

# 데이터베이스

## 강의 노트

제 7 회차  
SQL 언어

## ❖ 학습목표

- SQL의 특징을 나열할 수 있다.
- SQL로 기본 테이블을 생성하고 삭제할 수 있다.
- SQL 데이터 제어어의 명령어 종류를 나열할 수 있다.
- SQL로 데이터를 검색할 수 있다.
- SQL로 데이터를 삽입하거나 변경할 수 있다.
- SQL 뷰를 생성하고 삭제할 수 있다.

## ❖ 학습내용

- SQL 데이터 정의어와 데이터 제어어
- SQL 데이터 조작어와 SQL 뷰

## SQL 데이터 정의어와 데이터 제어어

1. SQL 언어 개요
2. SQL 데이터 정의어
3. SQL 데이터 제어어

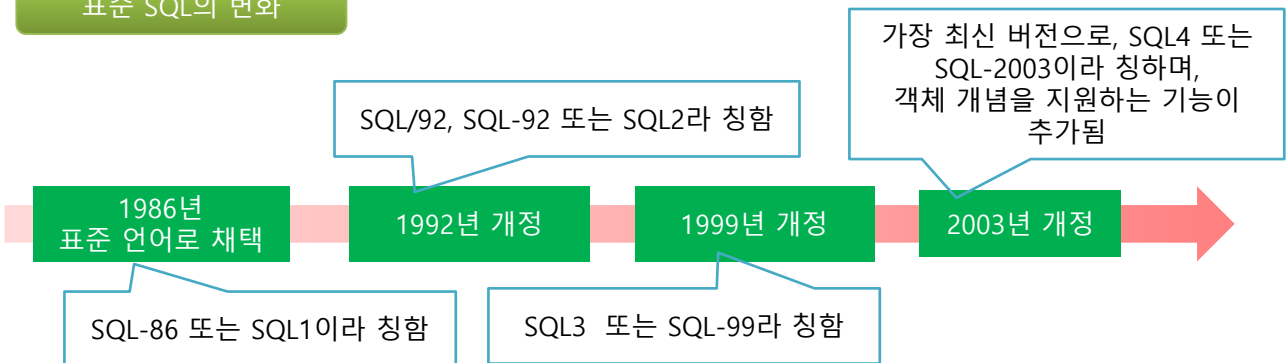
## 1. SQL 언어 개요

### 1) SQL 언어 개요

#### SQL이란?

- "Structured Query Language(구조화된 질의어)"의 약자로, 1974년 IBM 연구소에서 발표한 [SEQUEL\(Structured English QUery Language\)](#)에서 유래한 데이터 언어
- 현재 미국표준연구소(ANSI)와 국제표준기구(ISO)에서 관계 DB의 표준 언어로 채택
- 상용 DBMS인 IBM의 DB2와 SQL/DS뿐만 아니라, ORACLE, INFORMIX, SYBASE 등과 같은 거의 모든 회사에서 채택하고 있는 언어

#### 표준 SQL의 변화



## 1. SQL 언어 개요

### 2) SQL의 특징

단순히 검색만을 위한 데이터 질의어가 아니라, 데이터 정의어(DDL), 데이터 조작어(DML), 데이터 제어어(DCL)의 기능을 모두 제공하는 종합적인 DB 언어

