. 정렬과 연산

WHERE 조건식 ORDER BY 열명 ex) order by age

(오름차순)

- [문자열형 : 사전식 (1, 10)
 - [수식형 : 대소관계 (1, 2)
- ASC 오름차순 1 → 10
 DESC 내림차순 10 → 1

NULL 값

오름차순

- [가장 먼저 표시 : SQL Server, My SQL (가장 작은 값으로 간주)
- [가장 마지막에 표시 : Oracle (가장 크다고 간주)

결과 행 제한하기

LIMIT (표준 SQL x MySQL에서 지원)

SELECT 열명 FROM 테이블명 WHERE 조건식 ORDER BY 열명 **LIMIT** 행수

TOP → SQL Server 동일기능

SELECT **TOP 3** 열명 FROM 테이블명

ROWNUM → Oracle

SELECT * FROM 테이블명 WHERE ROWNUM <= 3;

↳ Where 질: 정렬하기 전 처리되므로 **LIMIT** 과 결과값 다름

열의 별명 ALIAS (생략 가능)

열명 SELECT price * quantity **AS** amount FROM 테이블명;

↳ 한글 SELECT price * quantity **AS** "금액" FROM 테이블명;

↳ DB 객체명 ↔ 문자열상수 (바운딩) (ABC)

SELECT 구에서 지정한 별명은 WHERE 절에서 사용 x

서버처리 순서: WHERE → GROUP BY → HAVING → SELECT → ORDER BY

↑
여기서 별명 지정

연산자 연산 : +, -, *, /, %

함수 연산 함수명 (인수 1, 인수 2 ...)

1) 수치형

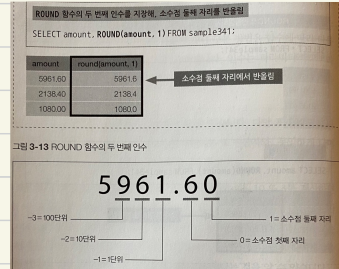
2) 문자열형

ex) 문자열 결합

SQL Server +

Oracle ||

MySQL CONCAT (행 1, 행 2)



날짜연산

문자열 데이터 → 날짜형 데이터 변환: TO_DATE ('2014/01/25', 'YYYY/MM/DD')

SELECT, WHERE, ORDER BY 등에서 사용 가능

ex) SELECT AVG (CASE WHEN quantity IS NULL THEN 0 ELSE quantity END) FROM 테이블;

CASE 문으로 데이터 변환하기

① 단순 CASE
 SELECT CASE a
 WHEN 1 THEN '남자'
 WHEN 2 THEN '여자'
 ELSE '기타'
 END (AS) 별칭 FROM 테이블명;

② 검색 CASE
 SELECT CASE
 WHEN a=1 THEN '남자'
 WHEN a=2 THEN '여자'
 ELSE '기타'
 END AS 성별 FROM 테이블명;

NULL 값 비교 불가 a = NULL (X)

NULL 값 비교 가능 WHEN a IS NULL THEN '없음'

COALESCE 함수 : NULL이 아닌 최초의 표현식 return

(v. 합용하다. 연합하다)

COALESCE (a, 0)

(a가 NULL이 아니면 a 그대로 출력
 a가 NULL이면 0 출력)



일반형 함수	함수 설명
NVL(표현식1, 표현식2) / ISNULL(표현식1, 표현식2)	표현식1의 결과값이 NULL이면 표현식2의 값을 출력한다. 단, 표현식1과 표현식2의 결과 데이터 타입이 같아야 한다. NULL 관련 가장 많이 사용되는 함수이므로 상당히 중요하다.
NVLIF(표현식1, 표현식2)	표현식1이 표현식2와 같으면 NULL을, 같지 않으면 표현식1을 리턴한다.
COALESCE(표현식1, 표현식2,)	임의의 개수 표현식에서 NULL이 아닌 최초의 표현식을 나타낸다. 모든 표현식이 NULL이라면 NULL을 리턴한다.

