

- 리뷰

SQL 종류

DQL	SELECT	조회
DDL	CREATE TABLE ALTER TABLE DROP TABLE	테이블 정의 수정 삭제
DML	INSERT UPDATE DELETE	테이블 내의 데이터 = 레코드(여러 컬럼) 저장 수정 삭제
DCL	GRANT REVOKE	DB 사용 권한 부여 회수 (단 DBA 만 사용 가능-system) hr 계정 명령어 사용불가
TCL	COMMIT ROLLBACK	트랜잭션 처리

테이블	레코드 형식 지정 저장 그릇
레코드	여러개 컬럼 구성 데이터 1개
권한	테이블이나 레코드 저장 조회 자격 부여 회수
트랜잭션	여러개 sql 연속 결합 논리적 1개 실행 결과 = 여러개 sql 실행 결합 all -> commit 일부 실행만, --> nothing -> rollback

```

create table 테이블명(
컬럼명1 타입(길이) constraint 제약조건명 타입,
.....
);

alter table 테이블명 add 추가컬럼명 타입(길이) constraint 제약조건명 타입;
alter table 테이블명 modify 수정컬럼명 타입(길이) constraint 제약조건명 타입;
alter table 테이블명 drop column 삭제컬럼명;
alter table 테이블명 rename column 이전컬럼명 to 새로운컬럼명;

drop table 테이블명; -> 컬럼정의 + 레코드 삭제 -> 영구 삭제

insert into 테이블명(컬럼명....) values(값,...);
update 테이블명 set 변경컬럼명= 값 WHERE 수정조건식;
delete 테이블명 where 삭제조건식;
insert into 테이블명(컬럼명....) select.....; --> 다른 테이블의 데이터를 복사(테이블 정의된 후)
update 테이블명 set 변경컬럼명= (select) WHERE (select);
delete 테이블명 where (select...)

```

create table 테이블명 as select... ; --> 다른 테이블의 컬럼명 타입 길이 복사, 다른 테이블의 데이터 복사 create table copy_emp as select * from emp; 6 개컬럼 복사 + 레코드 복사
grant 권한명 to 계정; revoke 권한명 from 계정; grant resource, connect to test; grant select on emp to test; revoke select on emp from test
commit--> 이전 sql 완벽 db 영구 반영 rollback --> 이전 sql 취소 insert / delete / update dml 실행 ----> 임시 공간 저장 ----> commit --> db 저장 dml 실행 ----> 임시 공간 저장 --> rollback

제약조건 종류 - 5개

not null	컬럼값 비어있으면 안된다 조건 지정
unique	컬럼값 중복되면 안된다 조건 지정 null 1개 허용
primary key	not null + unique 조건 지정(레코드 식별자)
check	사용자 정의 조건 지정
foreign key	다른 테이블의 컬럼(부모 컬럼)에 포함 값들만 사용 조건 지정

==> 효력 발생 - dml

1,2,6,7,8장 - 3일 - sql

회원가입정보 영구 저장 -> db

자바프로그램 ---->파일 ----> abc 1111 tv 2 1000000 card ----> ???프로그램

예 : 프로그램 독립적 데이터 저장 / 정해진 형식 취급 / 영구 저장

DB 테이블 정의 컬럼 타입(길이) 제약조건

자바프로그램 ---->DB ---->테이블, 아이디 VARCHAR2(10) abc 1111 tv 2 1000000 *card* ----> ???
프로그램

예 : 데이터 저장 순서 / 암호 숫자로만 취급 /

1> 자바프로그램에서 DB 저장 - 조회 기능 = JDBC

JAVA DATABASE CONNECTIVITY

java.sql패키지의 ---인터페이스

interface Connection { --> jdk 전문가 . db 종류마다 구현 필요. 각 db 회사 반드시.

void connect();

}

class OracleConnection implements Connection{

public void connect(){ 오라클 연결 라이브러리 호출 구현}

}

class MySqlConnection implements Connection{

public void connect(){ mysql 연결 라이브러리 호출 구현}

}

Connection c1 = new OracleConnection();

Connection c2 = new MySqlConnection();

java.lang

java.util

java.io

jdk / oracle db / mysql db --> oracle사 제공

0. db jdbc driver 호출

1. db 연결

(db종류, ip, port, 계정, 암호.)

