jinyPHP 2권 - 03. SQL문법

**03. SQL**

====================

요즘 사용하는 대부분의 데이터베이스 시스템은 관계형 모델을 바탕으로 설계가 되어 있습니다. 관계형 데이터 베이스 시스템을 사용하기 위해서는 이를 명령하고 처리할 수 있는 별도의 SQL명령들을 학습해야 합니다.

즉, SQL 명령문을 사용한다는 것은 해당 데이터베이스가 관계형 모델이라는 것을 말합니다. 개발자들이 자신의 주력 언어 이외에도 또다른 새로운 언어를 학습을 한다는 것은 부담스러운 일입니다. 또한, 새로운 언어들을 학습한다는 것은 기술을 적용하는데 진입장벽이 될 수도 있습니다. 하지만, 데이터베이스 시스템은 현대사회의 IT기술에서 매우 유용하고 핵심적인 부분이라 꼭 같이 학습을 하고 다양한 아이디어를 융합하는 것이 중요합니다.

**03.1 SQL 시작**

====================

관계형 데이터베이스를 주로 사용되는 명령방식은 SQL 명령입니다. SQL(Structured Query Language) 언어는 처음 IBM에서 시작된 언어 입니다. 관계형 데이터베이스의 아버지인 Edgar F.Codd 박사 또한 IBM 연구원 출신입니다.

SQL언어는 간단한 명령어의 집합 형태로 구성되어 있습니다. SQL언어는 관계형 데이터베이스 시스템에 테이블생성, 자료 삽입, 수정, 조회 등의 다양한 작업을 명령을 지시할 수 있습니다.

대부분의 데이터베이스는 관계형으로 공통적으로 SQL언어들을 지원합니다. 따라서 이제 데이터베이스에서 SQL 언어는 표준언어로 자리를 잡아 가고 있습니다. 또한 SQL언어는 ANSI (American National Standards Institute) 표준을 따라 한번의 학습으로 다양한 데이터베이스 시스템에 적용하여 사용할 수 있습니다. 하지만, 다른 몇개의 데이터 베이스는 약간의 다른 문법을 사용하는 경우도 있습니다.

오라클은 PL/SQL, SQL Server는 T-SQL 이라고 하기도 합니다. MYSQL은 표준 SQL을 따릅니다. SQL언어 한 개의 학습으로 다양한 데이터베이스를 사용할 수 있다는 것은 개발환경의 개선과 다양한 시스템을 적용하는데 있어서 기술장벽을 낮추는 효과가 발생합니다.

**03.2 SQL 연동**

====================

데이터베이스는 데이터처리를 위해서 콘솔화면을 통하여 SQL명령을 실행할 수 있습니다. 데이터베이스는 이러한 조작에 필요한 클라이언트 단말 응용프로그램을 제공을 하고 있지만 별도의 외부 프로그램 같이 처리를 하기 위해서는 연동작업이 필요합니다.

데이터베이스는 콘솔 터미널 작업을 위해서 별도의 GUI 환경을 제공하는 경우도 있습니다. 이러한 환경은 사람과 데이터베이스 사이의 데이터를 기준으로 하는 인터페이스 입니다. 데이터베이스와 프로그램 간의 처리를 위한 인터페이스는 바로 SQL 명령문의 전달과 처리 입니다.

데이터베이스와 연동을 하여 프로그램을 작성하기 위해서는 반드시 SQL 언어를 사전에 학습을 해두어야 합니다. 대부분의 작업명령들을 SQL로 생성하여 프로그램에서 데이터베이스로 전송을 하기 때문입니다.

당연히 PHP에서도 데이터베이스를 연동을 하기 위해서는 SQL 명령문을 사용합니다. PHP는 데이터베이스에 접속을 하고, 작성한 SQL명령을 전송함으로써 DB 작업을 수행합니다. PHP 언어만 알고 SQL을 모른다면 데이터베이스를 바로 연동을 할 수 는 없습니다.

데이터베이스는 기본 별개의 서비스 프로그램입니다. SQL명령을 사용하지 않고 직접적으로 데이터베이스 관련 파일들에 접근하여 작업을 할 수는 없습니다. 기존의 언어와 데이터베이스를 연동하기 위해서는 데이터베이스 언어인 SQL문법을 알고 있어야 합니다.

SQL언어를 배우면 MYSQL 이외에 다양한 데이터베이스 시스템을 PHP에서도 사용을 할 수 있습니다. SQL언어를 배운다는 것은 다양한 데이터베이스 시스템을 활용할 수 있는 능력을 향상 시키는 것입니다.

**03.3 작성방법**

====================

SQL명령은 다른 프로그래밍 언어들 보다 복잡하고 어렵지 않습니다. 몇 개의 키워드와 작성 순서 요령만 터득하면 쉽게 사용을 할 수 있습니다.

**03.3.1 구성**

====================

SQL 명령은 크게 3가지 요소로 구성되어 있습니다.

* 명령을 지정하는 부분 (select .. from …)
* 검색조건 (where 절)
* 정렬방법 (order by 절)

단순한 구조로 되어 있습니다. 하지만 단순한 명령들을 잘 조합하면 보다 정교한 데이터베이스 작업을 할 수 있는 강력한 명령 언어 입니다.

**03.3.2 대소문자**

====================

SQL언어는 대소문자를 구분하지 않습니다. 대문자 또는 소문자로 사용을 하거나, 또는 낙타표기법 스타일로 섞어서 사용을 할 수도 있습니다.

* CREATE DATABASE
* Create Database
* create database

따라서 위와 같은 대소문자를 섞어서 사용한 SQL도 같은 문장으로 해석됩니다.

**03.3.3 최적화**

====================

데이터베이스 SQL 명령은 간결하게 명령을 작성합니다. 관계형 데이터베이스는 입력된 SQL 쿼리명령을 분석하고 데이터베이스를 처리합니다.

또한 관계형 데이터베이스는 다수의 테이블에 분할하고, 수많은 자료들을 디스크에 저장을 하기 때문에 하드웨어 적인 성능 부분을 많이 고려 하였습니다. 하드웨어의 성능과 처리 속도를 고려한 쿼리문장의 생성은 자원을 효율적으로 사용하는데 중요한 요소 입니다.

요즘들어 하드웨어 성능은 나날이 발전이 되고 있습니다. 하드웨어 성능으로 인한 속도 고려 부분들은 많이 개선이 되었지만, 이와 비례하여 서비스의 접속과 사용자들이 많아 지고 있습니다. 이 또한 효율적인 쿼리의 작성은 이전보다 더 중요합니다.

SQL 쿼리 명령은 관계형 데이터베이스를 활용하는데 매우 중요한 언어입니다. 또한, 작성에 많은 부분들을 고려하여 생각해야 되는 요소 이기도 합니다.

**03.4 SQL 구분**

====================

SQL 명령은 한 개단위 또는 여러 개의 단위를 묽어서 실행을 할 수 있습니다.

**03.4.1 세미콜론**

====================

PHP 에서는 각각의 명령을 구분하기 위해서 마지막에 구분기호 세미콜론(;)을 삽입합니다. SQL명령 또한 한 개의 쿼리 명령문을 구분하기 위해서 동일한 기호의 세미콜론(;)을 사용합니다.

**쿼리문법)**

|  |
| --- |
| SQL 명령어**;** |

|화면|

|  |
| --- |
| mysql> select \* from members  -> ; |

명령을 작성할때 세미콜론을 작성하지 않으면 오류를 발생하거나 추가로 입력(->)을 요구하기도 합니다. 만일 세미콜론이 없는 경우에는 오류가 발생할 수 있습니다.

경우에 따라서는 종결기호 세미콜론을 다른 기호로 변경을 할 수 있습니다.

**03.4.2 기호 변경**

====================

SQL에서 기본적인 종결기호는 세미콜론 입니다. 하지만 프로시저나 트리거등의 향상된 SQL 쿼리 명령을 실행해야 할 경우 세미콜론 기호는 중복되어 쿼리실행에 오류를 발생할 수 있습니다.

이런경우 임시로 종결기호를 다른 기호형태로 변경하여 입력을 처리할 수 있습니다. DELIMITER 키워드는 종결문자를 변경하는 SQL 명령문 입니다.

**쿼리문법)**

|  |
| --- |
| **DELIMITER 기호** |

만일 **DELIMITER //** 로 입력하면 종결문자는 // 로 변경이 됩니다. 다시 **DELIMITER ;** 로 하면 종결문자는 다시 세미콜론으로 복원이 됩니다.

**03.5 오류**

====================

관계형 데이터베이스는 입력되는 SQL 쿼리문장을 읽고 분석하여 데이터를 처리합니다. 따라서 정확한 동작을 위해서는 정확한 키워드와 문법을 맞추어 작성을 하여야 합니다.

만일 SQL 문법에 어긋나거나, 잘못된 키워드, 철자오류 등이 있을 경우 시스템은 에러 메시지를 출력합니다. 아래 화면은 콜솔에서 철자가 틀린 명령어를 입력한 예 입니다.

|화면|

|  |
| --- |
| mysql> create databas jiny;  ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'databas jiny' at line 1 |

이렇게 콘솔에서 오류 메시지를 출력하는 것은 사용자 입장에서 매우 편리합니다. 오류를 바로 확인하여 수정을 할 수 있기 때문입니다.

하지만, PHP와 같이 프로그램을 연동하여 SQL 명령문을 전달하는 경우 오류 메시지를 받아 처리를 할 수 없습니다. 따라서, SQL 명령문을 작성을 할 때 정확히 철자와 문법에 맞추어 작성을 하는 것이 중요합니다.

**03.6 데이터 반환**

====================

SQL 쿼리명령의 동작에는 크게 2가지의 종류가 있습니다. 출력결과 반환이 있는 명령문과 출력반환이 없는 명령문 입니다.

use, delimiter 와 같은 명령은 데이터베이스에 실행명령을 하지만 출력결과는 없는 명령들 입니다. 그냥 데이터베이스 시스템에 명령을 전달하는 역할만 수행을 합니다.

|화면|

|  |
| --- |
| mysql> use jiny;  Database changed |

데이터 출력은 없고 결과 메시지만 출력합니다.

이와 달리 show, select 와 같은 명령은 데이터를 반환하는 명령문 입니다. 데이터베이스는 입력된 명령을 분석하여 실행후에, 결과 값을 다시 화면에 출력을 합니다.

|화면|

|  |
| --- |
| mysql> select \* from members;  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  | Id | LastName | FirstName | Address | City | Country | manager | email |  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  | 1 | hojin | lee | shinchon | seoul | korea | infohojin | hojin@jinyphp.com |  | 2 | jiny | NULL | NULL | NULL | korea | infohojin | jiny@jinyphp.com |  | 3 | james | kim | youngdun-po | seoul | korea | info | james@jinyphp.com |  +----+----------+-----------+-------------+-------+---------+-----------+-------------------+  3 rows in set (0.00 sec) |

위의 화면은 회원 목록을 출력하는 select 명령입니다. 명령에 반응한 데이터를 출력합니다.

**03.7 주석처리**

============

다수의 SQL 명령을 작성하는 경우도 있을 것입니다. 명령들이 많아 별도의 .sql 파일로 저장을 할 수도 있습니다. 하지만 모든 명령들이 실행되는 것만 같이 작성을 해야 하는 것은 아닙니다.

SQL언어도 PHP와 같이 특정 SQL을 배제하거나 설명을 하기 위해서 주석처리를 기능을 제공합니다. 주석처리는 한개씩 명령을 수행하는 것과 달리 별도의 SQL 명령파일을 통하여 일괄 작업할때 사용하시면 편리합니다.

SQL 주석은 PHP와 같이 2가지 방법을 모두 지원합니다. 주석으로 처리된 내용은 MYSQL에서는 실행되지 않습니다.

**03.7.1 한줄주석**

============

한줄 주석은 라인의 시작 부분에 --를 입력하면 됩니다. 한줄 주석은 지정한 한줄만 주석으로 처리되어 실행되지 않습니다.

|  |
| --- |
| -- 이부분은 주석입니다.  --Select all:  --SELECT \* FROM members;  SELECT \* FROM members; |

주석은 SQL의 설명 또는 명령을 주석으로 처리할 수 있습니다.

|  |
| --- |
| SELECT \* FROM members -- WHERE City='Seoul'; |

또한 문장의 중간에 -- 가 삽입이 되면 시작위치부터 한 줄 끝까지 주석으로 처리되어 실행부분에서 제외가 됩니다.

**03.5.2 여러줄 주석**

============

여러줄의 주석은 /\*로 시작하여 \*/로 끝나는 부분으로 다중주석 처리를 할 수 있습니다. 지정한 블록 부분 전체를 주석으로 처리하는 방법입니다.

|  |
| --- |
| /\*Select all the columns  of all the records  in the members table:\*/  SELECT \* FROM members; |

또한 명령 중장 중간에도 /\* ~ \*/ 부분을 만나면 sql 실행에서 제외가 됩니다.

|  |
| --- |
| /\*SELECT \* FROM Customers;  SELECT \* FROM Products;  SELECT \* FROM Orders;  SELECT \* FROM Categories;\*/  SELECT \* FROM Suppliers;  SELECT userName, /\*City,\*/ Country FROM members; |

주석처리는 베이터베이스를 백업하거나 복구할 때 유용하게 사용을 할 수 있는 기능입니다.

보통 mysqldump와 같이 백업명령을 수행하여 작업을 할 때, 기본적인 주석 설명들이 같이 생성이 됩니다.