|  |  |  |
| --- | --- | --- |
| 오라클 실습 | | |
|  | 오라클의 최고권한 관리자인 sys계정으로 접속하여 system 사용자의 비밀번호를 1234로 변경한 후, system계정으로 로그인 하시오  alter user system  identified by 1234; |  |
| 현재 접속한 유저명이 무엇인지 알아보는 명령어  show user; |  |
| 아래의 조건으로 데이터베이스를 생성하시오   1. TableSpace 명 : space0907 2. 파일명 : space0907.dbf 3. 파일 사이즈 : 2M   create tablespace space0907  datafile '경로\파일명.dbf'  size 2m; |  |
| 아래의 조건으로 유저를 생성하시오   1. 유저명 ; user0907 2. 비밀번호 : 1234 3. 사용할 TableSpace : space0907 4. 테이블스페이스에 대한 사용 할당 무제한   create user user0907  indentified by 1234  default tablespace space0907  quota unlimited on space0907; |  |
| 아래의 조건으로 유저에게 권한을 부여하시오   1. 접속 권한 2. 테이블 생성 권한   grant create session , create table to user0907; |  |
| user0907 계정으로 로그인 한 후, 아래의 조건에 해당하는 테이블을 생성하시오   1. 테이블명 : member 2. 컬럼 구성  * 이름 : name , 영문 20자 넣을 크기로 생성 * 나이: age , 정수형으로 생성 * 출생일 : birth , 날짜형으로 생성   create table member(  name varchar(20),  age number,  birth date  );  < 완성된 테이블 개념도 >   |  |  |  | | --- | --- | --- | | name | age | birth | |  |  |  | |  |  |  | |  |
| member 테이블에 두사람의 정보를 아래의 값으로 넣으시오   1. 이름: 노홍철, 나이:30 , 생년월일 1989-09-01 2. 이름: 박나래 , 나이:27 , 생년월일 1986-07-15   insert into member values('노홍철', 30, '1989-09-01');  insert into member values('박나래', 27, '1986-07-15'); |  |
| member 테이블에 입력된 모든 레코드를 출력하시오  select \* from member |  |
| member 테이블의 데이터 중 이름이 박나래 인 사람의 레코드만 출력하시오  select \* from member where name = '박나래' |  |
| member 테이블의 데이터 중 생년월일이 1989-09-01 인 사람의 이름과 생년월일을 출력하시오  select name, birth from member where birth = '1989-09-01'; |  |
| member 테이블에서 나이가 27세인 사람의 이름, 나이를 출력하시오  select name, age from member where age = 27; |  |
| member 테이블에서 이름이 노홀철인 사람의 이름을 홍철로 수정하시오  update member set name = '홍철' where name = '노홍철'; |  |
| member 테이블에서 이름이 박나래인 사람의 레코드를 삭제하시오  delete from member where name = '박나래' |  |
| 컬럼제거 하는법  alter table member  drop column age;  member 테이블을 제거하시오 (DDL)  drop table member; |  |
| MySQL실습 | | |
|  | cmd 창을 열은 후, mysql이 설치된 디렉토리의 bin으로 이동하여 mysql 접속명령어를 이용하여 mysql root 계정으로 로그인 하시오  cd C:\Program Files\MariaDB 10.5\bin  mysql -u root -p 1234 |  |
|  | 아래의 조건으로 데이터베이스를 생성하시오   1. 데이터베이스명 : mydb   create database mydb; |  |
|  | 아래의 조건으로 유저를 생성하시오   1. 유저명 : dev0907 2. 비밀번호 : 1234   create user 'dev0907'@'localhost' identified by '1234; |  |
|  | dev0907계정에게 mydb에 대한 모든사용 권한을 부여하고 , 갱신하시오  grant all privileges on mydb.\* to 'dev0907'@'localhost'; |  |
|  | dev0907계정으로 로그인 하시오  mysql -u dev0907 -p |  |
|  | dev0907계정에서 아래의 조건으로 테이블을 생성하시오   1. 테이블명 : member 2. 컬럼 구성  * 이름 : name , 영문 20자 넣을 크기로 생성 * 나이: age , 정수형으로 생성 * 출생일 : birth , 날짜형으로 생성   < 완성된 테이블 개념도 >   |  |  |  | | --- | --- | --- | | name | age | birth | |  |  |  | |  |  |  |   create table member(  -> name varchar(20),  -> age int,  -> birth date); |  |
|  | member 테이블에 두사람의 정보를 아래의 값으로 넣으시오   1. 이름: 노홍철, 나이:30 , 생년월일 1989-09-01 2. 이름: 박나래 , 나이:27 , 생년월일 1986-07-15   insert into member(name,age,birth) values('노홍철',30,'1989-09-01');  insert into member(name,age,birth) values('박나래',27,'1986-07-15'); |  |
|  | member 테이블에 입력된 모든 레코드를 출력하시오  select \* from member; |  |