관계 데이터 모델의 데이터 연산 언어인 관계 대수와 관계 해석을 기초로 하는 고급 데이터 언어

데이터 연산에 대한 처리 과정은 명시하지 않고, 데이터 연산의 결과만을 명시하는 비절차적 언어 즉, 선언적 언어이므로, 사용자 편의 중심의 언어

SQL의 표준화로 인해 상용 관계 DBMS들 간의 전환이 용이

단말기(Terminal)에서 단독으로 대화식 질의어로 사용

자바, C/C++ 등으로 개발된 응용 프로그램에 삽입해서 사용

개개의 레코드(튜플) 단위로 처리하기 보다는 레코드의 집합 단위로 처리

릴레이션, 튜플, 속성 등과 같은 관계 데이터 모델의 공식 용어 대신, 테이블, 행, 열과 같은 일반적인 용어를 사용

## 2. SQL 데이터 정의어

### 1) SQL 언어 개요

#### SQL 데이터 정의어(DDL: Data Definition Language)란?

DB 객체의 구조를 정의하거나 변경 또는 삭제하기 위해 사용하는 언어

#### DB 객체란?

생성된 DB를 효율적으로 관리하고 동작하게 만드는 요소로서,  
논리적인 데이터 저장 구조이며, 스키마 객체라고도 칭함

### 2) DB객체의 종류

#### 테이블(Table)

행과 열로 구성된 기본적인 데이터 저장 영역의 기본 단위

#### 뷰(View)

하나 이상의 테이블 또는 다른 뷰로부터 유도된 논리적 관점에서의  
가상 테이블

#### 시퀀스(Sequence)

자동으로 생성되는 유일한(Unique) 일련 번호로서,  
대개 기본 키 값을 생성하는데 사용

#### 인덱스(Index)

DB 검색의 성능 향상을 위해서 만들어지는 데이터 구조

#### 동의어(Synonym)

DB 객체의 복잡한 이름을 간단하게 해주기 위해 부여하는 별칭

## 2. SQL 데이터 정의어

### 3) 데이터 정의어의 명령어 종류와 테이블의 종류

#### 데이터 정의어의 명령어 종류

DDL 명령어	역할
CREATE	DB(또는 스키마) 및 DB 객체를 생성함
ALTER	기존의 DB 객체를 변경함
DROP	DB(또는 스키마) 및 DB 객체를 삭제함

#### 테이블의 종류

##### 기본 테이블 (Base Table)

- CREATE TABLE **명령**으로 만들어지는 테이블
- DBMS의 파일로 생성되고 저장됨

##### 가상 테이블 (Virtual Table)

- CREATE VIEW **명령**으로 만들어지는 테이블
- 어떤 기본 테이블로부터 만들어지는 테이블
- 독자적으로 존재할 수 없음

##### 임시 테이블 (Temporary Table)

- 질의문 처리 과정의 중간 결과로 만들어지는 테이블

## 2. SQL 데이터 정의어

### 4) 기본 테이블 생성

#### CREATE TABLE 명령

새로운 테이블을 생성하는 명령

여러 가지 무결성 제약조건을 함께 지정할 수 있음

테이블 생성 권한을 가진 사용자만 사용할 수 있는 명령

#### CREATE TABLE 명령의 일반 형식

```
CREATE TABLE 기본테이블_이름
  ({열이름 데이터타입 [NOT NULL] [DEFAULT 값]}+
  [PRIMARY KEY (열이름_리스트)],
  {[UNIQUE (열이름_리스트)]}*
  {[FOREIGN KEY(열이름_리스트)
  REFERENCES 참조테이블_이름[(열이름_리스트)]
  [ON DELETE 옵션]
  [ON UPDATE 옵션] ,]}*
  [CONSTRAINT 제약조건이름] [CHECK(조건식)] );
```



#### SQL 명령의 형식을 기술할 때 사용하는 기호들

- 1) **중괄호({ })** : 값의 리스트(List), 즉 **여러 값을 나열**해야 함을 의미함
- 2) **대괄호([ ])** : **생략 가능함**(Optional)을 의미함
- 3) **+** : 괄호 안의 내용이 **1번 이상 반복**될 수 있음을 의미함. 즉, 최소 1번은 나타나야 함을 의미함
- 4) **\*** : 괄호 안의 내용이 **0번 이상 반복**될 수 있음을 의미함. 즉, 나타나지 않을 수도 있음을 의미함
- 5) **수직바(|)** : 둘 중 하나를 선택하라는 '**OR**'를 의미함



## 2. SQL 데이터 정의어

### 4) 기본 테이블 생성

#### 각 키워드의 기능

NOT NULL	속성이 널 값을 가질 수 있는지 여부를 설정함		
DEFAULT	속성의 디폴트 값을 설정함		
PRIMARY KEY	기본 키(개체 무결성 제약조건)를 설정함		
UNIQUE	속성이 유일한 값을 가져야 한다는 무결성 제약조건을 설정함		
FOREIGN KEY	외래 키(참조 무결성 제약조건)를 설정함 ※ SQL에서는 참조된 테이블의 후보 키도 외래 키가 될 수 있음		
옵션	NO ACTION, SET NULL, SET DEFAULT, CASCADE 가운데 하나를 설정함		
	예	ON DELETE SET NULL	ON UPDATE CASCADE
		피참조 튜플(열 값)이 삭제되면, 널 값으로 갱신하도록 설정함	피참조 튜플(열 값)이 갱신되면, 이 튜플을 참조하는 튜플도 파급적으로 갱신하도록 설정함
CONSTRAINT	제약조건의 이름을 설정함		
CHECK(조건식)	삽입 또는 갱신 시에 만족해야 하는 무결성 제약조건을 설정함		

#### 대표적인 데이터 타입

NUMBER(p, s)	소수점을 포함해서 20자리까지의 숫자 기억
INT, INTEGER, SMALLINT	정수 기억
CHAR(n)	고정 길이 문자열 기억
VARCHAR(n), VARCHAR2(n)	가변 길이 문자열 기억
DATE	YY/MM/DD 형식의 날짜 기억
TIME	hh:mm:ss 형식의 시간 기억
LONG	최대 2기가 바이트까지의 가변 길이 문자 기억
LOGICAL	논리 값(T/F) 기억

