**[ MySQL ]**

1. MySQL의 특징

* SQL에 기반을 둔 관계형 DBMS 중 하나
* Oracle, Infomix 등의 데이터베이스는 고가이지만 MySQL 은 무료(배포판)
* 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용가능
* 처리 속도가 상당히 빠르고 대용량의 데이터 처리 용이
* 설치방법이 쉽고 초보자도 익히기 쉬움.
* 보안성이 우수

1. MySQL 설치
   * <http://dev.mysql.com/downloads/mysql/> 운영체제에 맞는 MySQL을 다운받는다.
   * 수업시간에는 mysql installer community 8.0.39. msi버전을 사용한다.
   * 다운받은 설치파일을 실행하여 “full”로 설치한다. MySQL Workbench 확인
   * 업그레이드 적용 메시지가 나올 경우 “No” 선택
2. MySQL의 시작
   * C:\Program Files\MySQL\MySQL Server 8.0\bin에서 cmd실행하여 mysql –u계정 -p비밀번호 실행 : 관리자 계정 root로 접속 : mysql -uroot –pmysql
   * 사용자 계정 생성 및 권한 부여
     + 계정 생성 create user user01 identified by ‘password’;
     + 권한 부여 grant all on \*.\* to user01;

grant all privileges on \*.\* to user01;

flush privileges;

* + - 권한 박탈 revoke all on \*.\* from user01;
    - 계정 삭제 drop user user01;
  + 데이터베이스
    - 존재하는 데이터베이스들 목록 보기 : show databases;
    - 새로운 데이터베이스 생성 명령 : create database 데이터베이스명;

反 : Drop database 데이터베이스명

* + - 데이터베이스로 들어가기 : use 데이터베이스명
    - 내가 어디서 작업하는지 모를 때 select database();
  + 실습
    - show databases;
    - use mysql;
    - show tables;
    - create database kimdb;
    - drop database kimdb;
    - create database kimdb;
    - use kimdb;
    - select database(); -- 내가 어디서 작업하는지 모를 때
    - alter database kimdb charset = utf8; -- 한글이 깨질 경우
    - ALTER TABLE table\_name CHARSET = utf8;

1. DDL
   * 테이블 생성
     + MySQL 자료형

Tinyint, smallint, int, integer, numeric(m,d), double, char, varchar, BLOB, date, datetime, timestamp, time

* + - 제약조건

Unique, primary key, not null, foreign key, check, 그외 제약조건은 아니나 default

ex. drop table if exists emp;

create table emp(

empno numeric(4) primary key,

ename varchar(20) not null,

nickname varchar(20) unique,

sal numeric(7,2) check(sal>0),

hiredate DATETIME default now(),

comm numeric(7,2) default 0);

create table emp(

empno numeric(4),

ename varchar(20) not null,

nickname varchar(20),

sal numeric(7,2) check(sal>0),

hiredate DATETIME default now(),

comm numeric(7,2) default 0,

primary key(empno),

unique(nickname),

check(sal>0));

insert into emp (empno, ename, nickname, sal) values (1111, '홍', '홍홍', 7000.71);

select \* from emp;

commit;

ex. 학과테이블 : 학과번호, 학과이름

drop table if exists major;

create table major(

major\_code int primary key auto\_increment, -- auto increment일 경우 int타입

major\_name varchar(30));

select \* from major;

insert into major (major\_name) values ('경영정보');

insert into major (major\_name) values ('컴퓨터공학');

insert into major (major\_name) values ('통계학');

insert into major (major\_name) values ('빅데이터');

ex. 학생테이블 : 학번, 이름, 학과번호

create table STUDENT(

stu\_id int not null primary key,

name varchar(15),

major\_code int references major (major\_code));

desc STIDENT;

insert into student values (1001, '홍길동', 1);

insert into student values (1002, '김길동', 2);

insert into student values (1003, '신길동', 3);

SELECT \* FROM STUDENT;

SELECT \* FROM STUDENT S, MAJOR M

WHERE S.MAJOR\_CODE=M.MAJOR\_CODE;

insert into student values (1009, '홍', 9);

SELECT STU\_ID, NAME, MAJOR\_NAME

FROM STUDENT S LEFT OUTER JOIN MAJOR M

ON S.major\_code=M.major\_code;

select s.student\_id, s.student\_name, major\_name

from student s right outer join major m

on s.major\_code=m.major\_code;

drop table if exists student;

create table student (

stu\_id int not null primary key,

name varchar(15),

major\_code int,

foreign key(major\_code) references major (major\_code));

insert into student values (1001, '홍길동', 1);

select \* from student;

drop table student;

drop table major; -- parent table 먼저 drop 불가

* 이하는 JDBC(java-dbms연동)때 쓸 tables

drop table if exists division;

create table division(

DNO int not null primary key,

dname varchar(20),

phone varchar(20),

position varchar(20));

show tables;

drop table if exists personal;

create table personal (

pno int primary key,

pname varchar(10) not null,

job varchar(15) not null,

manager int,

startdate date,

pay int,

bonus int,

dno int ,

foreign key(dno) references division(dno));

show tables;

desc division;

insert into division values (10, 'finance','02-777-7777','종로');

insert into division values (20, 'research','041-888-7777','대전');

insert into division values (30, 'sales','02-999-7777','인천');

insert into division values (40, 'marketing','02-555-7777','강남');

select \* from division;

load data local infile 'd:/mega-IT/ personal.txt' replace into table personal;

select \* from personal;

select pno, pname '이름' from personal;

truncate personal;

load data local infile 'd:/mega\_IT/personal.txt' replace into table personal; -- MySql 5.x에서 가능

insert into personal values (1111,'smith','manager', 1001, '1990-12-17', 1000, null, 10);

insert into personal values (1112,'ally','salesman',1116,'1991-02-20',1600,500,30);

insert into personal values (1113,'word','salesman',1116,'1992-02-24',1450,300,30);

insert into personal values (1114,'james','manager',1001,'1990-04-12',3975,null,20);

insert into personal values (1001,'bill','president',null,'1989-01-10',7000,null,10);

insert into personal values (1116,'johnson','manager',1001,'1991-05-01',3550,null,30);

insert into personal values (1118,'martin','analyst',1111,'1991-09-09',3450,null,10);

insert into personal values (1121,'kim','clerk',1114,'1990-12-08',4000,null,20);

insert into personal values (1123,'lee','salesman',1116,'1991-09-23',1200,0,30);

insert into personal values (1226,'park','analyst',1111,'1990-01-03',2500,null,10);

delete from personal where pno=1001;

* + 테이블 변경
    - alter table 테이블명 add 필드명 타입;

alter table 테이블명 modify 필드명 타입;

alter table 테이블명 drop 필드명;

* + 테이블 삭제
    - drop table table\_name;

1. DML (Query - Auto-Commit Transections체크되어 있으면 autocommit됨)
   * select문 (and 연산자는 && 또는 and)

-- 1. 사번, 이름, 급여를 출력

-- 2. 급여가 2000~5000 사이 모든 직원의 모든 필드

-- 3. 부서번호가 10또는 20인 사원의 사번, 이름, 부서번호

-- 4. 보너스가 null인 사원의 사번, 이름, 급여 급여 큰 순정렬

-- 5. 사번, 이름, 부서번호, 급여. 부서코드 순 정렬 같으면 PAY 큰순

-- 6. 사번, 이름, 부서명

-- 7. 사번, 이름, 상사이름

-- 8. 사번, 이름, 상사이름(상사가 없는 사람도 출력하되 상사가 없는 경우 ★CEO★로 출력) – 우선

-- 9. 이름이 s로 시작하는 사원 이름 (like 이용, substr함수이용, instr함수 이용등 다양하게 사용 가능)

