MySql workbook

MYSQL

SQL

SQL

- MySQL 데이터 형식
- 변수의 활용
- MySQL 내장 함수
- 테이블 조인
- SQL 프로그래밍

1 MySQL의 데이터 형식

- MySQL에서 지원하는 데이터 형식의 종류
  - Data Type으로 표현
    - 데이터 형식, 데이터형, 자료형, 데이터 타입
  - 데이터 형식에 대한 이해
    - SELECT문 더욱 잘 활용
    - 테이블의 생성 효율적으로 하기 위해 필요
  - MySQL에서 데이터 형식의 종류는 30개 정도
    - 중요하고 자주 쓰는 형식에 대해 중점 학습

1 MySQL의 데이터 형식

MySQL에서 지원하는 데이터 형식의 종류

- 숫자 데이터 형식

데이터 형식	바이트 수	숫자 범위	설명
BIT(N)	N/8		1~64bit를 표현. b'0000' 형식으로 표현
TINYINT	1	-128~127	정수
★SMALLINT	2	-32,768~32,767	정수
MEDIUMINT	3	-8,388,608~8,388,607	정수
★INT INTEGER	4	약 -21억~+21억	정수
★BIGINT	8	약 -900경~+900경	정수
★FLOAT	4	-3.40E+38~-1.17E-38	소수점 아래 7자리까지 표현
★DOUBLE REAL	8	-1.22E-308~1.79E+308	소수점 아래 15자리까지 표현
★DECIMAL(m, [d]) NUMERIC(m, [d])	5~17	$-10^{38}+1$~$+10^{38}-1$	전체 자릿수(m)와 소수점 이하 자릿수(d)를 가진 숫자형 예) decimal(5, 2)은 전체 자릿수를 5자리로 하되, 그 중 소수점 이하를 2자리로 하겠다는 의미

1 MySQL의 데이터 형식

◆ MySQL에서 지원하는 데이터 형식의 종류

- 문자 데이터 형식

데이터 형식		바이트 수	설명
★CHAR(n)		1~255	고정길이 문자형. n을 1부터 255까지 지정. character의 약자 그냥 CHAR만 쓰면 CHAR(1)과 동일
★VARCHAR(n)		1~65535	가변길이 문자형. n을 사용하면 1부터 65535까지 지정. Variable character의 약자
BINARY(n)		1~255	고정길이의 이진 데이터 값
VARBINARY(n)		1~255	가변길이의 이진 데이터 값
TEXT 형식	TINYTEXT	1~255	255 크기의 TEXT 데이터 값
	TEXT	1~65535	N 크기의 TEXT 데이터 값
	MEDIUMTEXT	1~16777215	16777215 크기의 TEXT 데이터 값
	★LONGTEXT	1~4294967295	최대 4GB 크기의 TEXT 데이터 값
BLOB 형식	TINYBLOB	1~255	255 크기의 BLOB 데이터 값
	BLOB	1~65535	N 크기의 BLOB 데이터 값
	MEDIUMBLOB	1~16777215	16777215 크기의 BLOB 데이터 값
	★LONGBLOB	1~4294967295	최대 4GB 크기의 BLOB 데이터 값
ENUM(값들…)		1 또는 2	최대 65535개의 열거형 데이터 값
SET(값들…)		1, 2, 3, 4, 8	최대 64개의 서로 다른 데이터 값

1 MySQL의 데이터 형식

MySQL에서 지원하는 데이터 형식의 종류

- 날짜와 시간 데이터 형식

데이터 형식	바이트 수	설명
★DATE	3	날짜는 1001-01-01~9999-12-31까지 저장되며 날짜 형식만 사용 'YYYY-MM-DD' 형식으로 사용됨
TIME	3	-838:59:59.000000~838:59:59.000000까지 저장되며 'HH:MM:SS' 형식으로 사용
★DATETIME	8	날짜는 1001-01-01 00:00:00~9999-12-31 23:59:59까지 저장되며 형식은 'YYYY-MM-DD HH:MM:SS' 형식으로 사용
TIMESTAMP	4	날짜는 1001-01-01 00:00:00~9999-12-31 23:59:59까지 저장되며 형식은 'YYYY-MM-DD HH:MM:SS' 형식으로 사용. time_zone 시스템 변수와 관련이 있으며 UTC 시간대로 변환하여 저장
YEAR	1	1901~2155까지 저장. 'YYYY' 형식으로 사용

DATE	TIME	DATETIME
2020-10-19	12:35:29	2020-10-19 12:35:29

1 MySQL의 데이터 형식

MySQL에서 지원하는 데이터 형식의 종류

- 기타 데이터 형식
  - JSON 데이터 형식은 MySQL 5.7.8 이후부터 지원

데이터 형식	바이트 수	설명
GEOMETRY	N/A	공간 데이터 형식으로 선, 점 및 다각형 같은 공간 데이터 개체를 저장하고 조작
JSON	8	JSON(JavaScript Object Notation) 문서를 저장

- LONGTEXT, LONGBLOB
  - LOB (Large Object, 대량의 데이터) 을 저장
  - LONGTEXT, LONGBLOB 데이터 형식 지원
  - 지원되는 데이터 크기는 약 4GB의 파일을 하나의 데이터로 저장 가능
  - P.234 표의 예시를 보며 이해

1 MySQL의 데이터 형식

◆ 데이터 형식과 형 변환

▪ 데이터 형식 변환 함수

- 데이터 형식 중에서 가능한 것은 BINARY, CHAR, DATA, DATATIME, DECIMAL, JSON, SIGNED INTEGER, TIME, UNSIGNED INTEGER
- CAST( ), CONVERT( ) 함수 주로 사용
- 함수 사용법

```
형식:
CAST ( expression AS 데이터형식 [ (길이) ] )
CONVERT ( expression ,  데이터형식 [ (길이) ] )
```

1 MySQL의 데이터 형식

◆ 데이터 형식과 형 변환

- 암시적인 형 변환
  - CAST( )나 CONVERT( ) 함수를 사용하지 않고 형이 변환되는 것
  - 예시를 보며 이해하는 것이 빠름

```
SELECT '100' + '200' ; -- 문자와 문자를 더함 (정수로 변환되서 연산됨)
SELECT CONCAT('100', '200'); -- 문자와 문자를 연결 (문자로 처리)
SELECT CONCAT(100, '200'); -- 정수와 문자를 연결 (정수가 문자로 변환되서 처리)
SELECT 1 > '2mega'; -- 정수인 2로 변환되어서 비교
SELECT 3 > '2MEGA'; -- 정수인 2로 변환되어서 비교
SELECT 0 = 'mega2'; -- 문자는 0으로 변환됨
```

'100' + '200'	CONCAT('100', '200')	CONCAT(100, '200')	1 > '2mega'	3 > '2MEGA'	0 = 'mega2'
300	100200	100200	0	1	1

1 MySQL의 데이터 형식

MySQL 내장 함수

제어 흐름 함수

- 제어 흐름 함수는 프로그램의 흐름 제어하는 역할

- IF (수식, 참, 거짓)
  - 수식이 참 또는 거짓인지 결과에 따라서 2중 분기

- IFNULL(수식1, 수식2)
  - 수식1이 NULL이 아니면 수식1이 반환
  - 수식1이 NULL이면 수식2가 반환

1 MySQL의 데이터 형식

MySQL 내장 함수

제어 흐름 함수

- NULLIF(수식1, 수식2)
  - 수식1과 수식2가 같으면 NULL을 반환
  - 다르면 수식1을 반환

- CASE ~ WHEN ~ ELSE ~ END
  - CASE는 내장 함수는 아니며 연산자Operator로 분류
  - 다중 분기에 사용

```
SELECT  CASE 10
            WHEN 1  THEN  '일'
            WHEN 5  THEN  '오'
            WHEN 10 THEN  '십'
            ELSE '모름'
        END;
```

1 MySQL의 데이터 형식

MySQL 내장 함수

문자열 함수

- 문자열 함수는 문자열을 조작. 활용도 높음

- ASCII (아스키 코드), CHAR(숫자)
  - 문자의 아스키 코드값을 돌려주거나 숫자의 아스키 코드값에 해당하는 문자를 돌려줌

- BIT_LENGTH(문자열), CHAR_LENGTH(문자열), LENGTH(문자열)
  - 할당된 Bit 크기 또는 문자 크기를 반환
  - CHAR_LENGTH( )는 문자의 개수 반환
  - LENGTH( )는 할당된 Byte 수 반환

1 MySQL의 데이터 형식

MySQL 내장 함수

문자열 함수

- CONCAT(문자열1, 문자열2,...), CONCAT_WS(문자열1, 문자열2,...)
  - 문자열을 이어줌
  - CONCAT_WS( )는 구분자와 함께 문자열을 이어주는 역할

- ELT(위치, 문자열1, 문자열2, ...), FIELD(찾을 문자열, 문자열1, 문자열2, ...), FIND_IN_SET (찾을 문자열, 문자열 리스트), INSTR(기준 문자열, 부분 문자열), LOCATE(부분 문자열, 기준 문자열)
  - ELT( )는 위치 번째에 해당하는 문자열 반환
  - FIELD( )는 찾을 문자열의 위치를 찾아 반환 → 없으면 0
  - FIND_IN_SET( )은 찾을 문자열을 문자열 리스트에서 찾아 위치 반환
    - » 문자열 리스트는 콤마(,)로 구분되어 있고 공백이 없어야 함
  - INSTR( )는 기준 문자열에서 부분 문자열 찾아 그 시작 위치 반환
  - LOCATE( )는 INSTR( )와 동일하지만 파라미터의 순서가 반대

1 MySQL의 데이터 형식

MySQL 내장 함수

문자열 함수

- FORMAT(숫자, 소수점 자릿수)
  - 숫자를 소수점 아래 자릿수까지 표현
  - 1000단위마다 콤마 표시해 줌

- BIN(숫자), HEX(숫자), OCT(숫자)
  - 2진수, 16진수, 8진수의 값을 반환

- INSERT(기준 문자열, 위치, 길이, 삽입할 문자열)
  - 기준 문자열의 위치부터 길이만큼 지우고 삽입할 문자열 끼워 넣음

- LEFT(문자열, 길이), RIGHT(문자열, 길이)
  - 왼쪽 또는 오른쪽에서 문자열의 길이만큼 반환

1 MySQL의 데이터 형식

MySQL 내장 함수

문자열 함수

- UCASE(문자열), LCASE(문자열)
  - 소문자를 대문자로, 대문자를 소문자로 변경
- UPPER(문자열), LOWER(문자열)
  - 소문자를 대문자로, 대문자를 소문자로 변경
- LPAD(문자열, 길이, 채울 문자열), RPAD(문자열, 길이, 채울 문자열)
  - 문자열을 길이만큼 늘린 후에 빈 곳을 채울 문자열로 채움
- LTRIM(문자열), RTRIM(문자열)
  - 문자열의 왼쪽/오른쪽 공백을 제거
  - 중간의 공백은 제거되지 않음

1 MySQL의 데이터 형식

- MySQL 내장 함수
  - 문자열 함수
    - TRIM(문자열), TRIM(방향 자를_문자열 FROM 문자열)
      - TRIM(문자열)은 문자열의 앞뒤 공백을 모두 없앰
      - TRIM(방향 자를_문자열 FROM 문자열) 에서 방향은 LEADING(앞), BOTH(양쪽), TRAILING(뒤) 으로 표시
    - REPEAT(문자열, 횟수)
      - 문자열을 횟수만큼 반복
    - REPLACE(문자열, 원래 문자열, 바꿀 문자열)
      - 문자열에서 원래 문자열을 찾아서 바꿀 문자열로 바꿈
    - REVERSE(문자열)
      - 문자열의 순서를 거꾸로 바꿈

1 MySQL의 데이터 형식

- MySQL 내장 함수
  - 문자열 함수
    - SPACE(길이)
      - 길이만큼의 공백을 반환
    - SUBSTRING(문자열, 시작위치, 길이) 또는 SUBSTRING(문자열 FROM 시작위치 FOR 길이)
      - 시작위치부터 길이만큼 문자를 반환
      - 길이가 생략되면 문자열의 끝까지 반환
    - SUBSTRING_INDEX(문자열, 구분자, 횟수)
      - 문자열에서 구분자가 왼쪽부터 횟수 번째까지 나오면 그 이후의 오른쪽은 버림
      - 횟수가 음수면 오른쪽부터 세고 왼쪽을 버림

1 MySQL의 데이터 형식

MySQL 내장 함수

수학 함수

- ABS(숫자)
  - 숫자의 절댓값 계산
- ACOS(숫자), ASIN(숫자), ATAN(숫자), ATAN2(숫자1, 숫자2), SIN(숫자), COS(숫자), TAN(숫자)
  - 삼각 함수와 관련된 함수 제공
- CEILING(숫자), FLOOR(숫자), ROUND(숫자)
  - 올림, 내림, 반올림 계산
- CONV(숫자, 원래 진수, 변환할 진수)
  - 숫자를 원래 진수에서 변환할 진수로 계산

1 MySQL의 데이터 형식

MySQL 내장 함수

수학 함수

- DEGREES(숫자), RADIANS(숫자), PI ( )
  - 라디안 값을 각도값으로, 각도값을 라디안 값으로 변환

- EXP(X), LN(숫자), LOG(숫자), LOG(밑수, 숫자), LOG2(숫자), LOG10(숫자)
  - 지수, 로그와 관련된 함수 제공

- MOD(숫자1, 숫자2) 또는 숫자1 % 숫자2 또는 숫자1 MOD 숫자2
  - 숫자1을 숫자2로 나눈 나머지 값을 구함

- POW(숫자1, 숫자2), SQRT(숫자)
  - 거듭제곱값 및 제곱근을 구함

1 MySQL의 데이터 형식

MySQL 내장 함수

수학 함수

- RAND( )
  - RAND( )는 0 이상 1 미만의 실수 구함
  - 'm<= 임의의 정수 < n'를 구하고 싶다면 FLOOR(m + (RAND( ) * (n-m)) 사용

- SIGN(숫자)
  - 숫자가 양수, 0, 음수인지 판별
  - 결과는 1, 0, -1 셋 중에 하나 반환

- TRUNCATE(숫자, 정수)
  - 숫자를 소수점을 기준으로 정수 위치까지 구하고 나머지는 버림

1 MySQL의 데이터 형식

MySQL 내장 함수

날짜 및 시간 함수

- ADDDATE(날짜, 차이), SUBDATE(날짜, 차이)
  - 날짜를 기준으로 차이를 더하거나 뺀 날짜 구함

- ADDTIME(날짜/시간, 시간), SUBTIME(날짜/시간, 시간)
  - 날짜/시간을 기준으로 시간을 더하거나 뺀 결과를 구함

- CURDATE( ), CURTIME( ), NOW( ), SYSDATE( )
  - CURDATE( )는 현재 연-월-일
  - CURTIME( )은 현재 시:분:초
  - NOW( )와 SYSDATE( )는 현재 '연-월-일 시:분:초

1 MySQL의 데이터 형식

MySQL 내장 함수

날짜 및 시간 함수

- YEAR(날짜), MONTH(날짜), DAY(날짜), HOUR(시간), MINUTE(시간), SECOND(시간), MICROSECOND(시간)
  - 날짜 또는 시간에서 연, 월, 일, 시, 분, 초, 밀리 초 구함

- DATE( ), TIME( )
  - DATETIME 형식에서 연-월-일 및 시:분:초만 추출

- DATEDIFF(날짜1, 날짜2), TIMEDIFF(날짜1 또는 시간1, 날짜1 또는 시간2)
  - DATEDIFF( )는 날짜1-날짜2의 일수를 결과로 구함

- DAYOFWEEK(날짜), MONTHNAME( ), DAYOFYEAR(날짜)
  - 요일(1:일, 2:월~7:토) 및 1년 중 몇 번째 날짜인지 구함

1 MySQL의 데이터 형식

- MySQL 내장 함수
  - 날짜 및 시간 함수
    - LAST_DAY(날짜)
      - 주어진 날짜의 달의 마지막 날짜를 구함 (그 달이 몇 일까지?)
    - MAKEDATE(연도, 정수)
      - 연도에서 정수만큼 지난 날짜 구함
    - MAKETIME(시, 분, 초)
      - 시, 분, 초를 이용해서 '시:분:초'의 TIME 형식 만듦
    - PERIOD_ADD(연월, 개월수), PERIOD_DIFF(연월1, 연월2)
      - PERIOD_ADD( )는 연월에서 개월만큼의 개월이 지난 연월 구함
      - PERIOD_DIFF( )는 연월1-연월2의 개월수 구함

1 MySQL의 데이터 형식

- MySQL 내장 함수
  - 날짜 및 시간 함수
    - QUARTER(날짜)
      - 날짜가 4분기 중에서 몇 분기인지를 구함
    - TIME_TO_SEC(시간)
      - 시간을 초 단위로 구함

1 MySQL의 데이터 형식

MySQL 내장 함수

시스템 정보 함수

- 시스템의 정보를 출력하는 함수 제공

- USER( ), DATABASE( )
  - 현재 사용자 및 현재 선택된 데이터베이스 출력

- FOUND_ROWS( )
  - 바로 앞의 SELECT문에서 조회된 행의 개수 구함

- VERSION( )
  - 현재 MySQL의 버전

- SLEEP(초)
  - 쿼리의 실행을 잠깐 멈춤

1 MySQL의 데이터 형식

- MySQL 내장 함수 이용한 예제
  - 예제를 통한 학습
    - TEXT 데이터 형식을 이용해 대량의 데이터를 입력
    - 입력 과정에서 내장 함수 REPEAT() 사용
    - 데이터 크기 확인 과정에서 내장 함수 LEGNTH() 사용
    - P.253~258 예제 참조

1 MySQL의 데이터 형식

JSON 데이터

- JSON (JavaScript Object Notation) 이란?
  - 웹과 모바일 응용프로그램 등과 데이터 교환하기 위한 개방형 표준 포맷
  - 속성(Key) 과 값 (Value) 으로 쌍을 이루며 구성
  - JavaScript 언어에서 파생
  - 특정한 프로그래밍 언어에 종속되어 있지 않은 독립적인 데이터 포맷
  - 포맷이 단순하고 공개되어 있기에 거의 대부분의 프로그래밍 언어에서 쉽게 읽거나 쓸 수 있도록 코딩 가능
  - MySQL 5.7.8부터 지원

2 조인

조인 (Join)

- 조인의 개념?
  - 두 개 이상의 테이블을 서로 묶어서 하나의 결과 집합으로 만들어 내는 작업

- 데이터베이스의 테이블
  - 여러 개의 테이블로 분리하여 저장
    - 중복과 공간 낭비를 피하고 데이터의 무결성 위함
  - 분리된 테이블들은 서로 관계(Relation) 를 가짐
    - 1대 다 관계에서 일어나는 데이터 처리 필요성

2 조인

INNER JOIN(내부 조인)

- 조인 중에서 가장 많이 사용되는 조인
  - 대개의 업무에서 조인은 INNER JOIN 사용
  - 일반적으로 JOIN이라고 얘기하는 것이 이 INNER JOIN 지칭
  - 사용 형식

```
SELECT <열 목록>
FROM <첫 번째 테이블>
      INNER JOIN <두 번째 테이블>
      ON <조인될 조건>
[WHERE 검색조건]
```

2 조인

◆ OUTER JOIN(외부 조인)

- 조인의 조건에 만족되지 않는 행까지도 포함시키는 것
  - '왼쪽 테이블의 것은 모두 출력되어야 한다' 고 해석하면 이해 쉬움

```
SELECT <열 목록>
FROM <첫 번째 테이블(LEFT 테이블)>
 <LEFT ¦ RIGHT ¦ FULL> OUTER JOIN <두 번째 테이블(RIGHT 테이블)>
      ON <조인될 조건>
[WHERE 검색조건] ;
```

2 조인

◆ CROSS JOIN(상호 조인)

- 한쪽 테이블의 모든 행들과 다른 쪽 테이블의 모든 행 조인
  - CROSS JOIN의 결과 개수는 두 테이블 개수를 곱한 개수
  - 카티션곱(Cartesian Product) 이라고도 부름
  - 상호조인 도식화 예시

회원 테이블(userTbl)

아이디	이름	생년	지역	국번	전화번호	키	가입일
LSG	이승기	1987	서울	011	1111111	182	2008.8.8
KBS	김범수	1979	경남	011	2222222	173	2012.4.4
KKH	김경호	1971	전남	019	3333333	177	2007.7.7
JYP	조용필	1950	경기	011	4444444	166	2009.4.4
SSK	성시경	1979	서울			186	2013.12.12
LJB	임재범	1963	서울	016	6666666	182	2009.9.9
YJS	윤종신	1969	경남			170	2005.5.5
EJW	은지원	1978	경북	011	8888888	174	2014.3.3
JKW	조관우	1965	경기	018	9999999	172	2010.10.10
BBK	바비킴	1973	서울	010	0000000	176	2013.5.5

PK

구매 테이블(buyTbl)

순번	아이디	물품명	분류	단가	수량
1	KBS	운동화		30	2
2	KBS	노트북	전자	1000	1
3	JYP	모니터	전자	200	1
4	BBK	모니터	전자	200	5
5	KBS	청바지	의류	50	3
6	BBK	메모리	전자	80	10
7	SSK	책	서적	15	5
8	EJW	책	서적	15	2
9	EJW	청바지	의류	50	1
10	BBK	운동화		30	2
11	EJW	책	서적	15	1
12	BBK	운동화		30	2

PK FK

2 조인

CROSS JOIN(상호 조인)

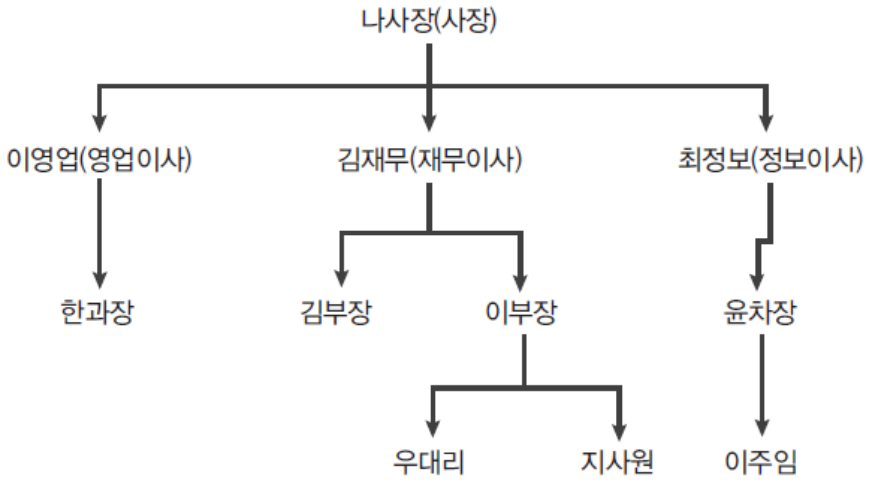
- 회원 테이블과 구매 테이블의 상호조인 구문

```
USE sqlDB;
SELECT *
   FROM buyTbl
     CROSS JOIN userTbl ;
```

2 조인

SELF JOIN(자체 조인)

- 자기 자신과 자기 자신이 조인한다는 의미
  - 조직도와 관련된 테이블



직원 이름(EMP) – 기본 키	상관 이름(MANAGER)	구내 번호
나사장	없음(NULL)	0000
김재무	나사장	2222
김부장	김재무	2222–1
이부장	김재무	2222–2
우대리	이부장	2222–2–1
지사원	이부장	2222–2–2
이영업	나사장	1111
한과장	이영업	1111–1
최정보	나사장	3333
윤차장	최정보	3333–1
이주임	윤차장	3333–1–1

Table & View

Table & View

- 테이블의 생성
- 제약 조건 : 기본 키, 외래 키 등
- 테이블 압축과 효율성 및 임시 테이블의 활용
- 뷰의 개념과 장단점

1 테이블

테이블 만들기

- MySQL Workbench 이용해 테이블 생성
  - 오류 메시지 이해할 수 있어야 함
    - Ex) 구매 테이블의 외래 키로 설정된 userid에 데이터가 입력되기 위해서는 입력될 값이 회원 테이블의 userid열에 존재해야 한다는 사항 이해

1 테이블

◆ 테이블 만들기

- **SQL**로 테이블 생성
  - SQL 에서 직접 실행시키는 명령어와 작동을 이해한 다음 Workbench에 익숙해지는 것이 개념 이해와 관리에 효율적
    - Ex) FOREIGN KEY REFERENCES userTbl(userID)
      - » 'userTbl 테이블의 userID열과 외래 키관계를 맺어라'

1 테이블

◆ 제약 조건

- 제약 조건 (Constraint) 이란?
  - 데이터의 무결성을 지키기 위한 제한된 조건 의미
  - 특정 데이터를 입력할 때 어떠한 조건을 만족했을 때에 입력되도록 제약
    - Ex) 동일한 주문등록 번호로 회원 중복 가입하지 못함
  - 데이터 무결성을 위한 제약조건
    - PRIMARY KEY 제약 조건
    - FOREIGN KEY 제약 조건
    - UNIQUE 제약 조건
    - DEFAULT 정의
    - NULL 값 허용

1 테이블

◆ 데이터 무결성 위한 제약 조건

- PRIMARY KEY 제약 조건
  - '기본 키Primary Key' 의 개념
    - 테이블에 존재하는 많은 행의 데이터를 구분할 수 있는 식별자
    - 중복되어서도 안되며 비어서도 안됨
    - Ex) 회원 테이블의 회원 아이디, 학생 테이블의 학번
  - 기본 키로 생성한 것은 자동으로 클러스터형 인덱스 생성
  - 테이블에서는 기본 키를 하나 이상의 열에 설정 가능
  - 기본 키 생성 방법

```
CREATE TABLE userTbl
( userID  char(8) NOT NULL PRIMARY KEY,
  name    nvarchar(10) NOT NULL,
   --- 중간 생략 ---
```

1 테이블

◆ 데이터 무결성 위한 제약 조건

- 외래 키 제약 조건
  - 두 테이블 사이의 관계 선언
    - 데이터의 무결성을 보장해 주는 역할
  - 외래 키 관계를 설정하면 하나의 테이블이 다른 테이블에 의존
  - 설정된 외래 키 제약 조건은 SHOW INDEX FROM buyTbl문으로 확인
  - ON DELETE CASCADE / ON UPDATE CASCADE
    - 기준 테이블의 데이터가 변경되었을 때 외래 키 테이블도 자동으로 적용되도록 설정

1 테이블

- 데이터 무결성 위한 제약 조건
  - UNIQUE 제약 조건
    - '중복되지 않는 유일한 값'을 입력해야 하는 조건
    - PRIMARY KEY와 거의 비슷하며 차이점은 UNIQUE는 NULL 값 허용
      - NULL은 여러 개가 입력되어도 상관 없음
      - Ex) 회원 테이블의 예를 든다면 주로 Email 주소 Unique로 설정

1 테이블

◆ 데이터 무결성 위한 제약 조건

- DEFAULT 정의
  - 값 입력하지 않았을 때 자동으로 입력되는 기본 값 정의하는 방법

- **Null** 값 허용
  - NULL 값을 허용하려면 NULL을, 허용하지 않으려면 NOT NULL 사용
    - PRIMARY KEY가 설정된 열에는 생략하면 자동으로 NOT NULL

1 테이블

테이블 삭제

- DROP TABLE 테이블이름;
- 외래 키 제약 조건의 기준 테이블은 삭제할 수가 없음
  - 먼저 외래 키가 생성된 외래 키 테이블을 삭제해야 함
- 동시에 여러 테이블 삭제도 가능

1 테이블

◆ 테이블 수정

- ALTER TABLE문 사용
  - 열의 추가
    - 기본적으로 가장 뒤에 추가
    - 순서를 지정하려면 제일 뒤에 FIRST 또는 AFTER 열이름 지정
  - 열의 삭제
    - ALTER TABLE userTbl

      DROP COLUMN 열 이름;

1 테이블

◆ 테이블 수정

- ALTER TABLE문 사용
  - 열의 이름 및 데이터 형식 변경
    - ALTER TABLE userTbl

      CHANGE COLUMN name uName VARCHAR(20) NULL ;

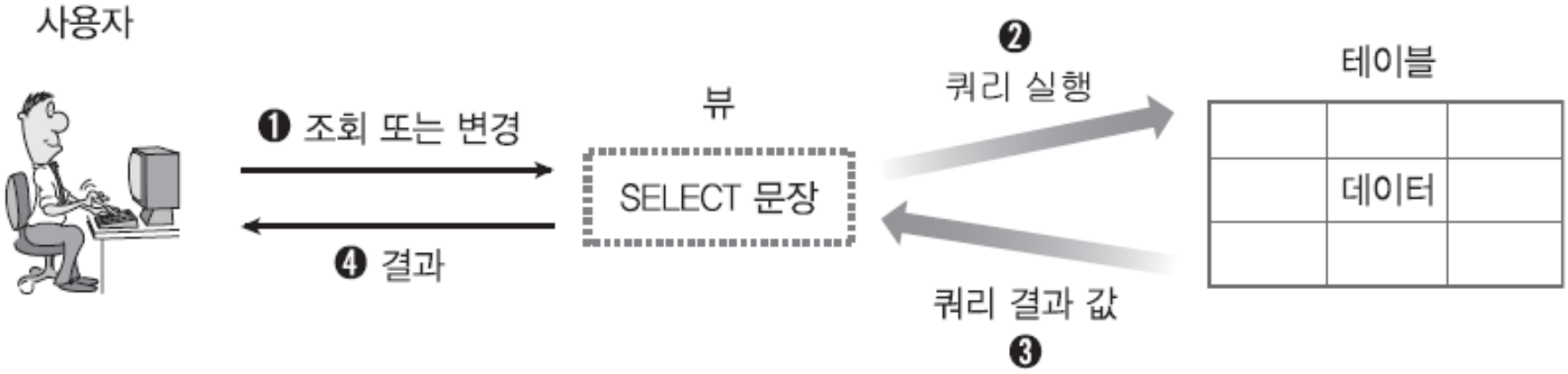
  - 열의 제약 조건 추가 및 삭제
    - 외래 키 연결 된 경우 외래 키부터 삭제 후 제약 조건 삭제 가능
    - ALTER TABLE userTbl

      DROP PRIMARY KEY;

2 뷰

◆ 뷰의 작성과 활용

- 일반 사용자 입장에서는 테이블과 동일하게 사용하는 개체
  - SELECT 문의 결과처럼 테이블의 형태를 가진 경우 새로운 테이블로 접근 가능
- 뷰의 작동 방식



2 뷰

뷰의 작성과 활용

- 뷰의 장점
  - 보안에 도움 – 사용자가 중요한 정보에 바로 접근하지 못함
  - 복잡한 쿼리 단순화
    - 긴 쿼리를 뷰로 작성
    - 뷰를 테이블처럼 사용 가능