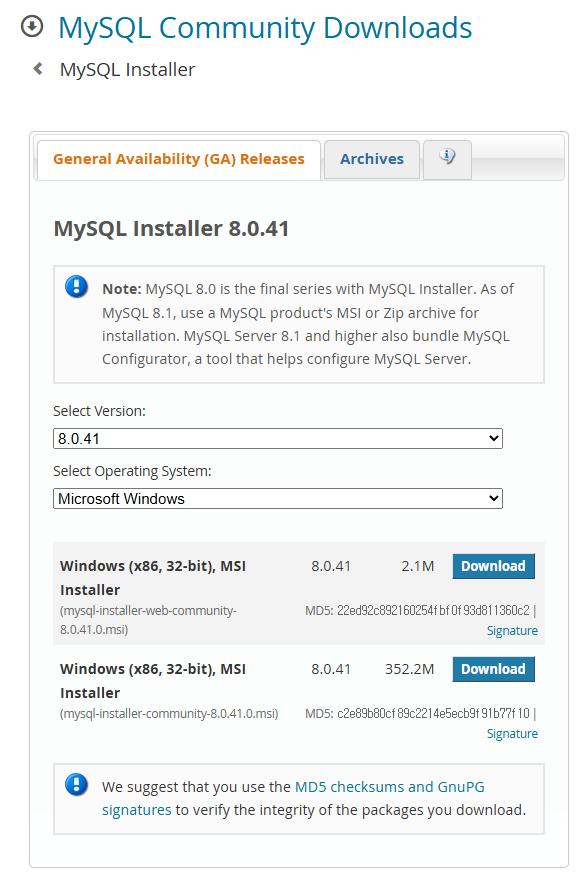
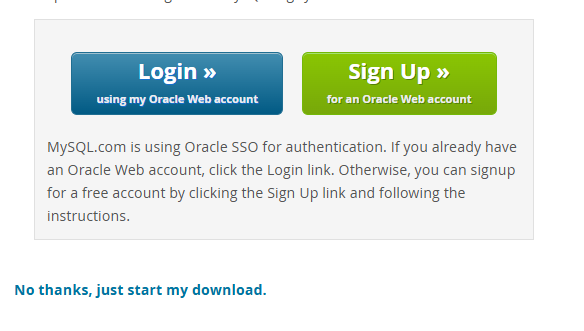
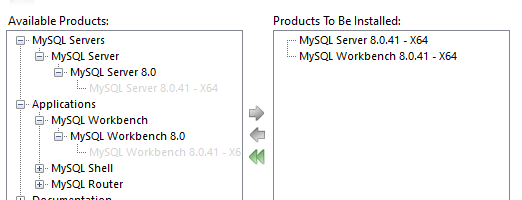
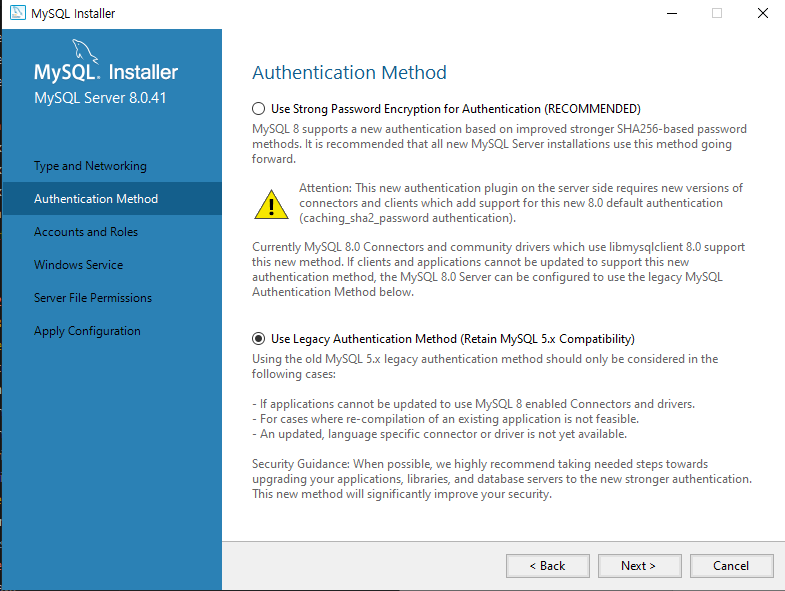
**MySQL 설치**

