

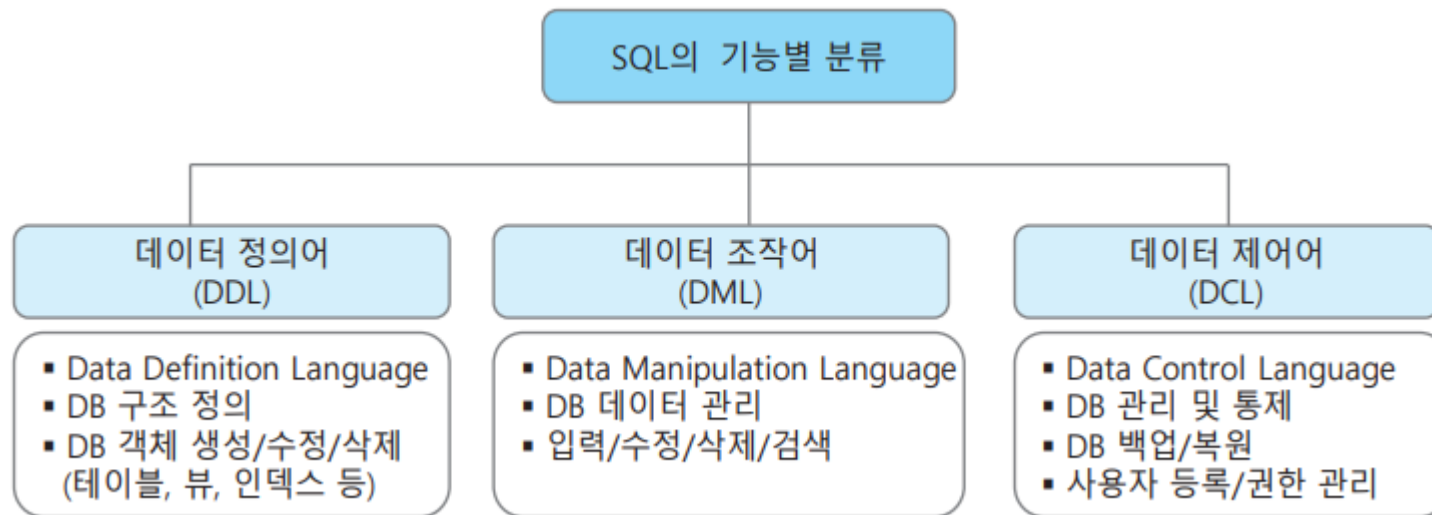
02.SQL 기초

1. SQL 개요

- SQL(Structured Query Language)

- 관계형 데이터베이스 표준 언어로서 가장 많이 사용되는 데이터 언어
 - SYSTEM R'이라는 실험용 DBMS를 위한 데이터 언어로 IBM 연구소에서 처음 개발
 - 현재는 미국 표준(ANSI)과 국제 표준(ISO) 관계형 데이터베이스 표준 언어로 승인
- ORACLE, MS SQL-Server, MySQL 등 거의 모든 관계형 DBMS가 지원

