

JSP

-DataBase

1.DataBase
2.SQL
3.SQL문법

* 데이터 베이스(DataBase)

- 수 많은 데이터들을 관리하기 위한 수단으로 데이터베이스가 사용됩니다.
- 데이터베이스는 데이터의 **추가, 삭제, 검색, 수정, 이동** 등의 기능이 있습니다.
- 그리고 이 데이터베이스를 관리하는 도구가 DBMS(DataBase Management System)입니다.
- DBMS는 SQL언어와 데이터베이스를 연결해주는 도구입니다.
- DBMS의 중 가장 많이 사용되는 DBMS는 **RDBMS(관계형 데이터베이스: Relational DBMS)**입니다.
- 우리가 흔히 사용하는 오라클, **MySQL**, MS SQL 등이 대표적인 관계형 데이터베이스입니다.



(DataBase)

* SQL이란?

- Structured Query Language의 약자로서 데이터베이스로부터 데이터를 조회하고 삭제하는 등의 작업을 수행할 때 사용하는 언어입니다.

* 테이블과 레코드

- RDBMS에서 데이터를 저장하는 장소를 **테이블**이라고 합니다.
- 테이블을 데이터들을 저장하며 데이터들의 정보를 갖고 있습니다.
- 테이블의 구조와 관련된 정보를 테이블 **스키마**(Schema)라고 부릅니다.
- 테이블의 구조는 각각의 정보를 저장하는 **컬럼**(column)과 컬럼 타입 그리고 컬럼의 길이로 구성됩니다.
- 각 컬럼에 저장된 데이터 값을 **레코드**(record)라고 부릅니다.
- 레코드, 컬럼 그리고 테이블을 사용해서 데이터를 저장하고 조회하는 등의 작업을 수행하는 것을 데이터베이스 프로그래밍이라고 부릅니다.

user_id	user_pw	user_name	user_email	user_address
good	1234	홍길자	google	서울
hong	1234	홍길자	google	서울
kim	1234	홍길자	google	서울
ooh	1234	홍길자	google	서울
park	1234	홍길자	google	서울

* SQL 문법

- SQL문의 특징은 대/소문자를 구분하지 않습니다.

DDL(Data Definition Language)

- **CREATE** : 테이블이나 인덱스, 뷰 등 데이터베이스 객체를 생성
- **DROP** : 생성된 데이터베이스 객체를 삭제
- **ALTER** : 이미 생성된 데이터베이스 객체를 수정

DML(Data Manipulation Language)

- **SELECT** : 테이블이나 뷰에 있는 데이터 조회
- **INSERT** : 데이터를 신규로 생성
- **UPDATE** : 이미 생성된 데이터를 수정
- **DELETE** : 데이터를 삭제
- **COMMIT** : 트랜잭션 처리, 변경된 데이터를 최종 적용
- **ROLLBACK** : 트랜잭션 처리, 변경된 데이터를 적용하지 않고 이전으로 되돌림

DCL(Data Control Language)

- **GRANT** : 사용자에게 특정 권한을 부여
- **REVOKE** : 사용자에게 부여된 권한을 회수

1. 데이터베이스 생성

- create database [db이름] default character set utf8;

a. 사용자 계정 생성

- create user '[계정명]' identified by '[암호]';

b. 권한 부여

- grant [권한] on [데이터베이스이름].[테이블명] to '[계정]';

- 데이터베이스이름.* 는 모든 테이블 접근을 허용한다.

c. 권한 삭제

- revoke all on [데이터베이스이름].[테이블명] from [user이름];

2. 테이블 생성 문법

- create table [테이블 이름] (
 [컬럼명1] [컬럼데이터 타입(byte)],
 [컬럼명2] [컬럼데이터 타입(byte)],

);