1. 구글 – mysql 검색
2. MySQL Installer8.0.41









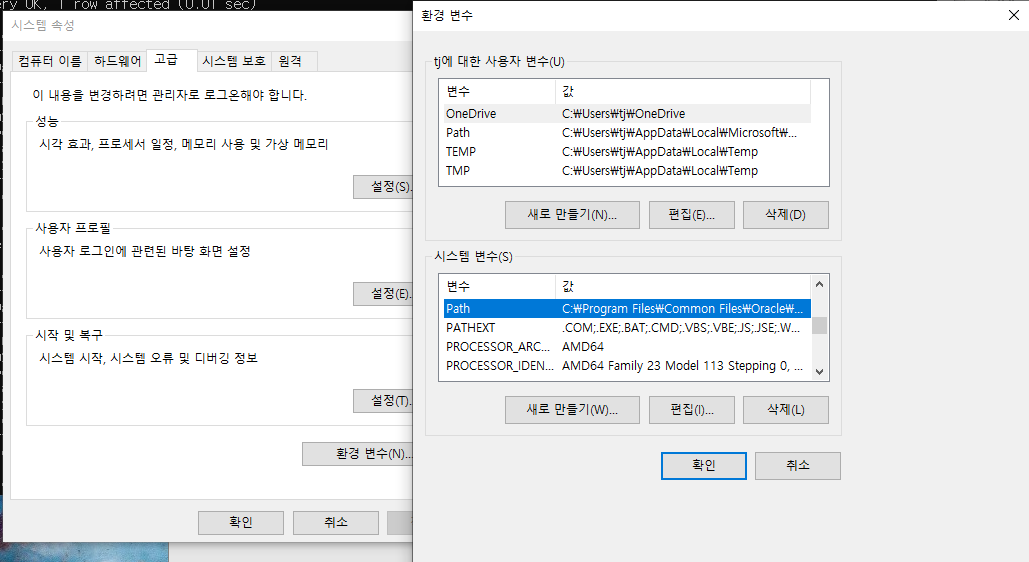
Password : 1234

나머지는 전부 next

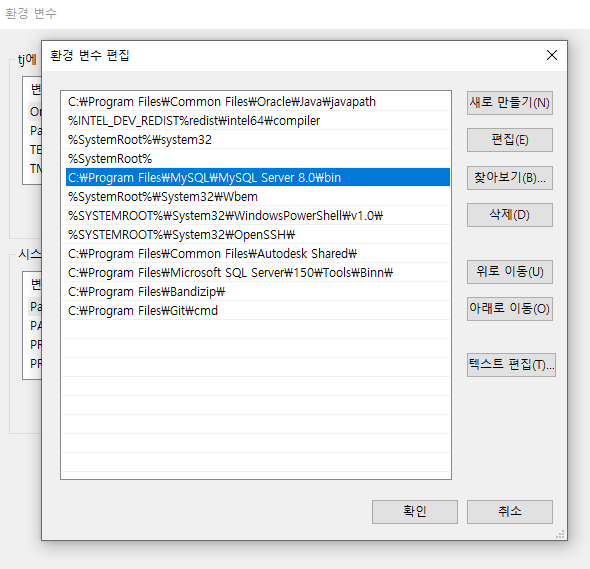
* **환경변수 설정**

검색 – 시스템 환경 변수 편집 – 환경변수

Path – 편집



새로 만들기 – 찾아보기 – MySQL의 bin 추가



**cmd에서 MySQL 접속**

mysql -u root -p

password 1234 입력

* **대소문자 구분 없음**

show databases; -> 데이터베이스 목록 보기

create database test; -> 데이터베이스 생성

drop database test; -> 데이터 베이스 삭제 (drop database ‘db명’)

use sample; -> 데이터 베이스 사용(선택) (use ‘db명’)

show tables; -> use 샘플 데이터베이스 안의 테이블 보기

**회원 테이블 만들기**

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL PRIMARY KEY, # primary key로 설정

mem\_name VARCHAR(10) NOT NULL,

mem\_number TINYINT NOT NULL,

addr CHAR(2) NOT NULL,

phone1 CHAR(3) NULL,

phone2 CHAR(8) NULL,

height TINYINT UNSIGNED NULL,

debut\_date DATE NULL

PRIMARY KEY(num); -> 따로 유일키 지정해도 됨

);

