■ Mysql 명령어

SHOW DATABASES; <현재 있는 데이터 베이스 이름을 본다.>

USE 데이터베이스이름; <데이터베이스이름를 사용한다.>

CREATE DATABASE 데이터베이스이름; <데이터베이스이름으로 데이터베이스를 만듬.>

DROP DATABASE 데이터베이스이름; <데이터베이스이름의 데이터베이스를 삭제>

-DATABASE- :USE 데이터베이스이름; 명령으로 사용할 데이터베이스 선택해야함

*SHOW TABLES; <데이터베이스안에 선언한 테이블 이름이 출력.>

*DESC 테이블이름; <데이터베이스안의 테이블이름의 구조를 본다.>

Ex) DESC aa; 라고 하면 aa라는 테이블 구조가 출력된다.

■ Sql 문

▲ select 문

형식]

select [distinct] {* | column[as Alias], ...}

from table명

[where 조건]

[group by 컬럼명]

[order by 컬럼명 [asc|desc]

Ex)

Select * from employee;

select ename, sal from emp;

▲ insert문

형식] insert into 테이블명 [(컬럼명, ….)] values(값1, …)

Ex)

insert into member(id, password, name, email, address)

values('jaen','1111','자앤','jaen@androidjava.com','서울시 서초구');

▲ update문

형식] update 테이블명 set 컬럼명=value, … [where 조건];

Ex) update member set address='서울시 강남구' where id='jaen';

▲ delete문

형식] delete from 테이블명 [where 조건];

Ex) delete from member where id='jaen';

■ 내장 함수

1. 숫자관련 함수

ABS(숫자)	절대값 출력.
CEILING(숫자)	값보다 큰 정수 중 가장 작은 수.
FLOOR(숫자)	값보다 작은 정수 중 가장 큰 수[실수를 무조건 버림 (음수일 경우는 제외)].
ROUND(숫자,자릿수)	숫자를 소수점 이하 자릿수에서 반올림. (자릿수는 양수,0,음수를 갖을 수 있다.)
TRUNCATE(숫자,자릿수)	숫자를 소수점 이하 자릿수에서 버림.
POW(X,Y) or POWER(X,Y)	X의 Y승
MOD (분자, 분모)	분자를 분모로 나눈 나머지를 구한다.(연산자 %와 같음)
GREATEST(숫자1,숫자2,숫자3...)	주어진 수 중 제일 큰 수 리턴.
LEAST(숫자1,숫자2,숫자3...)	주어진 수 중 제일 작은 수 리턴.
INTERVAL(a,b,c,d.....)	a(숫자)의 위치 반환

2. 문자관련 함수

ASCII(문자)

문자의 아스키 코드값 리턴.

CONCAT('문자열1','문자열2','문자열3'...)

문자열들을 이어준다.

INSERT('문자열','시작위치','길이','새로운문자열')

문자열의 시작위치부터 길이만큼 새로운 문자열로 대치

REPLACE('문자열','기존문자열','바뀔문자열')

문자열 중 기존문자열을 바뀔 문자열로 바꾼다.

INSTR('문자열','찾는문자열')

문자열 중 찾는 문자열의 위치값을 출력.

LEFT('문자열',개수)

문자열 중 왼쪽에서 개수만큼을 추출.

RIGHT('문자열',개수)

문자열 중 오른쪽에서 개수만큼을 추출.

MID('문자열',시작위치,개수)

문자열 중 시작위치부터 개수만큼 출력.

SUBSTRING('문자열',시작위치,개수)

문자열 중 시작위치부터 개수만큼 출력.

LTRIM('문자열')

문자열 중 왼쪽의 공백을 없앤다.

RTRIM('문자열')

문자열 중 오른쪽의 공백을 없앤다.

TRIM('문자열')

양쪽 모두의 공백을 없앤다.

LCASE('문자열') or LOWER('문자열')

소문자로 바꾼다.

UCASE('문자열') or UPPER('문자열')

대문자로 바꾼다.

REVERSE('문자열')

문자열을 반대로 나열한다.