- SQL을 구성하는 3가지 부속 언어의 분류와 관련 주요 기능



SQL의 발전 과정

● SQL 표준의 역사

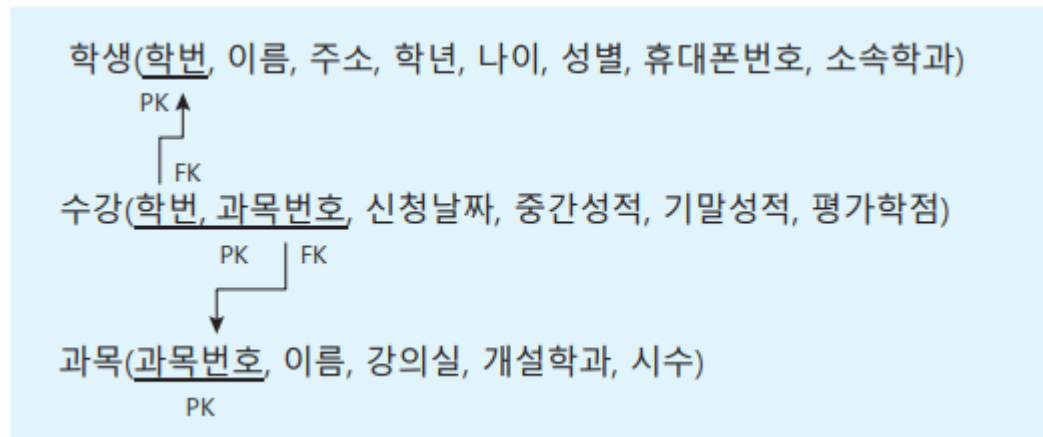
표준 버전	승인연도	특징
최초 안(SQL-86)	1986년	기본 기능인 데이터 정의어, 조작어 그리고 무결성을 포함한 표준 미국 ANSI 표준 채택(1986), 국제 ISO 표준 채택(1987)
SQL1(SQL-89)	1989년	ANSI/ISO SQL 표준을 보완 및 확장
SQL2(SQL-92)	1992년	SQL-89를 대폭 보완 및 확장 3단계의 표준 준수 수준(도입,중간,완전)을 제시
SQL3(SQL-1999)	1999년	SQL-92를 확장 및 보완 객체지향 개념, 재귀 쿼리, 데이터베이스 트리거 등을 도입
SQL-2003 4차	2003년	XML 관련 개념 및 특징 지원
SQL-2006 5차	2006년	XML 데이터 처리 지원
SQL-2008 6차	2008년	커서, 트리거 개념 확장
SQL-2011 7차	2011년	임시 데이터베이스 지원

2. SQL 실습 준비

2.1 예제 데이터베이스 univDB

- SQL 실습을 위해서 먼저, 'univDB' 예제 데이터베이스를 구성
- 수강 관련 정보를 저장할 '학생', '수강', '과목' 3개의 테이블

● univDB 데이터베이스 스키마



univDB 데이터베이스의 내용

학생

학번	이름	주소	학년	나이	성별	휴대폰번호	소속학과
s001	김연아	서울 서초	4	23	여	010-1111-2222	컴퓨터
s002	홍길동	미정	1	26	남	NULL	통계
s003	이승엽	NULL	3	30	남	NULL	정보통신
s004	이영애	경기 분당	2	NULL	여	010-4444-5555	정보통신
s005	송윤아	경기 분당	4	23	여	010-6666-7777	컴퓨터
s006	홍길동	서울 종로	2	26	남	010-8888-9999	컴퓨터
s007	이은진	경기 과천	1	23	여	010-2222-3333	경영

과목

과목번호	이름	강의실	개설학과	시수
c001	데이터베이스	126	컴퓨터	3
c002	정보보호	137	정보통신	3
c003	모바일웹	128	컴퓨터	3
c004	철학개론	117	철학	2
c005	전공글쓰기	120	교양학부	1

수강

학번	과목번호	신청날짜	중간성적	기말성적	평가학점
s001	c002	2019-09-03	93	98	A
s004	c005	2019-03-03	72	78	C
s003	c002	2017-09-06	85	82	B
s002	c001	2018-03-10	31	50	F
s001	c004	2019-03-05	82	89	B
s004	c003	2020-09-03	91	94	A
s001	c005	2020-09-03	74	79	C
s003	c001	2019-03-03	81	82	B
s004	c002	2018-03-05	92	95	A

MySQL 실습 준비

3) MySQL 버전과 시스템 날짜시간 정보 확인

```
SELECT version( ) ; -- MySQL 버전 표시  
SELECT current_date( ), current_time( ), now( ) ; -- 현재 날짜와 시간 표시
```

4) 현재 접속한 사용자 아이디 확인

```
SELECT user( ) ; -- 현재 MySQL 사용자 표시
```

5) 데이터베이스 목록 확인

```
SHOW DATABASES ; -- 현재 데이터베이스 목록 표시
```

MySQL 실습 준비

6) 'univDB' 실습 예제 데이터베이스를 생성

-- 데이터베이스 생성

```
DROP DATABASE IF EXISTS univDB ;
```

```
CREATE DATABASE IF NOT EXISTS univDB ;
```

-- SQL 명령어를 실행할 대상인 기본 데이터베이스를 univDB로 지정

```
USE univDB ;
```

-- 학생 테이블 생성

```
CREATE TABLE 학생
```

```
( 학번 CHAR(4) NOT NULL,  
  이름 VARCHAR(20) NOT NULL,  
  주소 VARCHAR(50) NULL DEFAULT '미정',  
  학년 INT NOT NULL,  
  나이 INT NULL,  
  성별 CHAR(1) NOT NULL,  
  휴대폰번호 CHAR(14) NULL,  
  소속학과 VARCHAR(20) NULL,  
  PRIMARY KEY (학번) ) ;
```