Tinyint : 자바에 byte와 비슷(-128~127)

Tinyint unsigned : 음수 불가. 양수 범위가 두 배로 커짐. (0~255)

Primary key : 유일키

INSERT INTO member(name, tel, address) VALUES ( , , ); -> 선택해서 삽입

INSERT INTO member VALUES( , , );

* 앞에 auto\_increment가 있을 경우, 그 자리에 null 추가해줘야 함(다른 값 넣으면 그 값이 들어감)

DESC member; -> 테이블 구조 보여줌

SELECT \* FROM member; -> 모든 내용 보여줌

SELECT mem\_id, mem\_name FROM member; -> member 테이블에서 mem\_id, mem\_name 만 보여줌

RENAME TABLE bye TO buy; -> 테이블 이름을 bye->buy로 변경

ALTER TABLE member ADD post\_num CHAR(20); -> 맨 뒤에 필드 추가

ALTER TABLE member ADD post\_num CHAR(20) AFTER mem\_number; -> mem\_nember 뒤에 필드 추가

ALTER TABLE member DROP post\_num; -> post\_num 필드 삭제

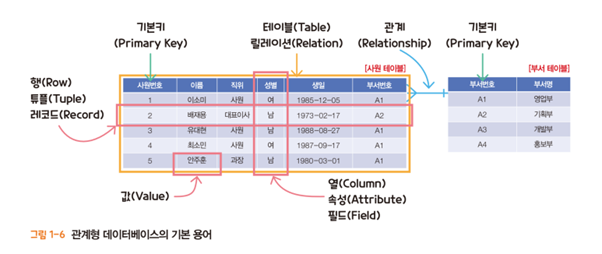
ALTER TABLE member CHANGE phone1 tel1 char(4); -> phone1을 tel1으로 바꿈

ALTER TABLE person ADD COLUMN agee double; -> column 추가

ALTER TABLE person CHANGE COLUMN agee age double; -> column 명 변경

ALTER TABLE person MODIFY COLUMN age int; -> 데이터 타입을 int로 변경

ALTER TABLE person DROP COLUMN age; -> age column 삭제



**조건식 조회**

SELECT id, name, address, gender FROM member WHERE gender='w';

SELECT name, tel FROM member WHERE name LIKE ‘박%’;

* ‘박’으로 시작하는 모든 이름
* Not like – 포함하지 않는

**레코드 정렬**

SELECT 필드명1, 필드명2 FROM 테이블명 ORDER BY 필드명;

* 필드명 순으로 오름차순 정렬

SELECT 필드명1, 필드명2 FROM 테이블명 ORDER BY 필드명 DESC;

* 내림차순 정렬

**레코드 수정**

UPDATE 테이블명 SET 필드명=필드값 WHERE 조건식;

UPDATE member SET age=37 WHERE name=’정한나’;

**레코드 삭제**

DELETE FROM 테이블명 WHERE 조건식

DELETE FROM member WHERE name=’김수련’;

DELETE FROM member; -> 전체 레코드 삭제

C:\Users\tj>mysqldump -uroot -p1234 test > test.sql

* 저 위치에 test 데이터베이스가 test.sql 파일로 저장됨

DROP TABLE IF EXISTS `person`; -> person 테이블이 이미 있으면 지워라

SELECT 고객번호, 담당자명, 고객회사명, 마일리지 as 포인트, 마일리지\*1.1 as "10% 인상된 마일리지" from 고객;

* 마일리지\*1.1 가능. 마일리지를 "10% 인상된 마일리지"로 변경 가능.

select 고객번호, 담당자명, 도시, 마일리지 as 포인트 from 고객 where 도시='서울특별시' order by 마일리지 desc;

