



## 저장 프로그램 & 프로시저

## 저장 프로그램 & 프로시저

### ☞ 저장 프로그램(Stored Program)

- 데이터베이스 응용 프로그램을 작성하는 데 사용되는 MySQL 전용 언어
- SQL에 변수, 제어, 입출력 등의 프로그래밍 기능을 추가
- 프로시저로 구현하여 객체 형태로 사용하며, 이는 MySQL DBMS에 저장
- 일반 프로그래밍 언어에서 사용하는 함수와 비슷한 개념
- 프로시저, 트리거, 사용자 정의 함수 모두 저장 프로그램

구분	프로시저	트리거	사용자 정의 함수
공통점	저장 프로시저		
정의 방법	CREATE PROCEDURE	CREATE TRIGGER	CREATE FUNCTION
호출 방법	CALL 문으로 직접 호출	INSERT, DELETE, UPDATE 문이 실행될 때 자동으로 실행	SELECT 문에 포함
기능 차이	SQL 문으로 할 수 없는 복잡한 로직 수행	기본값 제공, 데이터 제약 준수, SQL 뷰의 수정, 참조 무결성 작업 등을 수행	속성 값을 가공하여 반환, SQL 문에서 직접 사용

## 저장 프로그램 & 프로시저

### ☞ 데이터 삽입 프로시저 예시

```
delimiter //  
CREATE PROCEDURE InsertBook(  
    myBookID INTEGER, myBookName VARCHAR(40) , myPublisher VARCHAR(40), myPrice INTEGER)  
BEGIN  
    INSERT INTO book VALUES(myBookID, myBookName, myPublisher, myPrice);  
END  
// delimiter;
```

### ☞ 프로시저 실행

```
CALL 프로시저이름
```

### ☞ 프로시저 삭제

```
DROP PROCEDURE 프로시저이름
```

## 저장 프로그램 & 프로시저

☞ 동일한 도서가 있는지 확인 후 삽입하는 프로시저

```
delimiter //
CREATE PROCEDURE BookInertUpdate(
    myBookID INTEGER, myBookName VARCHAR(40) , myPublisher VARCHAR(40), myPrice INTEGER)
BEGIN
    DECLARE mycount INTEGER;
    SELECT count(*) INTO mycount FROM book WHERE bookname LIKE myBookName;
    IF mycount !=0 THEN
        UPDATE book SET price=myPrice WHERE bookname LIKE myBookName;
    ELSE
        INSERT INTO book VALUES(myBookID, myBookName, myPublisher, myPrice);
    END IF
END
// delimiter;
```

## 저장 프로그램 & 프로시저

☞ 동일한 도서가 있는지 확인 후 삽입하는 프로시저 확인

```
CALL BookInsertUpdate(22, '스포츠 즐거움', '좋은출판사', 25000);
```

```
SELECT * FROM book;
```

```
CALL BookInsertUpdate(22, '스포츠 즐거움', '좋은출판사', 20000);
```

```
SELECT * FROM book;
```

## 저장 프로그램 & 프로시저

### 👉 결과를 반환하는 프로시저

```
delimiter //  
CREATE PROCEDURE AveragePrice(OUT AverageVal INTEGER)  
BEGIN  
    SELECT AVG(price) INTO AverageVal FROM book WHERE price IS NOT NULL;  
END  
// delimiter;
```

### 👉 프로시저 실행

```
CALL AveragePrice(@myValue);  
SELECT @myValue;
```