-- 학생 테이블 입력

```
INSERT INTO 학생
```

```
VALUES ('s001', '김연아', '서울 서초', 4, 23, '여', '010-1111-2222', '컴퓨터') ;
```

```
INSERT INTO 학생
```

```
VALUES ('s002', '홍길동', DEFAULT, 1, 26, '남', NULL, '통계') ;
```

```
INSERT INTO 학생
```

```
VALUES ('s003', '이승엽', NULL, 3, 30, '남', NULL, '정보통신') ;
```

```
INSERT INTO 학생
```

```
VALUES ('s004', '이영애', '경기 분당', 2, NULL, '여', '010-4444-5555', '정보통신') ;
```

```
INSERT INTO 학생
```

```
VALUES ('s005', '송윤아', '경기 분당', 4, 23, '여', '010-6666-7777', '컴퓨터') ;
```

```
INSERT INTO 학생
```

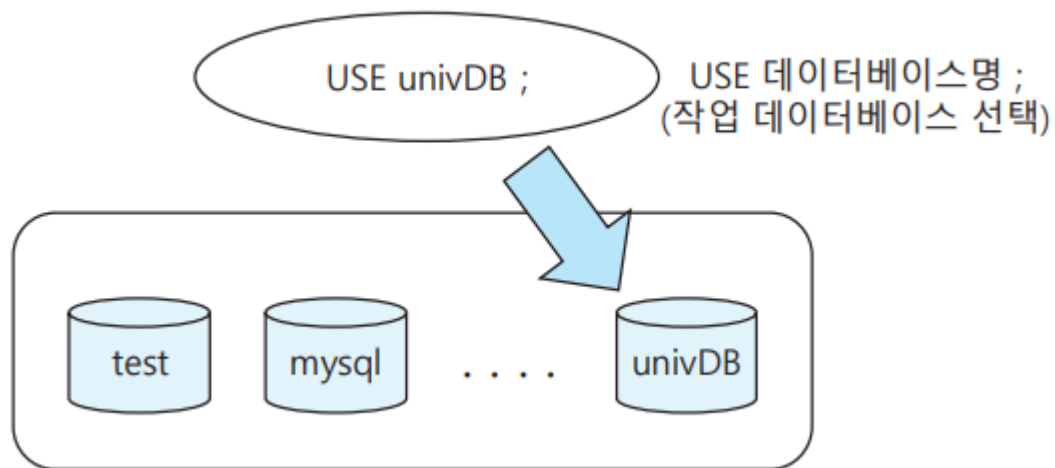
```
VALUES ('s006', '홍길동', '서울 종로', 2, 26, '남', '010-8888-9999', '컴퓨터') ;
```

```
INSERT INTO 학생
```

```
VALUES ('s007', '이은진', '경기 과천', 1, 23, '여', '010-2222-3333', '경영') ;
```

MySQL 실습 준비

● 작업 데이터베이스 선택



USE univDB ;	-- 작업 대상 데이터베이스 변경
SELECT database() ;	-- 현재 사용 데이터베이스 확인
SHOW TABLES ;	-- univDB 안의 생성 테이블 목록 확인
DESC 학생 ;	-- 학생 테이블 생성 정보 확인

3. SQL 데이터 조작성

3.1 데이터 검색 SELECT문

- 행 검색

- 테이블로부터 데이터를 검색하기 위해서는 SELECT문 사용

- SELECT문의 형식: SELECT절, FROM절

- SELECT 명령문의 형식
- []로 표시한 부분은 생략 가능, { }로 표시한 부분은 반복 가능
- ‘|’은 선택을, ‘+’는 1번 이상 반복을, ‘*’는 0번 이상 반복을 의미
- ①~⑥은 SELECT 명령문을 구성하는 6개의 절의 의미상의 실행 순서를 나타냄

```
①SELECT [DISTINCT|ALL] 열_리스트           ⑤
②FROM   테이블_리스트                     ①
        [WHERE 검색_조건식]                 ②
        [GROUP BY 그룹_기준열_리스트]       ③
        [HAVING 그룹_조건식]                 ④
        [ORDER BY { 정렬_기준열 [ASC|DESC][,] }+ ]; ⑥
```

