

# Part 1.

---

저장 프로시저



# 01 저장 프로시저

## 저장 프로시저

어떤 동작을 일괄 처리할 때 사용된다  
SQL에서 제공하는 프로그래밍 기능이다  
모듈화 처리에 사용

## 형식

```
DELIMITER $$  
CREATE PROCEDURE 저장 프로시저명(IN 또는 OUT 파라미터)  
BEGIN  
    SQL 쿼리;  
END$$  
DELIMITER ;  
  
CALL 프로시저명;
```

## 02 입력 매개변수 지정

### 프로시저 생성 / 삭제

```
use sqlldb;
drop procedure if exists userProc;
delimiter $$
create procedure userProc()
begin
    select * from usertbl;
end$$
delimiter ;

call userProc();
show procedure status where db='sqlldb';

-- 삭제
use sqlldb;
drop procedure userProc;
```

## 02 입력 매개변수 지정

### 매개변수 입력

```
-- 프로시저 생성
use sqlDB;
drop procedure if exists userProc1;
delimiter $$
create procedure userproc1 (IN userName varchar(10))
begin
    select * from usertbl where name = userName;
end $$
delimiter ;

-- 프로시저 호출
call userProc1('조관우');
call userProc1('김경호');
```

## 02 입력 매개변수 지정

### 두개 매개변수 입력

```
drop procedure if exists userProc2;  
delimiter $$  
create procedure userProc2(in userBirth int , in userHeight int)  
begin  
    select * from usertbl  
    where birthYear > userbirth and height > userHeight;  
end $$  
delimiter ;  
  
call userProc2(1970,178);
```

## 02 입력 매개변수 지정

### 프로시저 상세 확인

```
select * from information_schema.routines;
```

## 02 중간 문제

### 문제

- 1 price가 100 이상인 모든 행을 출력하는 프로시저 buyProc1을 만들고 확인
- 2 제품명을 입력받아 입력한값과 일치하는 행을 출력하는 buyProc2를 만들고 확인

## 02 중간 문제

답

1 price가 100 이상인 모든 행을 출력하는 프로시저 buyProc1을 만들고확인

```
USE sqlDB;
```

```
drop procedure if exists buyproc1;
```

```
delimiter $$
```

```
create procedure buyProc1 (in n int)
```

```
begin
```

```
    SELECT * FROM buyTbl WHERE price>=n;
```

```
end $$
```

```
delimiter ;
```

```
call buyproc1(50);
```

```
call buyproc1(150);
```



## 02 중간 문제

### 답

2 제품명을 입력받아 입력한값과 일치하는 행을 출력하는 buyProc2를 만들고 확인

```
USE sqlDB;
drop procedure if exists buyproc2;
delimiter $$
create procedure buyProc2(in prod varchar(20))
begin
    select * from buytbl where prodName=prod;
end $$
delimiter ;

call buyproc2('운동화');
call buyproc2('노트북');
```

## 02 출력 매개변수 지정

### 출력 매개변수 지정

```
drop procedure if exists userProc3;
delimiter $$
create procedure userProc3(in txtVal char(10),out outVal int)
begin
    create table if not exists testtbl
    (
        id int auto_increment primary key,
        txt char(10)
    );
    insert into testTbl values(NULL,txtVal); -- input 값
    select MAX(id) into outVal from testTbl; -- output 값
end $$
delimiter ;

call userProc3('김상중' ,@myValue);
select * from testtbl;
select @myValue;
```

## 02 문제

!

price가 최고인 값을 OUTPUT하는 프로시저를 만들어보세요

```
CREATE PROCEDURE buyp3
    @outValue INT OUTPUT
AS
    SELECT @outValue=Max(price) FROM buytbl
GO

DECLARE @myValue INT;
EXEC buyp3 @myValue OUTPUT;
PRINT @myValue
```

## 02 문제

답

```
use sqldb;  
drop procedure buyproc;  
delimiter $$  
create procedure buyproc(out outval int)  
begin  
select max(price) into outval from buytbl;  
end $$  
delimiter ;
```

```
call buyproc(@maxprice);  
select @maxprice;
```

# Part 2.

---

오류 처리



# 01 오류처리

## 예제

```
drop procedure if exists errorProc;
delimiter $$
create procedure errorProc()
begin
declare i INT;
declare hap int;
declare saveHap int;
-- 오류 처리 (DECLARE 액션 HANDLER FOR 오류조건 처리할문장; )
declare exit handler for 1264
begin
    select concat('에러 직전의 합 : ', saveHap);
end;
-> 다음장 계속
```

# 01 오류처리

## 예제

```
-- set
set i=1;
set hap=0;
while(true) do
    set saveHap =hap;
    set hap = hap+i;
    set i = i + 1;
end while;
end $$
delimiter ;

call errorProc();
show procedure status;
```

# Part 3.

---

테이블 전달





## 01 테이블을 인자로 전달

### 오류 - 파라미터로 테이블 이름 전달 불가

```
use sqldb;
drop procedure if exists nameProc;
delimiter $$
create procedure nameProc(in tblName varchar(20))
begin
    select * from tblname;
end $$
delimiter ;

call nameProc('userTbl');
```

## 01 테이블을 인자로 전달

### 동적 SQL로 처리

```
use sqldb;
drop procedure if exists nameProc;
delimiter $$
create procedure nameProc(in tblName varchar(20))
begin
    set @sqlQuery = concat('select * from ', tblName);
    prepare myQuery from @sqlQuery;
    execute myQuery;
    deallocate prepare myQuery;
end $$
delimiter ;
call nameProc ('usertbl');
```

## 01 테이블을 인자로 전달

### 문제

전달되는 두 테이블을 inner Join 하는 프로시저를 만드세요

표시열 : 모든 열

프로시저명 : JoinProc

## 01 테이블을 인자로 전달

답

```
use sqlldb;
```

```
delimiter $$
```

```
create procedure JoinProc(in tbl1 char(20) , in tbl2 char(20))
```

```
begin
```

```
    set @sqlQuery = concat('select * from ',tbl1,' inner join ',tbl2);
```

```
    prepare myQuery from @sqlQuery;
```

```
    execute myQuery;
```

```
    deallocate prepare myQuery;
```

```
end $$
```

```
delimiter ;
```

```
call JoinProc('usertbl','buytbl');
```

# END.

---

고생하셨습니다