-- 10. 사번, 이름, 급여, 부서명, 상사이름

-- 11. 부서코드, 급여합계, 최대급여

-- 12. 부서명, 급여평균, 인원수

-- 13. 부서코드, 급여합계, 인원수 인원수가 4명 이상인 부서만 출력

-- 14. 사번, 이름, 급여 회사에서 제일 급여를 많이 받는 사람

-- 15. 회사 평균보다 급여를 많이 받는 사람 이름, 급여, 부서번호

-- 16. 회사 평균 급여보다 많이 받는 사원의 사번, 이름, 급여, 부서명을 출력(부서명순 정열 같으면 급여 큰순 정렬)

-- 17. 자신이 속한 부서의 평균보다 많인 받는 사람의 이름, 급여, 부서번호, 반올림한 해당부서평균 (where절의 subQuery를 이용하거나 inline view를 이용)

-- 18. 입사가 가장 빠른 사람의 이름, 급여, 부서명

-- 19. 이름, 급여, 해당부서평균(select절의 subQuery를 이용하거나 inline view를 이용)

-- 20. 이름, 급여, 부서명, 해당부서평균(select절의 subQuery를 이용하거나 inline view를 이용)

* + 컬럼함수 – 오라클 함수와 다소 다른 점
    - select concat(pname, '(',job, ')') from personal; -- 오라클은 2개까지만. ||은 OR의 의미 연결연산자가 아니다.
    - select round(114.7, -1); -- cf) 오라클에서는 select round(114.7, -1) from dual;
    - select reverse('대박사건');
    - select reverse('HELLO');
    - select year(startdate) from personal;
    - select month(startdate) from personal;
    - select monthname(startdate) from personal;
    - select dayname(startdate) from personal; -- 요일이름 Wednesday..
    - select WEEKDAY(‘20230318’); --날짜의 주별 일자 출력(월요일은0, 화요일은1, 일요일은6)
    - select DAYOFYEAR(‘20230318’); --일년을 기준으로 한 날짜짜지의 수
    - select now();
    - select sysdate(); -- 현재 날짜 시간
    - select current\_date() – 현재 날짜만
    - select curdate();
    - select current\_time(); -- 현재 시간만
    - select curtime();
    - select date\_format(now(), '%y년%m월%e일 %a %p %h시 %i분');
    - select date\_format(now(), '%Y년%m월%e일 %a %H시 %i분');
    - select date\_format(startdata, '%y년%m월%e일') from personal;
      * %Y 4자리 연도 %y 2자리 연도
      * %M 긴월 이름(January, ) %m 숫자의 월(01,02,03..12)
      * %b 짧은 월 이름(Jan, ) %c 숫자의 월(1,2,3…12)
      * %e 월 내의 일(1,2,3..31) %d 월 내의 일자(01,02,…31)
    - select pname, pay, if(pay>3000, 'Good', 'Bad') from personal; -- 논리함수 존재
    - select pname, pay, bonus, pay+ifnull(bonus, 0) from personal; --오라클의 NVL()역할
    - select rpad(substr(pname, 1, 1), length(pname), '\*') from personal; -- 오라클과 동일
  + Top-n 역할
    - select pname, pay from personal order by pay desc;
    - select pname, pay from personal order by pay desc limit 3;
    - select pname, pay from personal order by pay desc limit 2,3; -- 순서가 0부터시작 2번째부터 3건
    - SELECT \* FROM personal ORDER BY pno desc LIMIT 4, 3;
  + Join
    - select \* from personal where dno=40;
    - select \* from division;
    - select pname, dname from personal p, division d
    - where p.dno=d.dno;
    - select pname, dname from personal p right outer join division d
    - on p.dno=d.dno;
    - select w.pname, w.manager, m.pno, m.pname from personal w left outer join personal m on w.manager = m.pno;
    - select pname, format(pay, 0) from personal;
    - select pname, format(pay, 1) from personal;
    - show tables;
    - show databases;
    - create database test;
    - use mysql;
    - show databases;
    - use test;