데이터 검색 SELECT문

- 테이블의 특정 열 검색

예제 5-1 전체 학생의 이름과 주소를 검색하시오.

```
SELECT 이름, 주소  
FROM 학생 ;
```

- 테이블의 모든 열 검색

예제 5-2 전체 학생의 모든 정보를 검색하시오.

```
SELECT 학번, 이름, 주소, 학년, 나이, 성별, 휴대폰번호, 소속학과  
FROM 학생 ;
```

```
SELECT *  
FROM 학생 ;
```

- 중복 행을 제거하고 검색: DISTINCT

예제 5-3 전체 학생의 소속학과 정보를 중복없이 검색하시오.

```
SELECT DISTINCT 소속학과  
FROM 학생 ;
```

데이터 검색 SELECT문

■ 조건 검색: WHERE절

```
SELECT ①[DISTINCT|ALL] 열_리스트  
FROM 테이블_리스트  
[WHERE ③검색_조건식] ;②
```

예제 5-4 학생 중에서 2학년 이상인 '컴퓨터'학과 학생의 이름, 학년, 소속학과, 휴대폰번호 정보를 검색하시오.

```
SELECT 이름, 학년, 소속학과, 휴대폰번호  
FROM 학생  
WHERE 학년>=2 AND 소속학과='컴퓨터' ;
```

예제 5-5 1, 2, 3학년 학생이거나 '컴퓨터'학과에 소속되지 않은 학생의 이름, 학년, 소속학과, 휴대폰번호 정보를 검색하시오.

```
SELECT 이름, 학년, 소속학과, 휴대폰번호  
FROM 학생  
WHERE (학년>=1 AND 학년<=3) OR NOT(소속학과='컴퓨터') ;
```

```
SELECT 이름, 학년, 소속학과, 휴대폰번호  
FROM 학생  
WHERE (학년 BETWEEN 1 AND 3) OR NOT(소속학과='컴퓨터') ;
```

데이터 검색 SELECT문

■ 순서화 검색: ORDER BY절

```
SELECT [DISTINCT|ALL] 열_리스트  
FROM 테이블_리스트  
[ORDER BY {①정렬_기준열 ②[ASC|DESC][,]}*] ;
```

예제 5-6 '컴퓨터'학과나 '정보통신'학과의 학생의 이름과 학년, 소속학과 정보를 학년의 오름차순(학년이 낮은 학생부터 높은 학생 순서)으로 검색하시오.

```
SELECT 이름, 학년, 소속학과  
FROM 학생  
WHERE 소속학과='컴퓨터' OR 소속학과='정보통신'  
ORDER BY 학년 ASC ;
```

예제 5-7 전체 학생의 모든 정보를 검색하되 학년을 기준으로 먼저 1차 오름차순 정렬하고, 학년이 같은 경우에는 이름을 기준으로 2차 내림차순 정렬하여 검색하시오.

```
SELECT *  
FROM 학생  
ORDER BY 학년 ASC, 이름 DESC ;
```

집계 함수(aggregate function)

● 집계 함수 검색

- 실제 테이블 저장 값이 아닌 행의 개수(count) 또는 특정 열의 값 평균(average)을 구하는 질의가 필요할 경우를 위해 SQL은 집계 함수를 제공
- 집계 함수
 - 각 열에 대한 기본 통계 결과를 반환하는 함수
 - COUNT(), MAX(), MIN(), SUM(), AVG()

● SQL 기본 집계 함수

집계 함수	기능
COUNT(열_이름)	특정 열 값의 개수 또는 행의 개수
MAX(열_이름)	특정 열 값 중에서 최댓값
MIN(열_이름)	특정 열 값 중에서 최솟값
SUM(열_이름)	특정 열 값의 합계
AVG(열_이름)	특정 열 값의 평균

데이터 검색 SELECT문

예제 5-8 전체 학생수를 검색하시오.

```
SELECT COUNT(*)  
FROM 학생 ;
```

```
SELECT COUNT(학번)  
FROM 학생 ;
```

```
SELECT COUNT(*) AS 학생수1, COUNT(주소) AS 학생수2, COUNT(DISTINCT 주소) AS 학생수3  
FROM 학생 ;
```

예제 5-9 '여'학생의 평균 나이를 검색하시오.

```
SELECT AVG(나이) '여학생 평균나이'  
FROM 학생  
WHERE 성별='여' ;
```