LENGTH('문자열')

문자열의 길이

3. 논리관련 함수 및 기능

case

When 조건 then

When 조건 then

ELSE

end

IF(논리식, 참일 때 값, 거짓일 때 값)

논리식이 참이면 참일 때 값을 출력하고 논리식이 거짓이면 거짓일 때 출력한다.

IFNULL(값1,값2)

값1이 NULL 이면 값2로 대치하고 그렇지 않으면 값1을 출력

COALESCE(값1, 값2)

값들 중 처음으로 null이 아닌 값을 리턴 전체가 null이면 null 리턴

4. 날짜관련 함수

NOW() or SYSDATE() or CURRENT_TIMESTAMP()	현재 날짜와 시간 출력
CURDATE() or CURRENT_DATE()	현재 날짜 출력
CURTIME() or CURRENT_TIME()	현재 시간 출력
DATE_ADD(날짜,INTERVAL 기준값)	날짜에서 기준값 만큼 더한다.

※ 기준값 : YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

DATE_SUB(날짜,INTERVAL 기준값)	날짜에서 기준값 만큼 뺀다.

※ 기준값 : YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

YEAR(날짜)	날짜의 연도 출력.
MONTH(날짜)	날짜의 월 출력.
MONTHNAME(날짜)	날짜의 월을 영어로 출력.
DAYNAME(날짜)	날짜의 요일일 영어로 출력.
DAYOFMONTH(날짜)	날짜의 월별 일자 출력.
DAYOFWEEK(날짜)	날짜의 주별 일자 출력(일요일(1),월요일(2)…토요일(7))
WEEKDAY(날짜)	날짜의 주별 일자 출력(월요일(0),화요일(1)...일요일(6))
DAYOFYEAR(날짜)	일년을 기준으로 한 날짜까지의 날 수.
WEEK(날짜)	일년 중 몇 번째 주.
FROM_DAYS(날 수)	00년 00월 00일부터 날 수 만큼 경과한 날의 날짜 출력.
TO_DAYS(날짜)	00 년 00 월 00일 부터 날짜까지의 일자 수 출력.

DATE_FORMAT(날짜,'형식') : 날짜를 형식에 맞게 출력

타입	기호	설명	기호	설명
년도	%Y	4자리 연도	%y	2자리 년도
월	%M %b	긴 월 이름 (January, ...) 짧은 월 이름(Jan, ...)	%m %c	숫자의 월 (01...12) 숫자의 월 (1...12)
요일	%W	긴 요일 이름 (Sunday, ...)	%a	짧은 요일 이름 (Sun, ...)
일	%D %w	월 내에서 서수 형식의 일 (1th, ...) 숫자의 요일 (0=Sunday, ...)	%d %e %j	월 내의 일자 (01...31) 월 내의 일자 (1...31) 일년 중의 날수 (001...366)
시	%l %h %I	12시간제의 시 (1...12) 12시간제의 시 (01...12) 12시간제의 시 (01...12)	%k %H	12시간제의 시 (0...23) 12시간제의 시 (00...23)
분	%i	숫자의 분 (00...59)		
초	%S	숫자의 초 (00...59)	%s	숫자의 초 (00...59)
시간	%r	12시간제의 시간 (hh:mm:ss AM 또는 PM)	%T	24시간제의 시간 (hh:mm:ss)
주	%U	일요일을 기준으로 한 주 (0...52)	%u	월요일을 기준으로 한 주 (0...52)
기타	%%	문자 '%'	%p	AM 또는 PM

5. 집계 함수

- 지정한 group에 함수 기능을 적용하여 처리하여 group당 1개가 조회 된다

- group by로 그룹을 지정하지 않으면 전체 데이타가 한개의 group으로 지정 된다.

COUNT(*)	테이블의 모든 레코드 수를 구한다.
COUNT(필드명)	NULL 값이 아닌 레코드 수를 구한다.
SUM(필드명)	필드명의 합계를 구한다.
AVG(필드명)	필드명의 평균값을 구한다.
MAX(필드명)	최대값을 구한다.
MIN(필드명)	최소값을 구한다.

■ Group by

- group함수를 Group by절과 함께 사용하면 레코드를 group by절에서 지정한 컬럼의 데이타를 기준으로 분류해서 group을 나눈 후 group함수 적용한다. group함수의 결과는 group의 개수만큼 조회된다.

  형식] group by 컬럼명; group 별 데이터가 조회된다.

  형식] group by 컬럼명 with rollup; group 별 데이터와 전체(또는 중간)집계 결과가 조회된다.

6. 기타함수

DATABASE()

현재의 데이터베이스 이름을 출력한다.

PASSWORD('문자열')

문자열을 암호화한다.

FORMAT(숫자,소수이하자리수)

숫자를 #,###,###.## 형식으로 출력

■ 데이터 타입

형태	데이터형	범위	크기
숫자형	TINYINT	-128 ~ 128 , 0 ~ 255	1 Byte
	SMALLINT	-32768 ~ 32767 , 0 ~ 65535	2 Byte
	MEDIUMINT	-8388608 ~ 8388607, 0 ~ 16777215	3 Byte
	INT, INTEGER	-2147483648 ~ 2147483647, 0 ~ 4294967295	4 Byte
	BIGINT	-9223372036854775808 ~ 9223372036854775807 0 ~ 18446744073709551615	8 Byte
	FLOAT	-3.402823466E+38 ~ -1.175494351E-38 1.175494351E-38 ~ 3.402823466E+38	4 Byte
	DOUBLE [PRECISION], REAL	1.7976931348623157E+308 ~ -2.2250738585072014E-308 0 ~ 2.2250738585072014E-308	8 Byte
	DECIMAL(M,D), NUMERIC(M,D)	데이터 베이스 설정 및 시스템에 따라 다름	가변적 크기
날자형	DATE	'1000-01-01' ~ '9999-12-31'	3 Byte
	TIME	'-838:59:59' ~ '838:59:59'	3 Byte
	DATETIME	'1000-01-01 00:00:00' ~ '9999-12-31 23:59:59'	8 Byte
	TIMESTAMP	'1970-01-01 00:00:01' ~ '2038-01-19 03:14:07'	4 Byte
	YEAR	1901 ~ 2155	1 Byte
문자(열)형	CHAR(M)	1~ 255 개의 문자	M <= 255 Byte
	BINARY(M)	1 ~ 255 개의 문자	M Byte
	VARCHAR(M), VARBINARY(M)	1 ~ 255 개의 문자	M Byte
	TINYBLOB, TINYTEXT	최대 2^8	입력된 길이 만큼
	BLOB, TEXT	최대 2^16	입력된 길이 만큼
	MEDIUMBLOB, MEDIUMTEXT	최대 2^24	입력된 길이 만큼
	LONGBLOB, LONGTEXT	최대 2^32	입력된 길이 만큼
	ENUM	최대 65525 개	1 ~ 2 Byte
	SET	최대 64 개의 셋	1 ~ 8 Byte

■ DDL (Data Definition Language)

- Database 에서 사용하는 다양한 객체를 생성, 삭제, 변경하는 문
- 수행하면 DB 에 바로 반영됨
- 종류 : table, index, sequence(mysql 인 경우 sequence 테이블을 생성하거나 auto_increment 사용), view, synonym 등
- 생성 : create 객체 객체명;

  ex) create index board_no;
- 삭제 : drop 객체 객체명;

  ex) drop index board_no;

1. Table 생성

   형식] create table 테이블이름(

   컬럼명 데이터타입 [[CONSTRAINT 제약조건이름] 컬럼레벨 제약조건]

   , …

   , [[CONSTRAINT 제약조건이름] 테이블레벨 제약조건]);

   ▲제약 조건

   Primary key(주키)

   - 행 데이타를 구별할 목적으로 사용하는 키
   - Primary key 로 사용될 컬럼의 데이터는 중복되지 않고 null 이 없어야 한다.
   - 주의점 :
   - 중복된 데이터를 저장하거나 null 데이터를 저장하면 error 발생

   Unique

   - 중복되지 않은 데이터만 저장할 경우
   - 주의점 : 중복된 데이터를 저장하면 error 발생

   Not null

   - null 이 없는 데이터
   - null 데이터를 저장하면 error 발생
   - 컬럼 레벨에서만 설정할 수 있다.