4장 데이터 추가, 삭제, 갱신 : DML (Data Manipulation)

- ① 데이터 추가 INSERT (행 값 추가)
- ② 데이터 갱신 UPDATE (셀 단위로 갱신)
- ③ 데이터 삭제
 - 1) 물리삭제 - DELETE (행 단위로 삭제)
 - 2) 논리삭제 → 삭제한 것으로 가정, 삭제플래그 SELECT

① 데이터 추가

- 1) 전체 행 추가 INSERT INTO 테이블명 VALUES (값1, 값2, ...)
- 2) 일부 열만 추가 INSERT INTO 테이블명 (열1, 열2) VALUES (값1, 값2) (4개는 Default 인 NULL로 입력)

② 데이터 갱신 - UPDATE 테이블명 SET 열명1 = 값1, 열명2 = 값2, ... WHERE 조건식 (바꾸기)

↑
 '열1에 값1을 넣으라' 테이블명상!
 비교연산자 아님

(Oracle : SET 구기 가능한 순서 무관
 MySQL : SET 구기 가능한 순서대로 처리)

SELECT 구, FROM 구, WHERE 구 등

서브쿼리 - 상부가 아닌 하부의 부수적 질의

ex) DELETE 명령의 WHERE 구에서 사용 (a 열이 가장 작은 행 삭제)

DELETE FROM 테이블 WHERE a = (SELECT MIN(a) FROM 테이블);

서브쿼리의 패턴

① 하나의 값 반환 = 스칼라 값 → SELECT 구의 서브쿼리는 '스칼라 서브쿼리' 여야 함

MIN(a)
80

② 복수의 행 / 하나의 열 반환 →

no
1
2
3

③ 하나의 행 / 복수의 열 반환

MIN(a)	MIN(b)
80	4

④ 복수의 행 / 복수의 열 반환

no	a
1	100
2	900
3	80

상관 서브쿼리 ⊕ EXISTS 조항 / NOT EXISTS

부도 명령문과 반환된 행이 있으면 T / 없으면 F 반환

자식 서브쿼리가 특정 관계를 맺는 서브쿼리

UPDATE 테이블 1 SET a = '있음' WHERE

EXISTS (SELECT * FROM 테이블 2 WHERE 테이블 1.no = 테이블 2.no 2);

열명 IN (집합) < 집합 안에 존재하면 T
" " x F

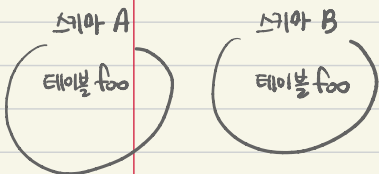
6장. 데이터베이스 객체 작성 / 삭제

테이블, 뷰, 인덱스 등 DB 내 정의하는 모든 것 ↔ 객체지향 프로그래밍 (자바, C++) 의 '객체' (상체 有)

'스키마'라는 그릇 안에 생성됨

스키마 정의 DDL (Data Definition Language)

: CREATE, ALTER, DROP, RENAME



CREATE VIEW 뷰명 ~

1) 테이블 작성 CREATE TABLE 테이블명 열명 자료형 (DEFAULT 기본값) (NULL / NOT NULL)
CREATE TABLE 학생 (no INTEGER NOT NULL, a VARCHAR (30), b DATE);

2) 테이블 삭제 DROP TABLE 테이블명

cf. 테이블 정의는 유지. 데이터만 삭제 DELETE (행 지정 가능)
cf. " TRUNCATE (행 지정 불가. 모든 데이터 삭제)
v. 길이를 줄인다

3) 테이블 변경 ALTER TABLE 테이블명 하부명령

① 열 변경 (추가·삭제·변경)

열 추가 ALTER TABLE 테이블명 ADD 열명 INTEGER;
속성 변경 ALTER TABLE 테이블명 MODIFY 열명 VARCHAR (20);
이름 변경 ALTER TABLE 테이블명 CHANGE 기존열명 신규열명 신규자료형 → MySQL
RENAME TO → 외글
열 삭제 ALTER TABLE 테이블명 DROP 열명;

② 제약 변경 (추가·삭제)

제약 추가 (열 ALTER TABLE 테이블명 MODIFY 열명 VARCHAR (30) NOT NULL;
테이블 ALTER TABLE 테이블명 ADD CONSTRAINT 제약명 PRIMARY KEY (no);
제약 삭제 (열 ALTER TABLE 테이블명 MODIFY 열명 VARCHAR (30);
테이블 ALTER TABLE 테이블명 DROP CONSTRAINT 제약명;
DROP PRIMARY KEY;

제약 - 테이블에 저장된 데이터를 제한

NOT NULL / UNIQUE
기본키 (Primary key)
외부참조 정합 제약

CREATE TABLE 테이블명 (
no INTEGER NOT NULL
name VARCHAR (20),
PRIMARY KEY (no)
);