그룹화 검색: GROUP BY절

```
SELECT [DISTINCT|ALL] 열_리스트②  
FROM 테이블_리스트  
[GROUP BY 그룹_기준열_리스트]①  
[HAVING 그룹_조건식] ;③
```

예제 5-10 전체 학생의 성별 최고 나이와 최저 나이를 검색하시오.

```
SELECT 성별, MAX(나이) AS '최고 나이', MIN(나이) AS '최저 나이'  
FROM 학생  
GROUP BY 성별 ;
```

예제 5-11 20대 학생만을 대상으로 나이별 학생수를 검색하시오.

```
SELECT 나이, COUNT(*) AS '나이별 학생수'  
FROM 학생  
WHERE 나이 >= 20 and 나이 < 30  
GROUP BY 나이 ;
```

그룹 조건 검색: HAVING절

- HAVING절은 그룹이 만족해야 하는 제한 조건을 명세
 - GROUP BY절을 통해 생성된 그룹 중에서도 특정 조건을 만족하는 그룹만으로 검색을 제한
 - WHERE절이 행에 대한 '검색_조건식'을 명세한다면 HAVING절은 행 그룹에 대한 '그룹_조건식'을 명세

예제 5-12 각 학년별로 2명 이상의 학생을 갖는 학년에 대해서만 학년별 학생수를 검색하시오.

```
SELECT 학년, COUNT(*) AS '학년별 학생수'
FROM 학생
GROUP BY 학년
HAVING COUNT(*) >= 2;
```


LIKE 문자 연산자 검색: ‘_’ , ‘%’ 와일드카드

- 부분 문자열(substring) 포함 여부를 검색하기 위해 WHERE절 ‘검색_조건식’에 사용되는 비교 연산자
- 검색 문자열 값을 정확히 모르거나 일부만 알고 있는 경우

기호	기능
_	문자열 중 특정 위치에 1개의 모든 문자를 허용
%	문자열 중 임의 위치에 0개 이상의 모든 문자(열)를 허용

- LIKE 와일드카드의 적용 예

적용 예	검색 일치	검색 불일치
LIKE ‘_ook’	book, look	good, books, ook
LIKE ‘9_7_5’	98765, 91715, 9a7b5	975, 98754
LIKE ‘%k’	book, milk, ok, k,	books
LIKE ‘b%k’	book, bblack, bk	kbkbk, bbooks

데이터 검색 SELECT문

예제 5-13 '이'씨 성을 가진 학생들의 학번과 학생 이름을 검색하시오.

```
SELECT 학번, 이름  
FROM 학생  
WHERE 이름 LIKE '이__' ;
```

예제 5-14 주소지가 '서울'인 학생의 이름, 주소, 학년을 학년 순(내림차순)으로 검색하시오.

```
SELECT 이름, 주소, 학년  
FROM 학생  
WHERE 주소 LIKE '%서울%'  
ORDER BY 학년 DESC ;
```

널 값 검색: IS NULL, IS NOT NULL

- 열 값으로 NULL을 갖고 있는 행을 검색할 경우
 - '=' 비교 연산자 사용 불가

예제 5-15 휴대폰번호가 등록되지 않은(널 값을 갖는) 학생의 이름과 휴대폰번호를 검색하시오.

```
SELECT 이름, 휴대폰번호  
FROM 학생  
WHERE 휴대폰번호 IS NULL ;
```



Note

널(NULL) 값 검색

■ NULL 비교 검색

NULL은 아직 지정되지 않은 값을 의미한다. 특정 열 값의 입력이 생략될 경우 대신 입력된다.

NULL은 '0', ' '(공백 문자), ''(빈 문자) 등과 다른 특별한 값으로 =, >, <= 등과 같은 비교 연산자로는 비교가 불가능하다. 널 값을 찾을 때는 '=' 대신 'IS NULL'을 사용하고 널이 아닌 값을 찾을 때는 <> 대신 'IS NOT NULL'을 사용한다.

■ NULL 연산

NULL을 집계 함수에 사용할 때 다음에 유의한다

- NULL이 포함된 산술식 연산의 결과는 항상 NULL이다.
- NULL이 포함된 행은 대부분의 집계 함수 결과에서 제외된다.
- '100 <> NULL'처럼 널 값을 일반 비교연산자로 비교하면 결과는 항상 거짓(false)이다.