**Limit : 개수 제한**

Select \* from 고객 limit 3; ->고객테이블에서 1행부터 시작하여 3개의 고객정보를 조회

Select \* from 고객 limit 10, 3; -> 10행부터 3개 보여줌

Select \* from 고객 order by 마일리지 desc limit 3; -> 마일리지 많은 3명

Select **distinct** 도시 from 고객; -> 중복 데이터 제거

Select 23+5 as 더하기,

23-5 as 빼기,

5\*2 as 곱하기,

23/5 as 실수나누기,

23 div 5 as 정수나누기,

23%5 as 나머지1,

23 mod 5 as 나머지2;

Select 23>=5, 23<=5, 5>5 , 23<23, 23=23, 23!=5, 23<>23;

* 1, 0, 0, 0, 1, 1, 0

select \* from 고객 where 담당자직위 != '대표 이사';

select \* from 고객 where 담당자직위 <> '대표 이사';

select \* from 고객 where 지역=""; -> 빈문자열

select \* from 고객 where 지역 is null; -> null

**빈 문자열을 null로 변경할 건데 안전모드가 걸려 있는 경우(Error 1175)**

방법1)

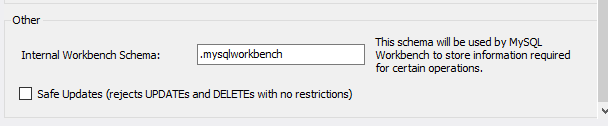
Set SQL\_SAFE\_UPDATES = 0;

Update 고객 set 지역=null where 지역=’’ and 고객번호 is not null;

* 빈 문자열인 지역을 null로 변경. 근데 update가 안전 모드로 설정되어 있어서 안 됨. 그래서 안전모드를 0으로 바꾸고 null로 변경.
* 끝나면 set SQL\_SAFE\_UPDATES = 1; 해서 원래대로 변경

방법2)

Edit> Preferences > SQL Editor > Safe Updates 옵션 체크 해제 후 프로그램 재실행



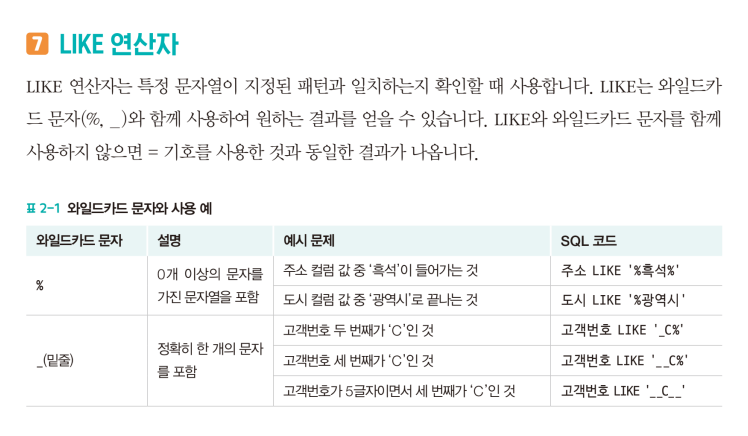
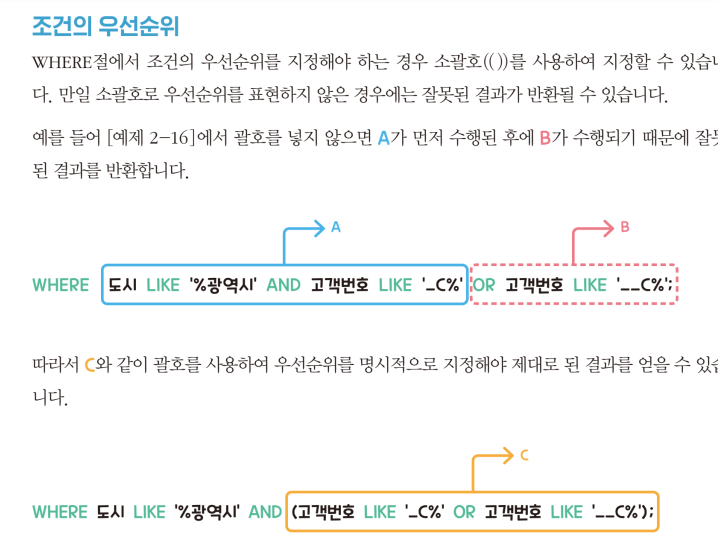
SELECT고객번호, 담당자명, 담당자직위 FROM 고객 WHERE 담당자직위 IN ('영업 과장','마케팅 과장'); -> or 조건이 여러 개일 경우 **IN** 사용

select 담당자명, 마일리지 from 고객 where 마일리지 BETWEEN 100000 AND 200000 ;

* 10만 ~ 20만

SELECT \* FROM 고객 WHERE 도시 LIKE '%광역시' AND (고객번호 like '\_C%' or 고객번호 like '\_\_C%'); -> 두 번째 글자가 C인 경우 or 세 번째 글자가 C인 경우

조건의 우선순위



글자의 길이

SELECT CHAR\_LENGTH('HELLO') // 영문자 개수 반환. 5

,LENGTH('HELLO') // 영문자 바이트 수. 5

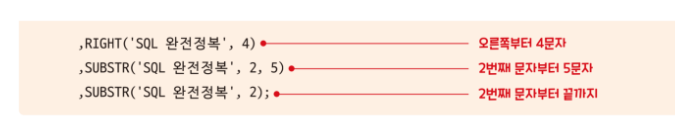
,CHAR\_LENGTH('안녕') // 문자 개수. 2

,LENGTH('안녕'); // 문자 바이트 수. 5

SELECT CONCAT(‘DREAMES’, ‘COME’, ‘TRUE’) // 붙어서 나옴. DREAMESCOMETRUE

, CONCAT\_WS(‘-‘, ‘2025’, ‘04’, ‘10’); // 2025-04-10

SELECT LEFT('SQL 완전정복', 3); // 왼쪽부터 3글자만 추출



**SUBSTR(문자열, 시작위치, 길이) OR SUBSTRING(문자열, 시작위치, 길이)**

SELECT SUBSTRING\_INDEX('서울시 동작구 흑석로', ' ', 2) // 서울시 동작구

,SUBSTRING\_INDEX('서울시 동작구 흑석로', ' ', -2); // 동작구 흑석로

**LPAD(문자열, 길이, 앞 채울 문자열) OR RPAD(문자열, 길이, 뒤 채울 문자열)**

SELECT LPAD('SQL',10,'#') // 문자 포함 10개. 앞 공백은 #으로 채움

,RPAD('SQL',5,'\*'); // 문자 포함 5개. 뒤 공백은 \*로 채움

SELECT LTRIM(‘ SQL ‘), RTRIM(‘ SQL ‘), TRIM(‘ SQL ‘) // 왼쪽, 오른쪽, 양쪽 공백 제거

SELECT TRIM(BOTH 'abc' FROM 'abcSQLabcabc') // SQL. 모든 abc 제거

,TRIM(LEADING 'abc' FROM 'abcSQLabcabc') // SQLabcabc. 왼쪽

,TRIM(TRAILING 'abc' FROM 'abcSQLabcabc'); // abcSQL. 오른쪽

SELECT FIELD('JAVA', 'SQL', 'JAVA', 'C') // 앞의 JAVA를 뒤의 문자 중에서 찾기

,FIND\_IN\_SET('JAVA', 'SQL,JAVA,C') // 앞의 JAVA를 ,로 구분된 뒤에서 찾기

,INSTR('네 인생을 살아라', '인생') // 앞 문장 중 뒤의 단어 시작 위치 찾기

,LOCATE('인생', '네 인생을 살아라'); // 앞 문자를 뒤의 문장에서 시작위치 찾기

SELECT REPEAT('\*',5);// \* 5개 반복

SELECT REPLACE('010.1234.5678', '.', '-'); // 맨 앞 문자열의 . 을 – 로 바꾸기

SELECT CEILING(123.45) // 올림

, FLOOR(123.55) // 내림

,ROUND(123.55) // 반올림

, ROUND(123.56,1) // 소수 첫째 자리까지 반올림

,TRUNCATE(123.55,1); // 소수 첫째 자리까지 내림

SELECT MOD(203, 4), 203 % 4 , 203 MOD 4; // 나머지

SELECT POWER(2,3) // 거듭제곱 (2의 세제곱 = 8)

, SQRT(16) // 제곱근 (루트 16 = 4)

, RAND() // 랜덤값

, RAND(100)

, ROUND(RAND() \* 100) // 0~100

**날짜**

SELECT NOW(); // 현재 날짜 시간

SELECT SYSDATE(), // 현재 날짜 시간

CURDATE(), // 현재 날짜

CURTIME(); // 현재 시간

SELECT NOW()

,YEAR(NOW()) // 연도

,QUARTER(NOW()) // 분기

,MONTH(NOW())

,DAY(NOW())

,HOUR(NOW())

,MINUTE(NOW())

,SECOND(NOW());

SELECT NOW()

,DATEDIFF('2026-12-20',NOW()) // 앞 날짜 – 뒤 날짜

, DATEDIFF(NOW(),'2026-12-20')

, TIMESTAMPDIFF(YEAR, NOW(), '2026-12-20') // 뒤 날짜 – 앞 날짜

, TIMESTAMPDIFF(MONTH, NOW(), '2026-12-20') // 20개월 남음

, TIMESTAMPDIFF(DAY, NOW(), '2026-12-20');

SELECT NOW()

, ADDDATE(NOW(), 50) // 50일 후

, ADDDATE(NOW(), INTERVAL 50 DAY) // 50일 후

, ADDDATE(NOW(), INTERVAL 50 MONTH)

, SUBDATE(NOW(), INTERVAL 50 HOUR); // 50시간 전

SELECT NOW()

, LAST\_DAY(NOW()) // 이번 달의 마지막 날짜

, DAYOFYEAR(NOW()) // 1년 중 며칠 째인지

, MONTHNAME(NOW()) // 이번 달 이름

, WEEKDAY(NOW()); // 몇 번째 요일(월요일:0)

SELECT CAST('1' AS UNSIGNED) // 문자를 숫자로 캐스팅(변경)

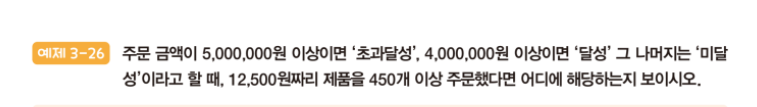
, CAST(2 AS CHAR(1)) // 숫자를 문자로

, CONVERT('1' , UNSIGNED) // 문자를 숫자로

, CONVERT(2 , CHAR(1)); // 숫자를 문자로

**IF(조건식, 참, 거짓)**

SELECT IF(1 \* 4 > 50, '참', '거짓');



SELECT CASE WHEN 12500\*450 > 5000000 THEN '초과달성'

WHEN 12500\*450 > 4000000 THEN '달성'

ELSE '미달성'

END;

SQL문의 종류

1. 데이터 정의어(DDL, Data Definition Language)

데이터 베이스나 테이블, 인덱스와 같은 데이터베이스 객체를 생성, 변경, 삭제할 때 쓰임

* Create : 데이터베이스 객체 생성
* Alter : 생성된 객체의 구조 변경
* Drop : 생성된 객체를 삭제
* Truncate : 테이블에 있는 데이터 모두 삭제

1. 데이터 조작어(DML, Data Management Language)

* 데이터 관리. 레코드 추가, 삭제, 변경할 때 쓰임
* Select : 테이블에서 조건에 맞는 데이터 검색
* Insert : 테이블에 새로운 레코드 삽입
* Update : 테이블에 있는 데이터 값을 수정
* Delete : 테이블에 있는 데이터 삭제

1. 데이터 제어어(DCL)

* GRANT : 특정 사용자 또는 특정 객체에 대해 생성, 수정 등 특정 작업에 권한 부여
* Revoke : grant문으로 부여한 권한을 철회할 때 사용

1. TCL

* Commit : 트랜잭션 작업 내용을 실제 데이터베이스에서 영구 저장
* Rollback : 트랜잭션 처리과정에서 발생한 변경 사항을 취소