기본키 제약 [NOT NULL 설정된 열
중복 데이터 입력, x

제약에 이름 붙이기
CONSTRAINT 제약명 PRIMARY KEY (no)

(색인)
인덱스 구조 - 데이터 검색시 쓰이는 키워드·대응하는 데이터 행의 장소 저장 테이블에 의존하는 별도의 DB 객체

인덱스에 쓰이는 검색 알고리즘
 1) 이진 트리 (binary tree)
 2) 해시

1) 이진 트리 (binary tree) by 이진 탐색 ↔ Full Table Scan
 데이터가 차례로 정렬되어야 함 · 집합을 반씩 나누어 조사
 · 인덱스 X
 · 테이블의 모든 값을 차례로 조사

테이블에 인덱스 작성 → 인덱스용 데이터 생성

인덱스 작성과 삭제 → 이 테이블의 어느 필드에 대한 인덱스?

작성 CREATE INDEX 인덱스명 ON 테이블명 (열1, 열2, ...)
 삭제 DROP INDEX 인덱스명 (ON 테이블명)
 (인덱스가 테이블 내 객체인 경우)

뷰 작성과 삭제

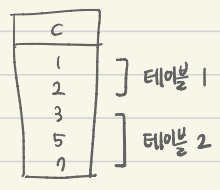
↳ 본래 객체로 등록할 수 없는 SELECT 명령을, 객체로서 이름을 붙여 관리 (테이블처럼 활용) = 가상 테이블
 ↓ 열재질 가능

작성 CREATE VIEW 뷰명 AS (SELECT 명령) / SELECT VIEW 뷰명 (n, v, s) AS
 삭제 DROP VIEW 뷰명
 SELECT no, a, a*2 FROM 테이블a;

1장. 복수의 테이블 다루기

집합 연산 UNION - SELECT 명령의 실행결과를 합집합으로 계산 (중복값 제거)

① 합집합
 SELECT a AS c FROM 테이블 1
 UNION
 SELECT b AS c FROM 테이블 2
 ORDER BY c;



조인을 사용하지 않고
 2개 이상 테이블 조회

UNION ALL (중복값 유지)

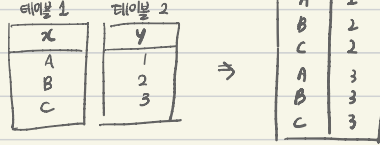
A → ALIAS는 A
 UNION ALL 모두를 여쭙
 B

② 교집합 INTERSECT (오래클은 MINUS)
 ③ 차집합 EXCEPT

• 테이블 결합 (Join)

① 교차 결합 Cross Join = 곱집합 개념

```
SELECT * FROM 테이블 1, 테이블 2
```



```
EX) SELECT A. 상품명, B. 재고수 FROM A, B
WHERE A. 상품코드 = B. 상품코드;
```

→ 귀찮 / 사용 X

```
SELECT A. 상품명, B. 재고수 FROM A CROSS JOIN B
```

② 내부 결합 INNER JOIN

```
EX) SELECT A. 상품명, B. 재고수
FROM A INNER JOIN B ON A. 상품코드 = B. 상품코드;
```

행이 A, B 모두 존재해야
결과에 포함됨

③ 외부 결합

< LEFT JOIN : A가만 있고 B는 NULL 인 값 포함
RIGHT JOIN : A가 없고 B가만 있는 값 포함

결합 테이블 중
조인 기준을 결정할 수 있음

8장. 데이터베이스 설계

- ERD Entity Relationship Diagram : '설계상 이렇게 연관되어 있다'
- 정규화 DB 설계 단계에서 테이블을 분할 → 데이터가 한 곳에만 저장되도록!
 - ① 제 1 정규화 : 반복되는 부분을 찾아 테이블 분할
 - ② 제 2 정규화 : 부분 함수 종속성을 찾아 테이블 분할 (PK에서 중복 제거)
 - ③ 제 3 정규화 : PK가 아닌 일반 컬럼에서 중복 제거

TCL Transaction Control Language : COMMIT, ROLL BACK

AUTO COMMIT : insert, update, delete 처리시 자동 커밋
트랜잭션 COMMIT

MySQL

```
START TRANSACTION;
INSERT INTO 주문 VALUES (4, '2014-03-01', 1);
COMMIT;
```

SQL Server : BEGIN TRANSACTION
Oracle : x