- 테이블 생성시 필요한 SQL 데이터 타입

A. MYSQL 문법

- 문자형 컬럼일 경우 CHAR, VARCHAR
- 정수형 컬럼일 경우 INT
- 날짜형 컬럼일 경우 DATETIME

```
# 데이터베이스 생성 명령
# 워크벤치에서 sql문 실행명령: 단축키 컨트롤 + 엔터
create database test default character set utf8;
```

```
# DB 사용자 계정 생성
create user 'jsp' identified by 'jsp';
```

```
# 사용자에게 권한 부여
grant all privileges on test.* to 'jsp';
```

```
# 테이블 생성 명령
# primary key - 컬럼의 주요키를 뜻하며 중복된 데이터를 방지합니다.
# not null - 데이터에 null값이 들어오는 것을 방지합니다.
```

```
create table member (
  id varchar(20) primary key not null,
  pw varchar(20) not null,
  name varchar(30) not null,
  email varchar(80)
);
```

4. INSERT 문

- insert문은 데이터를 저장하는 용도의 쿼리문입니다.
- `insert into` [테이블명] ([컬럼1], [컬럼2], ...) `values` ([값1], [값2], ...);
- 테이블명 뒤에 컬럼명을 명시하지 않으면 전체 컬럼에 대해 값을 지정해야 합니다.

5. UPDATE 문

- update 쿼리는 데이터를 수정하는 용도입니다.
- `update` [테이블명] `set` [수정할 컬럼]=[수정할 값] `where` [조건];

6. DELETE 문

- delete 쿼리는 데이터를 삭제하는 용도입니다.
- `delete from` [테이블명] `where` [조건];

데이터 추가

```
insert into member (id, pw, name) values ('abc123', '1234', '홍길동');
insert into member values ('def456', '4433', '이순신', 'aaa@bbb.com');
insert into member values ('kkk456', '3456', '김건모', 'aaa@naver.com');
insert into member values ('ghi789', '6753', '장보고', 'bbb@naver.com');
insert into member values ('aaa123', '2432', '박찬호', 'ccc@naver.com');
insert into member values ('bbb456', '7653', '손흥민', 'ddd@naver.com');
insert into member values ('ccc789', '2398', '기성용', 'eee@naver.com');
```

데이터 수정

```
update member set name='신사임당', email='shin012@daum.net' where id='abc123';
```

데이터 삭제

```
delete from member where id='def456';
```

7. SELECT 문

- select 쿼리는 데이터베이스로부터 정보를 조회하는 문장입니다.
- select [컬럼1], [컬럼2], ... from [테이블명] where [조건];
- 모든 컬럼을 조회하고 싶으면 조회할 컬럼 이름에 * 를 써줍니다.
- where절로 조건을 지정할 때 like를 사용하면 특정 문장을 포함하고 있는지 확인할 수 있습니다.

8. SELECT 문의 정렬

- 게시판이나 회원목록 등을 출력할 때 이름 순서나 아이디 오름차순 혹은 글 번호순으로 정렬하는 것이 일반적입니다.
- sql 쿼리문에서는 where 조건절 뒤에 order by절을 사용하여 데이터를 정렬합니다.
- 오름차순이면 asc, 내림차순이면 desc를 사용합니다.

데이터 조회

```
select id from member;
select name, email from member;

select email from member where id='abc123';
select pw from member where id='kkk456';
select * from member where id='bbb456' and pw='1234';

select * from member where name like '홍%';
select * from member where name like '%동';
select * from member where name like '%길%';
select * from member where email like '%naver%';
```

검색 데이터 정렬

```
select * from member order by name asc;
select * from member order by id desc;
select * from member order by name asc, id desc;

select * from member where name like 'ㅇ|%' order by id asc;
```

9. DROP 문

- 테이블을 삭제할 수 있는 쿼리문입니다.
- `drop table` [테이블 이름];

10. commit

- 서버에 입력한 쿼리문들의 상태 지점을 저장하는 용도.
- `commit`과 `rollback`은 **DML**문에서만 작동합니다.

11. rollback

- commit된 위치를 기준으로 commit 이후에 작성 쿼리문을 commit한 시점으로 되돌리는 문장.

```
# 테이블 객체 삭제
drop table member;
```

```
#오토 커밋 확인
SELECT @@AUTOCOMMIT;
SET AUTOCOMMIT = TRUE;
SET AUTOCOMMIT = FALSE;
```

```
# 트랜잭션 처리(commit, rollback)
insert into member values ('가가가', '1234', '가가가', '가가가');
commit;
```

```
delete from member where id='abc123';
rollback;
```