## 2. SQL 데이터 정의어

### 4) 기본 테이블 생성

#### 기본 테이블 생성 예제

##### 사원

```
CREATE TABLE 사원
(
  사번    INTEGER,
  이름    CHAR(10) NOT NULL,
  입사일  DATE NOT NULL,
  호봉    INTEGER,
  휴대폰  CHAR(13),
  PRIMARY KEY(사번),
  UNIQUE(휴대폰) CONSTRAINT 사원_unique_key);
```

#### 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999

## 2. SQL 데이터 정의어

### 4) 기본 테이블 생성

#### 기본 테이블 생성 예제

##### 부서

```
CREATE TABLE 부서
( 부서번호 INTEGER,
  부서명 CHAR(6) NOT NULL,
  부서장 INTEGER,
  사무실 CHAR(4),
  전화번호 CHAR(12),
  PRIMARY KEY(부서번호),
  UNIQUE(부서명, 사무실, 전화번호) );
```

#### 부서 테이블

부서번호	부서명	부서장	사무실	전화번호
100	기획실	1101	A402	02-111-2222
200	비서실	1102	A501	02-222-3333
300	총무부	1201	B311	041-333-4444
400	인사부	1203	B201	041-444-5555
500	자재부		A102	02-555-6666

## 2. SQL 데이터 정의어

### 4) 기본 테이블 생성

#### 기본 테이블 생성 예제

##### 근무

```
CREATE TABLE 근무
(
    사원번호 INTEGER,
    부서번호 INTEGER,
    직책 CHAR(8),
    PRIMARY KEY(사원번호, 부서번호),
    FOREIGN KEY(사원번호) REFERENCES 사원(사번)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY(부서번호) REFERENCES 부서(부서번호)
        ON DELETE CASCADE
        ON UPDATE CASCADE );
```

#### 근무 릴레이션

사원번호	부서번호	직책
1101	100	부장
1102	200	과장
1103	200	대리
1201	300	부장
1202	300	대리
1203	400	과장
1312	100	사원
1314	400	사원

## 2. SQL 데이터 정의어

### 5) 기본 테이블 삭제

#### DROP TABLE 명령

기존의 기본 테이블을 삭제하는 명령

테이블 구조와 함께 저장된 데이터와 인덱스까지 한꺼번에 삭제

테이블을 생성한 사용자 또는 테이블 삭제 권한이 부여된 사용자만 사용할 수 있는 명령

삭제 후 복구가 불가능하므로 신중하게 사용

#### DROP TABLE 명령의 일반 형식

```
DROP TABLE 기본테이블_이름  
{ RESTRICT | CASCADE };
```

이 기본 테이블을 참조하고 있는 뷰나 제약조건이 있으면, 이 기본 테이블을 삭제할 수 없음

이 기본 테이블을 참조하고 있는 뷰나 제약조건이 있으면, 그 뷰나 제약조건도 함께 모두 삭제함

## 2. SQL 데이터 정의어

### 6) 기본 테이블 변경

#### ALTER TABLE 명령

기존의 기본 테이블의 구조를 변경하는 명령

구조 변경 허용 범위

- ① 새로운 열을 추가할 수 있음 (ADD 절)
- ② 기존 열을 삭제할 수 있음 (DROP 절) (열 삭제는 오라클 8i 버전부터 지원함)
- ③ 기존 열의 데이터타입을 변경할 수 있음 (MODIFY 절)
- ④ 열에 대한 디폴트 값을 삭제하거나 새로 정의할 수 있음 (MODIFY 절)
- ⑤ 새로운 무결성 제약조건을 명시하거나 기존 무결성 제약조건을 삭제할 수 있음

#### ALTER TABLE 명령의 일반 형식

**ALTER TABLE** 기본테이블\_이름

([**ADD** 열이름 데이터타입] [**DEFAULT** 값] |

[**DROP** 열이름] [무결성 제약조건] [**RESTRICT** | **CASCADE**]

[**MODIFY** 열이름 [데이터타입] (**DROP DEFAULT** | **SET DEFAULT** 값));

이 열을 참조하고 있는 뷰나  
제약조건이 있으며, 이 열을 삭제할  
수 없음

이 열을 참조하고 있는 모든 뷰나  
제약조건을 함께 삭제함

## 2. SQL 데이터 정의어

### 6) 기본 테이블 변경

#### 기본 테이블 변경 예제

##### 새로운 열 추가하기

사원 테이블에 최대 10 바이트의 가변 길이 문자열 타입의 '최종학력' 열을 추가하고, 디폴트 값을 '대졸'로 설정함

```
ALTER TABLE 사원
    ADD 최종학력 VARCHAR2(10) DEFAULT '대졸';
```

##### 기존 열 삭제하기

사원 테이블의 '호봉' 열을 삭제하되, 이 열을 참조하는 뷰나 제약조건이 있으면 함께 삭제하도록 함

```
ALTER TABLE 사원
    DROP 호봉 CASCADE;
```

##### 기존 열의 데이터 타입 변경하기

사원 테이블의 '이름' 열의 데이터 타입을 20바이트의 가변 길이 문자열로 변경함

```
ALTER TABLE 사원
    MODIFY 이름 VARCHAR2(20);
```

## 2. SQL 데이터 정의어

### 6) 기본 테이블 변경

#### 기본 테이블 변경 예제

#### 열의 디폴트 값 변경하기

사원 테이블의 '최종학력' 열의 디폴트 값을 '고졸'로 수정함

```
ALTER TABLE 사원
    MODIFY 최종학력 SET DEFAULT '고졸';
```

#### 무결성 제약조건 삭제하기

사원 테이블의 '휴대폰' 열이 유일한 값을 갖도록 UNIQUE 옵션을 설정할 때 설정한 '사원\_unique\_key'라는 이름의 제약조건을 삭제함

```
ALTER TABLE 사원
    DROP CONSTRAINT 사원_unique_key;
```



### 3. SQL 데이터 제어어

#### 1) SQL 데이터 제어어

*데이터 제어어(DCL: Data Control Language)란?*

DB 사용자의 권한을 제어하기 위해서 사용하는 언어

데이터 제어어의 명령어 종류

GRANT

사용자에게 DB 객체에 대한 권한을 부여

REVOKE

사용자에게 이미 부여된 DB 객체에 대한 권한을 취소

#### 2) 권한 부여

GRANT 명령

DB 관리자가 한 명 이상의 사용자에게 여러 개의 시스템 권한을 부여할 때 사용하는 명령

GRANT 명령의 일반 형식

**GRANT** 시스템권한1[, 시스템권한2, ...]

**TO** 사용자명1[, 사용자명2, ...];

### 3. SQL 데이터 제어어

#### 2) 권한 부여

응용 프로그램 개발자가 가질 수 있는 시스템 권한의 종류

CREATE SESSION	사용자가 DB로 접속할 수 있는 권한
CREATE TABLE	사용자의 스키마에서 테이블을 생성할 수 있는 권한
CREATE SEQUENCE	사용자의 스키마에 시퀀스를 생성할 수 있는 권한
CREATE VIEW	사용자의 스키마에서 뷰를 생성할 수 있는 권한
CREATE PROCEDURE	사용자의 스키마에 내장된 프로시저, 함수, 패키지를 생성할 수 있는 권한

#### 권한 부여 사례

```
GRANT CREATE TABLE, CREATE SEQUENCE, CREATE VIEW
TO mkipark, yhkim;
```



DB 관리자가 'mkipark' 및 'yhkim' 사용자에게 테이블과 시퀀스, 뷰를 생성할 수 있는 권한을 부여

### 3. SQL 데이터 제어어

#### 3) 권한 철회

##### REVOKE 명령

사용자에게 부여한 권한을 철회할 때 사용하는 명령

##### REVOKE 명령의 일반 형식

```
REVOKE 시스템권한1[, 시스템권한2, ...]  
FROM 사용자명1[, 사용자명2, ...];
```

##### 권한 철회 사례

```
REVOKE CREATE TABLE, CREATE VIEW  
FROM mkpark;
```



DB 관리자가 'mkpark' 사용자에게 부여한 테이블 및 뷰 생성 권한을 철회

## SQL 데이터 조작용어와 SQL 뷰

1. SQL 데이터 조작용어
2. SQL 뷰

## 1. SQL 데이터 조작용어

### 1) SQL 데이터 조작용어

*데이터 조작용어(DML: Data Manipulation Language)란?*

데이터의 검색 및 삽입, 삭제, 변경을 위해서 사용하는 언어

#### 데이터 조작용어의 명령어 종류

SELECT

DB 객체에서 데이터를 검색

INSERT

DB 객체에 데이터를 삽입

DELETE

DB 객체에서 데이터를 삭제

UPDATE

DB 객체의 기존 데이터를 변경



#### SQL의 테이블과 관계 데이터 모델의 릴레이션과의 차이점

- ① SQL의 테이블은 반드시 기본 키를 가져야 하는 것은 아님
- ② SQL의 한 테이블에는 동일한 행(레코드)이 중복될 수 있음  
→ 검색할 때 'DISTINCT' 옵션을 지정하면, 동일한 행을 제거할 수 있음

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령

테이블에서 데이터를 검색할 때 사용하는 명령

검색한 결과도 테이블이 되는 폐쇄적 성질을 갖고 있는 명령

#### SELECT 명령의 일반 형식

```
SELECT [ALL | DISTINCT] { * | 열_리스트 [AS 열 별칭]}
FROM      테이블_리스트
[WHERE     조건]
[GROUP BY 열_리스트
[HAVING   그룹조건]]
[ORDER BY 열_리스트 [ASC | DESC]];
```