2. sql 작성 - 전송

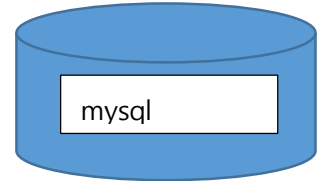
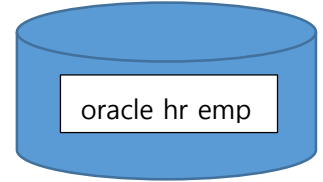
3. sql 실행 결과 이용

4. db 연결 해제

oracle jdbc driver

java.sql 패키지의 인터페이스
를 상속받아 오라클 라이브러
리 구현 제공 클래스 모임.

db 제공회사 배포



mysql jdbc driver = 클래스

-자바 코드를 mysql 맞게끔

자동 변경 라이브러리 호출

1> jdk / eclipse

2> java.sql.*

3> oracle db 설치

4> oracle jdbc driver 설치

(www.oracle.com 사이트 -> download --> jdk, oracle db, mysql db , jdbc driver

->windows, 11g xe)

10시 15분

0. db jdbc driver 호출

1. db 연결

(db종류, ip, port, 계정, 암호.)

2. sql 전송

3. sql 결과 검색

4. db 연결 해제

```
Connection con = DriverManager.getConnection()
```

java.sql.Connection 생성 - db 연결

java.sql.Connection close() - db 연결해제

java.sql.Statement

jdbc 전송 sql

ddl - 가능. 권장하지 않는다.(ddl 직접 db 내부 실행 1번)

dcl - system 계정 가능. 권장하지 않는다.

dml - jdbc sql

insert / update / delete --> db 실행결과 리턴 (레코드수)

```
Statement st = con.createStatement();
```

```
int rows = st.executeUpdate("insert | update | delete ");
```

==> dml 실행 변화행갯수

select - jdbc sql

(조회 - db 변화 x)

```
Statement st = con.createStatement();
```

```
ResultSet rs =st.executeQuery("select");
```

```
select id, name, salary from emp;
```

```
rs
```

100	이사원	78000.99
200	김대리	xxx
.....		

```
while(rs.next()){
```

```
    int id = rs.getInt(1); --> rs.getInt("id") --> 100
```

```
    or
```

```
    String id = rs.getString(1) --> "100"
```

```
    String name = rs.getString("name");
```

```
    double salary = rs.getDouble("salary")
```

```
}
```

rs.next(); ==> rs 의 첫번째 레코드 이동. true 리턴

rs.next(); ==> rs 의 두번째 레코드 이동. true 리턴

rs.next(); ==> rs 의 세번째 레코드 이동. true 리턴

rs.next(); ==> rs 의 네번째 레코드 이동 /없다 . false 리턴

```
java.sql.Connection
```

```
java.sql.Statement st = con.createStatement();
```

```
java.sql.ResultSet
```

자바	ResultSet	오라클
int	getInt("컬럼명") getString()	number(n)
double	getDouble("컬럼명") getString()	number(n, s)
String	getString("컬럼명")	varchar2(n)
java.sql.Date	getDate("컬럼명") getString("indate")	date 'rr/mm/dd' 조회 select to_char

		(날짜데이터, 'mm-dd-yyyy hh:mi:ss') as indate
--	--	---

(ddl – 실행 db 변화. 행의 수 변화x/o)

create table a (b number(5));

create table a as select * from emp;

class
EmpVO

```
class EmpDAO
ArrayList getEmp(){
    driver 로딩
    db 연결-시간, 메모리 부하 작
    업
    sql 전송
    emp 테이블의 모든 레코드
    조회 sql 결과 검색
    rs--> ArrayList list 복사
    db 해제
    return list;
```

```
class EmpMain
void test(){
    EmpDAO dao = new EmpDAO();
    ArrayList list = dao.getEmp();
```

emp 테이블 – ArrayList<EmpVO> 나 배열

레코드

컬럼

class EmpVO{ --> 레코드 1개 표현 객체

int id;--> 컬럼변수

String name;...

class명

xxxVO = VALUE OBJECT = 값 저장 / 추출 용도 객체 =

emp테이블 => EmpVO

EmpDO = DATA OBJECT

EmpDTO = DATA TRANSFER OBJECT

VO = DO = DTO

xxxDAO = DATA ACCESS OBJECT = 값 접근 용도 객체

자바 <----JDBC <----- DB 값

자바 <----파일 입력(java.io.FileReader..) <----- 파일 값

java.sql.Statement 객체

INSERT / UPDATE / DELETE / SELECT

int <-- executeUpdate()

ResultSet <-- executeQuery()

XXXVO / XXXDAO / MAIN...

-> tcl - jdbc sql

-> java.sql.PreparedStatement

-> 이클립스 data explorer -> sql 작성 - 실행 -결과 조회

-> java + db + jdbc -> 시험 + 과제

시험 - 5시부터 - 6시(퇴실qr) - 줌 퇴실 - 나머지 시간 180분 응시

과제 - 오후 시간 2시이후