**[ MySQL ]**

1. MySQL의 특징

* SQL에 기반을 둔 관계형 DBMS 중 하나
* Oracle, IBM, 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
   1. <http://dev.mysql.com/downloads/mysql/> 운영체제에 맞는 MySQL을 다운받는다.
   2. MSI 버전은 닷넷 프레임워크 4.0이 설치되어 있어야 한다.
   3. 다운받은 설치파일을 실행하여 설치한다. MySQL Workbench 확인
2. MySQL의 시작
   1. C:\Program Files\MySQL\MySQL Server 5.7\bin에서 cmd실행하여 mysql –u계정 -p비밀번호 실행 : 관리자 계정 root로 접속 : mysql -uroot -pmysql
   2. 데이터베이스
      * 존재하는 데이터베이스들 목록 보기 : show databases;
      * 새로운 데이터베이스 생성 명령 : create database 데이터베이스명;

反 : Drop database 데이터베이스명

* + - 데이터베이스로 들어가기 : use 데이터베이스명
    - 권한 부여 grant all on \*.\* to kim identified by ‘1234’;
    - 권한 박탈 revoke all on \*.\* from kim;
    - 내가 어디서 작업하는지 모를 때 select database();
  1. 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;
     + ALTER TABLE table\_name CHARSET = utf8;

1. DDL
   1. 테이블 생성
      * MySQL 자료형
        + Tinyint, smallint, int, integer, float(m,d), double, char, varchar, BLOB, date, datetime, timestamp, time
      * 제약조건
        + Unique, primary key, not null, default 값, foreign key

ex. 학과테이블 : 학과번호, 학과이름

drop table if exists major;

create table major(

major\_code int primary key auto\_increment,

major\_name varchar(30));

select \* from major;

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

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

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

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));

insert into student values (1001, '홍', 9);

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;

* 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 not null primary key,

pname varchar(10) not null,

job varchar(15) not null,

manager int,

startdata 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;

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;

* 1. 테이블 변경
     + alter table 테이블명 add 필드명 타입;

alter table 테이블명 modify 필드명 타입;

alter table 테이블명 drop 필드명;

* 1. 테이블 삭제
     + drop table table\_name;

1. DML
   1. select문 (and 연산자는 && 또는 and)

-- 1. 사번, 이름, 급여를 출력

select pno, pname, pay from personal;

-- 2. 급여가 2000~5000 사이

select \* from personal where pay between 2000 and 5000;

-- 3. 부서코드가 10또는 20인 사원의 사번, 이름, 부서코드

select pno, pname, dno from personal where dno in (10,20);

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

select pno, pname, pay from personal where bonus is null order by pay desc;

-- 5. 사번, 이름, 부서코드, 급여 부서코드 순 정열 같으면 큰 순

select pno, pname, dno, pay from personal order by dno, pay desc;

-- 6. 사번, 이름, 부서명

select pno, pname, dname from personal p, division d where p.dno=d.dno;

-- 7. 사번, 이름, 관리자명

select w.pno, w.pname, m.pname 상사명

from personal w, personal m

where w.manager = m.pno;

select w.pno, w.pname, m.pname 상사명

from personal w left outer join personal m

on w.manager=m.pno;

-- 8. 이름이 s로 시작하는 사원 이름

select pname from personal where pname like 's%';

-- 9. 사번, 이름, 급여, 부서명, 관리자명

select w.pno, w.pname, w.pay, dname, m.pname 관리자명

from personal w, personal m, division d

where w.manager = m.pno **&&** w.dno=d.dno;

-- 10. 부서코드, 급여합계, 최대급여

select dno, sum(pay), max(pay)

from personal group by dno;

-- 11. 부서명, 급여평균, 인원수

select dname, avg(pay), count(\*)

from division d, personal p

where d.dno=p.dno group by dname;

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

select dno, sum(pay), count(\*)

from personal group by dno having count(\*)>=4;

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

select pno, pname, pay from personal

where pay=(select max(pay) from personal);

-- 14. 회사 평균보다 급여를 많이 받는 사람 이름, 급여

select pname, pay, dno from personal

where pay>(select avg(pay) from personal);

-- 15. 14번에 부서명을 추가하고 부서명순 정열 같으면 급여 큰순

select pname, pay, dname

from personal p, division d

where p.dno=d.dno and

pay > (select avg(pay) from personal)

order by dname, pay desc;

-- 16. 자기부서평균보다 급여를 많이 받는 사람 이름, 급여

select pname, pay from personal p

where pay>(select avg(pay) from personal

where p.dno=dno);

select \* from personal p,

(select dno, avg(pay) avgsal from personal group by dno) a

where p.pay > a.avgsal and p.dno=a.dno;

-- 17. 입사가 가장 빠른 사람의 이름, 급여, 부서명

select pname,pay,dname from personal p,division d

where p.dno=d.dno and

startdata=(select min(startdata) from personal);

* 1. 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(startdata) from personal;
     + select month(startdata) from personal;
     + select monthname(startdata) from personal;
     + select dayname(startdata) from personal; -- 요일이름 Wednesday..
     + select WEEKDAY(‘20160318’); --날짜의 주별 일자 출력(월요일은0, 화요일은1, 일요일은6)
     + select DAYOFYEAR(‘20160318’); --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(), '%y/%m/%e');
     + 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
  2. Top-n 역할
     + select pname, pay from personal order by pay desc;
     + select pname, pay from personal order by pay desc limit 5;
     + select pname, pay from personal order by pay desc limit 3,5; -- 순서가 0부터시작 3번째부터 5건
     + SELECT \* FROM personal ORDER BY pno desc LIMIT 4, 3;
  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;