⑤  
①  
②  
③  
④  
⑥

5

앞의 모든 조건을 만족하는 열을 검색하기 위해서 SELECT 절이 다섯 번째로 해석됨

1

SELECT 문을 실행할 때, 맨 먼저 검색할 데이터가 어떤 테이블에 있는지를 알아야 하기 때문에, FROM 절이 가장 먼저 해석됨

2

어떤 조건을 만족하는 열을 검색 할지 알아야 하기 때문에, WHERE절이 두 번째로 해석됨

3

그룹에 대한 정보를 알기 위해서 GROUP BY 절이 세 번째로 해석됨

4

그룹에 대한 조건을 알기 위해서 HAVING 절이 네 번째로 해석됨

6

검색된 열을 정렬하는 ORDER BY 절이 맨 마지막에 해석됨

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 구성

구성	기능
SELECT 절	검색할 하나 이상의 데이터 항목 즉, 열 이름이나 열 별칭(Alias)을 기술 ① ALL : 검색 결과에서 중복되는 행을 그대로 표시하도록 지정함 ② DISTINCT : 검색 결과에서 중복되는 행을 삭제하도록 지정함 ③ * : 테이블의 모든 열을 검색하도록 지정하는 와일드카드(Wildcard) 문자임 ④ AS 열 별칭 : 열 이름에 새로운 이름을 부여하는 것으로, AS는 생략해도 됨
FROM 절	검색할 테이블 이름을 기술
WHERE 절	검색할 조건을 기술
GROUP BY 절	그룹 질의를 수행할 열 이름을 기술한다. 즉, 특정 열의 값이 동일한 행들을 그룹핑해서, 각 그룹별로 한 행의 질의 결과를 생성할 때 사용
HAVING 절	GROUP BY 절에 의해 구성된 그룹에 적용할 제한 조건을 기술한다.
ORDER BY 절	특정 열 값을 기준으로 질의 결과를 정렬할 때 기준이 되는 열 이름을 기술 ① ASC : 오름차순(Ascending) 정렬을 지정함 (디폴트 값임) ② DESC : 내림차순(Descending) 정렬을 지정함

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

##### 모든 열 검색하기

사원 테이블의 모든 열의 내용을 검색함

```
SELECT  *
FROM    사원;
```

#### 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999

#### 검색 결과 확인

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-10	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999



## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

#### 조건이 만족하는 모든 열 검색하기

사원 테이블에서 호봉이 11이라는 조건을 만족하는 사원의 모든 정보를 검색함

```
SELECT *
FROM 사원
WHERE 호봉 = 11 ;
```

#### 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999

#### 검색 결과 확인

사번	이름	입사일	호봉	휴대폰
1312	김민수	2011-03-01	11	011-1111-2222
1314	이정숙	2011-03-01	11	010-2222-3333

## 1. SQL 데이터 조작용어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

#### 조건이 만족하는 특정 열만 검색

사원 테이블에서 호봉이 11이라는 조건을 만족하는 사원의 사번과 이름만을 검색함

```
SELECT 사번, 이름
FROM 사원
WHERE 호봉 = 11 ;
```

#### 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999

#### 검색 결과 확인

사번	이름
-----	
1312	김민수
1314	이정숙

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

#### 중복된 행을 제거하고 검색하기

현재 직원들이 근무하는 부서번호가 무엇인지 근무 테이블에서 검색함

```
SELECT DISTINCT 부서번호
FROM 근무;
```

#### 근무 릴레이션

직원번호	부서번호	직책
1101	100	부장
1102	200	과장
1103	200	대리
1201	300	부장
1202	300	대리
1203	400	과장
1312	100	직원
1314	400	직원

#### 검색 결과 확인

부서번호

-----

100  
200  
300  
400

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

#### 행 정렬 검색하기

사원 테이블에서 입사일이 빠른 사람부터 출력하고, 입사일이 같은 경우 사원번호가 더 큰 번호를 갖는 사람부터 출력함

```
SELECT 사번, 이름, 입사일
FROM 사원
ORDER BY 입사일 ASC, 사원번호 DESC;
```

#### 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999

#### 검색 결과 확인

사번	이름	입사일
1201	최강희	1994-02-01
1101	김정아	1995-03-10
1102	이기원	1997-01-01
1203	박제성	1999-09-01
1103	박영종	2005-09-01
1202	조현수	2007-03-15
1314	이정숙	2011-03-01
1312	김민수	2011-03-01

내림차순(DESC) → (사번 1314, 1312)  
 오름차순(ASC) → (입사일 1994-02-01)

## 1. SQL 데이터 조작어

## 2) 데이터 검색

SELECT 명령의 데이터 검색 예제

GROUP BY 절을 이용해서 검색하기

사원\_2 테이블에서 같은 부서에 근무하는 직원들의 평균 연봉을 계산해서 출력함

```
SELECT    부서번호, AVG(연봉) AS 평균연봉
FROM      사원_2
GROUP BY  부서번호;
```

사원2\_ 테이블

사번	이름	입사일	부서번호	연봉	휴대폰
1101	김정아	1995-03-01	100	7000	010-1111-2222
1102	이기원	1997-01-01	200	6000	011-2222-3333
1103	박영종	2005-09-01	200	4500	010-3333-4444
1201	최강희	1994-02-01	300	7500	010-4444-5555
1202	조현수	2007-03-15	300	4000	010-5555-6666
1203	박제성	1999-09-01	400	5500	019-6666-7777
1312	김민수	2011-03-01	100	2000	011-7777-8888
1314	이정숙	2011-03-01	400	2000	010-8888-9999

검색 결과 확인

부서번호	평균연봉
-----	-----
100	4500
200	5250
300	5750
400	3750

## 1. SQL 데이터 조작어

### 2) 데이터 검색

#### SELECT 명령의 데이터 검색 예제

#### HAVING 절을 이용해서 검색하기

사원\_2 테이블에서 같은 부서에 근무하는 직원들의 평균 연봉을 계산해서 출력하되, 평균 연봉이 5000 이상인 부서만 출력함

```
SELECT    부서번호, AVG(연봉) AS 평균연봉
FROM      사원_2
GROUP BY  부서번호
HAVING    AVG(연봉) ≥ 5000 ;
```

#### 사원2\_ 테이블

사번	이름	입사일	부서번호	연봉	휴대폰
1101	김정아	1995-03-01	100	7000	010-1111-2222
1102	이기원	1997-01-01	200	6000	011-2222-3333
1103	박영종	2005-09-01	200	4500	010-3333-4444
1201	최강희	1994-02-01	300	7500	010-4444-5555
1202	조현수	2007-03-15	300	4000	010-5555-6666
1203	박제성	1999-09-01	400	5500	019-6666-7777
1312	김민수	2011-03-01	100	2000	011-7777-8888
1314	이정숙	2011-03-01	400	2000	010-8888-9999

#### 검색 결과 확인

부서번호	평균연봉
-----	-----
200	5250
300	5750

## 1. SQL 데이터 조작용어

### 3) 데이터 삽입

#### INSERT 명령

기존 테이블에서 새로운 데이터(행)를 삽입할 때 사용하는 명령

삽입할 테이블 이름과 열 이름들, 그리고 각 열의 값을 기술

#### INSERT 명령의 일반 형식

새로운 데이터 직접 삽입하기	다른 테이블의 데이터를 복사해서 삽입하기
<pre>INSERT INTO 테이블_이름 [(열이름_리스트)] VALUES (열값_리스트);</pre>	<p>부속 질의문(SELECT 문)을 이용해서 검색한 결과를 복사해서 삽입</p> <pre>INSERT INTO 테이블_이름 [(열이름_리스트)] SELECT 문;</pre>

#### INSERT 명령의 특징

VALUES 절에 열의 값을 명시하므로 한번에 하나의 행만 삽입할 수 있음

열 이름을 명시하지 않으면, 테이블에 정의된 열의 순서대로 값을 나열해야 함

## 1. SQL 데이터 조작어

## 3) 데이터 삽입

## 데이터 삽입 예제

## 새로운 데이터 직접 삽입하기

사원 테이블에 신입 사원 "이동문"의 데이터를 직접 삽입함

```
INSERT INTO 사원(사번, 이름, 입사일, 호봉, 휴대폰)
VALUES (1411, '이동문', 2012-09-01, 8, '010-8766-4889');
```

혹은

```
INSERT INTO 사원
VALUES (1411, '이동문', 2012-09-01, 8, '010-8766-4889');
```

## 삽입 결과 확인

## 사원 테이블

사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999



사번	이름	입사일	호봉	휴대폰
1101	김정아	1995-03-01	20	010-1111-2222
1102	이기원	1997-01-01	18	011-2222-3333
1103	박영종	2005-09-01	15	010-3333-4444
1201	최강희	1994-02-01	21	010-4444-5555
1202	조현수	2007-03-15	14	010-5555-6666
1203	박제성	1999-09-01	17	019-6666-7777
1312	김민수	2011-03-01	11	011-7777-8888
1314	이정숙	2011-03-01	11	010-8888-9999
1411	이동문	2012-09-01	8	010-9999-0000



## 1. SQL 데이터 조작어

### 3) 데이터 삽입

#### 데이터 삽입 예제

다른 테이블의 데이터를 복사해서 삽입하기

사원\_2 테이블의 데이터 가운데 100번 부서에 근무하는 사원의 사번과 이름 열의 값만 복사해서 '기획실\_사원' 테이블에 삽입함

```
INSERT INTO 기획실_사원(사번, 이름)
SELECT 사번, 이름 ;
FROM 사원_2;
WHERE 부서번호 = 100;
```

#### 삽입 결과 확인

##### 사원2 테이블

사번	이름	입사일	부서번호	연봉	휴대폰
1101	김정아	1995-03-01	100	7000	010-1111-2222
1102	이기원	1997-01-01	200	6000	011-2222-3333
1103	박영종	2005-09-01	200	4500	010-3333-4444
1201	최강희	1994-02-01	300	7500	010-4444-5555
1202	조현수	2007-03-15	300	4000	010-5555-6666
1203	박제성	1999-09-01	400	5500	019-6666-7777
1312	김민수	2011-03-01	100	2000	011-7777-8888
1314	이정숙	2011-03-01	400	2000	010-8888-9999



##### 기획실\_사원 테이블

사번	이름
1101	김정아
1312	김민수

## 1. SQL 데이터 조작어

### 4) 데이터 삭제

#### DELETE 명령

테이블에서 기존 데이터(행)를 삭제할 때 사용하는 명령

삭제할 데이터가 포함된 테이블 이름과 삭제 조건을 기술

#### DELETE 명령의 일반 형식

```
DELETE FROM 테이블_이름
[WHERE 조건];
```

#### 데이터 삭제 예제

한 개의 데이터(행) 삭제하기	복수 개의 데이터(행) 삭제하기
<p>사원 테이블에서 사번이 1203인 사원을 삭제한다.</p> <pre>DELETE FROM 사원 WHERE 사번 = 1203;</pre>	<p>사원_2 테이블에서 부서번호가 200인 사원을 모두 삭제한다.</p> <pre>DELETE FROM 사원_2 WHERE 부서번호 = 200;</pre>

## 1. SQL 데이터 조작성

### 5) 데이터 변경

#### UPDATE 명령

테이블에서 기존 데이터(행)를 변경할 때 사용하는 명령

변경할 데이터가 포함된 테이블 이름과 변경할 열 이름, 변경 조건을 기술

#### UPDATE 명령의 일반 형식

```
UPDATE 테이블_이름
SET {열이름 = 산술식}+
[WHERE 조건];
```

#### 데이터 변경 예제

한 개의 데이터(행) 변경하기	복수 개의 데이터(행) 변경하기
<p>사원 테이블에서 사번이 1202인 사원의 호봉을 15 호봉으로 변경한다.</p> <pre>UPDATE 사원 SET 호봉 = 15 WHERE 사번 = 1202;</pre>	<p>사원 테이블에서 호봉이 11인 사원의 호봉을 모두 12로 변경한다.</p> <pre>UPDATE 사원 SET 호봉 = 12 WHERE 호봉 = 11;</pre>

## 2. SQL 뷰

### 1) SQL 뷰

#### 뷰(View)란?

다른 테이블로부터 유도된 가상의 테이블(Virtual Table)로서,  
물리적으로 생성되지 않는 논리적인 테이블

### 2) SQL 뷰의 특징

- 1 논리적인 테이블이므로 실제로 물리적으로 생성되지 않으며, 뷰의 정의만 시스템에 저장됨
- 2 기본 테이블뿐만 아니라 다른 뷰를 기초로 생성할 수 있음
- 3 뷰 자체는 데이터를 직접 포함하지 않지만, 창문 역할을 하는 뷰를 통해서 데이터의 검색과 삽입, 변경이 가능함
- 4 뷰에 대한 검색은 기본 테이블과 거의 유사하지만, 뷰를 통한 삽입, 삭제, 변경은 일부 제약을 받음
- 5 외부 스키마는 뷰와 기본 테이블의 정의로 구성됨
- 6 뷰의 기본이 되는 기본 테이블이 삭제(변경)되면, 뷰도 자동으로 삭제(변경)됨
- 7 뷰는 ALTER 문을 사용해서 변경할 수 없으며, 변경이 필요한 경우 삭제하고 다시 생성해야 함
- 8 뷰는 DROP문을 사용해서 삭제할 수 있음

## 2. SQL 뷰

### 3) 뷰 사용의 장·단점

#### 장점

1

보안을 위해서 DB에 대한 접근을 제한할 수 있음

사용자는 특정 테이블의 데이터 가운데 뷰로 정의된 특정 부분만을 보게 됨

2

복잡한 질의를 단순한 질의로 변환할 수 있음

여러 테이블을 기초로 뷰를 생성하면 테이블 **조인(Join)**이 불필요하게 됨  
즉, 주로 사용하는 정보만을 대상으로 데이터 조작을 수행할 수 있음

3

데이터 독립성을 허용함

테이블이 변경되어도 뷰는 그대로 유지할 수 있으므로, 임시 사용자와 응용 프로그램에 대한 데이터 독립성을 제공할 수 있음

4

동일한 테이블에 대해서 다른 뷰를 생성할 수 있음

조건에 따라 데이터에 접근하는 사용자 그룹을 분류해서, 각각 동일한 테이블의 다른 뷰를 기초로 데이터를 조작할 수 있게 함

#### 단점

1

뷰의 정의를 변경할 수 없음

변경이 필요한 경우 뷰를 삭제하고 다시 생성해야 함

2

삽입, 삭제, 변경 연산에 제한이 많음

## 2. SQL 뷰

### 4) 뷰의 종류

#### 단순 뷰(Single View)

오직 하나의 테이블만을 기초로 생성된 뷰

산술식 등에 의해 데이터가 조작된 경우를 제외하면, 뷰를 통한 모든 DML 연산의 수행이 가능

#### 복합 뷰(Complex or Join View)

여러 테이블을 기초로 생성된 뷰

데이터 그룹핑 또는 그룹 함수를 사용해서 뷰를 생성

뷰를 통한 모든 DML이 항상 가능한 것은 아님

## 2. SQL 뷰

### 5) 뷰 생성

CREATE VIEW 명령

SELECT 문이 수행되어 가져온 열들로 구성된 뷰를 생성함

CREATE VIEW 명령의 일반 형식

```
CREATE VIEW 뷰_이름[(열_이름 리스트)]  
AS SELECT문  
[WITH CHECK OPTION];
```

삽입이나 변경할 때 뷰의 정의 조건을  
검사해서 뷰에 의해 접근 가능한 행만  
삽입 또는 변경될 수 있도록 지정함

## 2. SQL 뷰

## 5) 뷰 생성

## 뷰 생성 예

## 단순 뷰 생성하기

사원\_2 테이블에서 부서번호가 200인 사원 데이터의 사번과 이름, 연봉, 휴대폰만 포함하는 '비서실\_사원'이라는 이름의 뷰를 생성함

```
CREATE VIEW 비서실_사원(사번, 이름, 연봉, 휴대폰)
AS SELECT 사번, 이름, 연봉, 휴대폰
FROM 사원_2
WHERE 부서번호 = 200;
```

## 뷰 확인

## 비서실\_사원 뷰

사번	이름	연봉	휴대폰
1102	이기원	6000	011-1111-2222
1103	박영종	4500	010-2222-3333



## 2. SQL 뷰

### 5) 뷰 생성

#### 뷰 생성 예

#### 열 별칭을 사용해서 뷰 생성하기

사원\_2 테이블에서 부서번호가 300인 사원 데이터의 사번과 이름, 연봉, 휴대폰만 포함하는 '총무부\_사원'이라는 이름의 뷰를 생성하되, 열 이름은 id, name, salary, cellphone으로 지정함

```
CREATE VIEW 총무부_사원(id, name, salary, cellphone)
AS SELECT 사번, 이름, 연봉, 휴대폰
FROM 사원_2
WHERE 부서번호 = 300;
```

#### 뷰 확인

#### 총무부\_사원 뷰

id	name	salary	cellphone
1201	최강희	7500	010-1111-2222
1202	조현수	4000	010-2222-3333

## 2. SQL 뷰

### 6) 뷰 삭제

DROP VIEW 명령

기존의 뷰를 삭제함

DROP VIEW 명령의 일반 형식

```
DROP VIEW 뷰_이름  
[RESTRICT | CASCADE];
```

뷰가 다른 곳에서 참조되고 있지 않을 때만  
제거함

뷰를 참조하는 다른 모든 뷰나 제약조건을 함께  
제거함

뷰 삭제 예제

앞에서 생성한 총무부\_사원 뷰를 삭제하되, 이 뷰가 다른 곳에서 참조되고 있으면 삭제하지 않음

```
DROP VIEW 총무부_사원 RESTRICT;
```

## 2. SQL 뷰

### 7) 뷰 조작

#### 뷰 조작 연산의 한계점

기본 테이블에서 사용 가능한 검색(SELECT 명령)은 모두 사용할 수 있지만,  
삽입, 삭제, 변경은 제한적으로 사용 가능

삽입, 삭제, 변경이 허용되는 경우	삽입, 삭제, 변경이 허용되지 않는 경우
뷰가 하나의 기본 테이블의 행과 열의 부분 집합으로 정의된 경우	① 뷰의 열이 상수나 산술 연산자 또는 함수가 사용된 산술식으로 만들어진 경우 ② 그룹 함수(COUNT, SUM, AVG, MAX, MIN)를 사용해서 정의된 경우 ③ DISTINCT, GROUP BY, HAVING 절이 사용되어 정의된 경우 ④ 두 개 이상의 테이블이 관련되어 정의된 경우 ⑤ 변경할 수 없는 뷰를 기초로 정의된 경우

쉼터

DREAMISNOWHERE

위의 문장을 읽어보세요.

여러분은 이 문장을 어떻게 읽으셨나요?

Dream is no where (꿈은 어디에도 없다)이라고 읽으셨나요?

아니면 Dream is now here (꿈은 바로 여기에) 이라고  
읽으셨나요?

모든 것은 마음 먹기에 달려있습니다.

긍정의 힘! 밝은 곳을 보려고 노력합니다.