## **SQL**

SQL은 Structured Query Language (구조적 질의 언어)의 줄임말로, 관계형 데이터베이스 시스템(RDBMS)에서 자료를 관리 및 처리하기 위해 설계된 언어이다.

| SQL은 대소문자를 가리지 않는다.  단, 서버 환경이나 DBMS 종류에 따라 데이터베이스 or 필드명에 대해 대소문자를 구분하기도 한다.  SQL 명령은 반드시 **세미콜론(;)**으로 끝나야 한다.  고유의 값은 **따옴표('')**로 감싸준다.  ex) SELECT \* FROM EMP WHERE NAME = 'James';  SQL에서 **객체를 나타낼 때는 백틱(``)**으로 감싸준다.  ex) SELECT `COST`, `TYPE` FROM `INVOICE`;  주석은 일종의 도움말로, 주석 처리된 문장은 프로그램에서 동작하지 않는다.  **한 줄 주석은 문장 앞에 --** 를 붙여서 사용한다.  ex) -- SELECT \* FROM EMP; 이 쿼리는 실행되지 않습니다.  여러 줄 **주석은 /\* \*/** 로 감싸줍니다.  ex)  /\*  SELECT \* FROM EMP WHERE EMPID=(SELECT \* FROM EMP WHERE NAME='홍길동')  \*/ |
| --- |

### **- [ 데이터베이스 ]**

데이터베이스(DB: database)는 통합하여 관리되는 데이터의 집합체를 의미한다. 이는 중복된 데이터를 없애고, 자료를 구조화하여, 효율적인 처리를 할 수 있도록 관리된다. 따라서, 여러 업무에 여러 사용자가 데이터 베이스를 사용할 수 있다.

이러한 데이터베이스는 응용 프로그램과는 다른 별도의 미들웨어에 의해 관리된다. 데이터베이스를 관리하는 이러한 미들웨어를 데이터베이스 관리 시스템(DBMS: Database Management System)라고 한다.

| 1. 사용자의 질의에 대하여 **즉각적인 처리와 응답**이 이루어진다.  2. **생성, 수정, 삭제**를 통하여 항상 최신의 데이터를 유지한다.  3. 사용자들이 원하는 **데이터를 동시에 공유**할 수 있습니다.  4. 사용자가 원하는 데이터를 **주소가 아닌 내용에 따라 참조** 할 수 있습니다.  5. 응용프로그램과 데이터베이스는 독립되어 있으므로, 데이터의 논리적 구조와 응용 프로그램은 별개로 동작된다. |
| --- |

### **- [ DDL ]**

Data Definition Language. 데이터베이스나 테이블 등을 생성, 삭제하거나 그 구조를 변경하기 위한 명령어

#### **CREAT**

| **특수코드값** | **실제표현** | |
| --- | --- | --- |
| CREATE DATABASE | 새로운 데이터베이스를 생성해준다.  CREATE DATABASE 데이터베이스이름 ; | |
| USE | 데이터베이스를 선택.  USE 데이터베이스이름 ; | |
| CREATE TABLE | 데이터베이스는 하나 이상의 테이블로 구성되며, 이러한 테이블에 데이터를 저장하여 관리할 수 있다. CREATE TABLE 문은 새로운 테이블을 생성한다.  CREATE TABLE 데이터베이스이름  (  필드이름1 필드타입1,  필드이름2 필드타입2,  ...  ) | |

제약 조건(constraint)이란 데이터의 무결성을 지키기 위해 데이터를 입력받을 때 실행되는 검사 규칙을 의미한다. 이러한 제약 조건은 CREATE 문으로 테이블을 생성할 때나, ALTER 문으로 필드를 추가할 때도 설정할 수도 있다.

CREATE TABLE 문에서 사용할 수 있는 제약 조건은 다음과 같습니다.

| **특수코드값** | **실제표현** | |
| --- | --- | --- |
| NOT NULL | 해당 필드는 NULL 값을 저장할 수 없게 됩니다. | |
| UNIQUE | 해당 필드는 서로 다른 값을 가져야만 합니다. | |
| PRIMARY KEY | 해당 필드가 NOT NULL과 UNIQUE 제약 조건의 특징을 모두 가지게 됩니다. | |
| FOREIGN KEY | 하나의 테이블을 다른 테이블에 의존하게 만듭니다. | |
| DEFAULT | 해당 필드의 기본값을 설정합니다. | |

또한, AUTO\_INCREMENT 키워드를 사용하면 해당 필드의 값을 1부터 시작하여 새로운 레코드가 추가될 때마다 1씩 증가된 값을 저장한다. 이때 AUTO\_INCREMENT 키워드 다음에 대입 연산자(=)를 사용하여 시작값을 변경할 수 있다.

#### **ALTER**

| **특수코드값** | **실제표현** | |
| --- | --- | --- |
| ALTER DATABASE | ALTER DATABASE 문은 데이터베이스의 전체적인 특성을 수정할 수 있게 해준다. 이러한 데이터베이스의 특성은 데이터베이스 디렉터리의 db.opt 파일에 저장되어 있습니다. | |
| ALTER TABLE |  | |

CREATE, ALTER, DROP

### **- [ DML ]**

Data Manipulation Language. 데이터베이스에 저장된 데이터를 처리하거나 조회, 검색하기 위한 명령어

INSERT, UPDATE, DELETE, SELECT 등

| **특수코드값** | **실제표현** | **비고** |
| --- | --- | --- |
| 선언문 | <%! … %> | 자바 변수나 메소드를 정의  전역변수&전역 메소드로 사용된다 |
| 스크립틀릿 | <% … %> | 자바 로직 코드를 작성하는데 사용  JSP 페이지가 서블릿 프로그램에서 서블릿 클래스로 변환할 때 JSP 컨테이너가 자바 코드가 삽입되어 있는 스크립트 태그를 처리하고 나머지는 HTML 코드나 일반 텍스트로 간주 |
| 표현문 | <%= … %> | 변수, 계산식, 메소드 호출 결과를 문자열 형태로 출력 |
| 주석 | <%-- --%> | Ctrl + Shift + / |
| 지시자 | <%@ %> | 페이지의 속성  <%@ page import="java.util.Arragys"%>  <%@ include file="include01.jsp" %> |

### **- [ DCL ]**

Data Control Language. 데이터베이스에 저장된 데이터를 관리하기 위하여 데이터의 보안성 및 무결성 등을 제어하기 위한 명령어

GRANT, REVOKE 등

#### **exception**

exception은 page지시어에서 오류 페이지로 지정된 JSP 페이지에서 예외가 발생할 때 전달되는 java.lang.Throwable의 인스턴스에 대한 참조 변수다. 현재 페이지를 처리하다 발생하는 예외상황에 대한 정보를 가져올 수 있다.

| **실제표현** | **비고** |
| --- | --- |
| getMessage() | 문자열로 된 오류 메시지를 반환한다. |
| printStackTrace() | 스택 추적 정보를 출력한다. |
| toString() | 예외 클래스 이름과 함께 오류 메시지를 반환한다. |

----------------------------------------------

[workspace] > [.metadata] > [org.eclipse.wst.server.core] > [tmp0] > [wtpwebapps] > [ServletStudy-20220823-youri] > [WEB-INF] > [classes]

IP: 집주소 느낌. www 인터넷

= IPv4 255.255.255.255

= IPv6

포트번호: 서버 종류 구분하는것? 8000번 8080번 등

web서버 포트는 기본적으로 8080, 80, 8000 등 사용

데이터베이스 : 3306

포트포어딩? KT나 SKT에서 주는 아이디를 외부 IP라고 부른다.

> 공유기가 나옴(여러대 컴퓨터가 쓰기 위함) 공유기는 본인들끼리만 받는다

> 외부에서 8000 8080 서버를 배포할때…. 들어가게 해줌.

MVC > M: / V:JSP /C:Servlet

#### 