   Foreign key (외래키)

   - 다른 테이블의 primary key(부모키)컬럼의 데이터를 참조
   - 자식(참조) 테이블 : 데이터를 참조하는 테이블
   - 외래키로 설정된 field 는 부모키에 없는 데이터를 입력하면 오류 발생

- 부모키를 참조하고 있는 외래키에 데이터가 있다면 부모키 삭제시 오류 발생

  ⇨ 옵션에 따라 다르다.

  ⇨ 옵션 종류

  delete cascade : 부모키 삭제시 외래키에 해당하는 모든 행을 삭제

  delete set null : 부모키 삭제시 외래키에 null 값이 설정됨

형식]

컬럼 레벨 : [constraint foreign key 외래키명] references 참조할테이블(참조할 주키)[on 옵션]

테이블 레벨

[constraint] foreign key [외래키명] (컬럼명) references 참조할테이블(참조할 주키)[on 옵션]

Check

- Insert, update 시 데이터 조건에 맞는지 검사
- 조건 맞지 않으면 error 발생

Defult

- 값이 입력되지 않거나 check 조건에 맞지 않으면 기본적으로 설정되는 값

2. 테이블 복제

Create table 테이블명 as select 문

Ex)

Create table s_emp as select empno, ename, sal from emp;

3. Table 변경

alter table 테이블명(add | modify , change, drop) 컬럼이름

- 컬럼 추가

  alter table 테이블명 add 컬럼명 타입;

  Ex) s_emp 테이블에 deptno 추가

  alter table s_emp add deptno int;

- 컬럼 타입 변경

  alter table 테이블명 modify 컬럼명 타입;

  ex) alter table s_emp modify deptno varchar(30);

- 컬럼명, 타입 변경

  alter table 테이블이름 change 이전컬럼이름 변경할_컬럼이름 변경할_타입;

  ex) alter table s_emp change deptno address varchar(200);

- 컬럼 삭제

  alter table 테이블이름 drop 칼럼이름

  ex) alter table s_emp drop address;

4. Table 삭제

- 부모테이블인 경우 삭제가 안됨.

  => 자식 테이블 삭제후 부모 테이블을 삭제해야 한다.

- 형식]

  drop table 테이블명

5. Truncate

- 테이블의 모든 데이타를 삭제
- 복구 할 수 없다.
- 형식] truncate 테이블명;

■ DML (Data Manipulation Language)

- data 를 insert, update, delete 하는 문장
- DML 문을 수행하면 수행된 내용이 임시저장소에 저장되어 실제 DB 에는 반영되지 않는다.
- DML 문을 수행 후 TCL(Transaction controll language)을 수행해야 DB 에 반영되거나 수행된 내용이 취소된다.
- record lock

  insert, update, delete 문을 수행하면 commit 또는 rollback 하기 전까지 해당 record 는 lock 된다.

TCL 문

commit	: 작업한 모든 dml 문을 디비에 반영
savepoint 이름	: 임시 저장 위치
rollback	: 작업한 모든 dml 문 수행 취소
rollback to 세이브포인트이름	: 지정한 savepoint 까지 취소

1. Insert 문

insert into 테이블명(컬럼명, ...) values(값,...)

- 테이블에 지정한 컬럼에 지정한 값으로 insert 된다.

insert into 테이블명 values(값,...)

- 테이블 구조에 설정된 모든 컬럼에 구조에 지정된 순서대로 값이 insert 된다.

2. Update 문

- 전체 데이타 수정

  updata 테이블명 set 컬럼명=value;

- 조건에 맞는 데이타만 수정

  updata 테이블명 set 컬럼명=value where 조건;

  => mysql workbench 인 경우 기본키에 해당하는 레코드를 수정하는 경우가 아니면 처리 안되게 설정되어 있다.

  설정 해제방법 edit=> preference => sql editor => safe update 체크 해제

3. Delete 문

- 전체 데이타 삭제

  delete from 테이블명;

- 조건에 맞는 데이타만 삭제

  delete from 테이블명 where 조건;

  => mysql workbench 인 경우 기본키에 해당하는 레코드를 삭제하는 경우가 아니면 처리 안되게 설정되어 있다.

  설정 해제방법 edit=> preference => sql editor => safe update 체크 해제