|  | member 테이블의 데이터 중 이름이 박나래 인 사람의 레코드만 출력하시오  select \* from member where name='박나래'; |  |
|  | member 테이블의 데이터 중 생년월일이 1989-09-01 인 사람의 이름과 생년월일을 출력하시오  select name, birth from member  where birth='1989-09-01'; |  |
|  | member 테이블에서 나이가 27세인 사람의 이름, 나이를 출력하시오  select name, age from member where age=27; |  |
|  | member 테이블에서 이름이 노홀철인 사람의 이름을 홍철로 수정하시오  update member set name='홍철' where name='노홍철'; |  |
|  | member 테이블에서 이름이 박나래인 사람의 레코드를 삭제하시오  delete from member where name='박나래'; |  |
|  | member 테이블을 제거하시오  drop table member; |  |
| SQL문 기초 | | |
|  | 급여가 2000 이하인 사원의 이름, 급여를 출력하시오  select ename, sal from emp  where sal <= 2000; |  |
|  | 이름이 S로 시작하는 사원의 이름, 급여, 관리자의 사원번호를 출력하시오  select ename, sal, mgr from emp  where ename like 'S%';  이름 중간에 A자가 들어가는 사원의 사원번호, 이름을 출력하시오  select empno, ename from emp where ename like '%A%';  이름의 끝이 T로 끝나는 사원의 이름, 업무, 부서번호를 출력하시오  select ename, job, deptno from emp where ename like '%T'; |  |
|  | 커미션이 0인 사원의 이름 , 급여, 커미션을 출력하시오  select ename, sal, comm from emp where comm = 0; |  |
|  | 커미션 계약을 하지 않은 사원의 이름, 급여 ,커미션을 출력하시오  select ename, sal, comm from emp where comm is null; |  |
|  | 입사일이 가장 빠른 사원의 이름, 입사일을 출력하시오  select min(hiredate) from emp; |  |
|  | 사원의 총 수, 평균급여, 급여의 합, 최대급여, 최소급여를 출력하시오  select count(\*) as 사원총수 from emp;  select avg(sal) as 평균급여 from emp;  select max(sal) as 최대급여 from emp;  select min(sal) as 최소급여 from emp;  select count(\*) as 사원총수, round(avg(sal),1) as 평균급여, max(sal) as 최대급여, min(sal) as 최소급여 from emp; |  |
| 서브쿼리 / 집계함수 / Group by | | |
|  | .ALLEN과 부서가 같은 사원들의 사원명, 입사일을 출력하되 높은 급여순으로 출력하세요.  select ename, hiredate from emp  where deptno = (select deptno from emp where ename = 'ALLEN')  order by sal desc; |  |
|  | 최대급여를 받는 사원보다 입사일이 늦은 사원의 이름, 입사일을 출력하세요.  select ename, hiredate from emp  where hiredate > (select hiredate from emp  where sal = (select max(sal) from emp)); |  |
|  | 이름에 "T"자가 들어가는 사원들의 급여의 합을 구하세요.  select sum(sal) from emp where ename like '%T%'; |  |
|  | 20번 부서의 최고 급여보다 급여가 높은 사원의 사원번호,사원명,급여를 출력하세요.  select empno, ename, sal from emp where sal > (select max(sal) from emp where deptno = '20'); |  |
|  | SCOTT의 급여에서 1000 을 뺀 급여보다 적게 받는 사원의 이름,급여를 출력하세요.  select ename, sal from emp where sal < (select sal from emp where ename = 'SCOTT') - 1000; |  |
|  | 이름이 S로 시작하고 마지막글자가 H인 사원의 이름을 출력하세요.  select ename from emp where ename like 'S%H'; |  |
|  | WARD가 소속된 부서의 사원들의 평균 급여보다 급여가 높은 사원의 이름 ,급여를 출력하세요.  select ename, sal from emp where sal >  (select avg(sal) from emp where deptno = (select deptno from emp where ename = 'WARD')); |  |
|  | 아래와 같이 , 사원테이블을 대상으로 JOB의 종류를 출력하는 쿼리문을 작성하시오 ( 중복없이 )  select job from emp group by job;   |  | | --- | | JOB | | SALESMAN | | CLERK | | PRESIDENT | | MANAGER | | ANALYST | |  |
|  | JOB과 그 JOB에 속한 사원수를 출력하되, 아래와 같이 컬럼의 제목이 한글로 표현되도록 출력하세요  select job as 담당업무, count(job) as 사원수 from emp group by job;   |  |  | | --- | --- | | 담당업무 | 사원수 | | SALESMAN | 5 | | CLERK | 3 | | PRESIDENT | 1 | | MANAGER | 3 | | ANALYST | 2 | |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |