

# 2023년 1학기 데이터베이스 및 실습

## 9주차 실습과제(총 6문제, 42점)

학번 : 2019161016

이름 : 김나훈

9주차 강의 교안을 참조하여 저장 프로그램을 작성하고 결과를 첨부하시오.

1. 마당 데이터베이스에서 동일 인물인지 확인한 후 고객 정보를 삽입하는 프로시저를 작성하시오(예제5-2, 23page참조). 단 고객의 이름이 같으면 전화번호와 주소를 업데이트 한다(3점).

USE madang;

DELIMITER //

CREATE PROCEDURE customerInsertOrUpdatee(

IN mycustID INTEGER,

IN myname VARCHAR(40),

IN myaddress VARCHAR(50),

IN myphone VARCHAR(20)

)

BEGIN

DECLARE mycount INTEGER;

SELECT COUNT(\*) INTO mycount FROM customer

WHERE name = myname;

IF mycount != 0 THEN

UPDATE customer SET phone = myphone, address = myaddress

```
WHERE name = myname;
```

```
ELSE
```

```
INSERT INTO customer(custid, name, address, phone)
```

```
VALUES(mycustID, myname, myaddress, myphone);
```

```
END IF;
```

```
END;
```

```
//
```

```
DELIMITER ;
```

- 데이터가 없을 때.

```
25 • call customerinsertorupdate1(6, '김나훈', '대한민국', '010-1234-5678');
26 • select * from customer
27
```

Result Grid				Filter Rows: Search	Edit: [Icons]	Export/Import: [Icons]
	custid	name	address	phone		
▶ 1	1	박지성	영국 맨체스터	000-5000-0001		
2	2	김연아	대한민국 서울	000-6000-0001		
3	3	장미란	대한민국 강원도	000-7000-0001		
4	4	추신수	미국 캘리포니아	000-8000-0001		
5	5	박세리	대한민국 대전	010-1111-1111		
6	6	김나훈	대한민국	010-1234-5678		
	NULL	NULL	NULL	NULL		

데이터가 있을 때.

```
25 • call customerinsertorupdate1(6, '김나훈', '미국', '010-1111-1111');
26 • select * from customer
27
```

Result Grid				Filter Rows: Search	Edit: [Icons]	Export/Import: [Icons]
	custid	name	address	phone		
▶ 1	1	박지성	영국 맨체스터	000-5000-0001		
2	2	김연아	대한민국 서울	000-6000-0001		
3	3	장미란	대한민국 강원도	000-7000-0001		
4	4	추신수	미국 캘리포니아	000-8000-0001		
5	5	박세리	대한민국 대전	010-1111-1111		
6	6	김나훈	미국	010-1111-1111		
	NULL	NULL	NULL	NULL		

‘김나훈’, ‘대한민국’, ‘010-1234-5678’ 이  
‘김나훈’, ‘미국’, ‘010-1111-1111’ 로 바뀌었다.

2. Orders 테이블에 저장된 판매가격의 평균을 반환하는 프로시저를 작성하시오(예제 5-3, 25page참조, 3점)

```
delimiter //
```

```
CREATE PROCEDURE averagesalePrice1(
```

```
OUT AverageVal INTEGER)
```

```
BEGIN
```

```
SELECT avg(saleprice) INTO AverageVal
```

```
FROM orders WHERE saleprice IS NOT NULL;
```

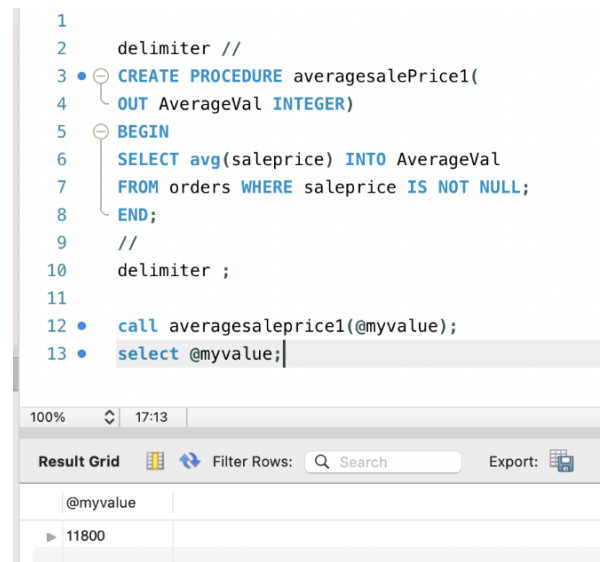
```
END;
```

```
//
```

```
delimiter ;
```

```
call averagesaleprice1(@myvalue);
```

```
select @myvalue;
```



The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
1  
2 delimiter //  
3 CREATE PROCEDURE averagesalePrice1(  
4 OUT AverageVal INTEGER)  
5 BEGIN  
6 SELECT avg(saleprice) INTO AverageVal  
7 FROM orders WHERE saleprice IS NOT NULL;  
8 END;  
9 //  
10 delimiter ;  
11  
12 call averagesaleprice1(@myvalue);  
13 select @myvalue;
```

The result grid shows the output of the last query, which is the value of @myvalue, 11800.

@myvalue
11800

3. Orders 테이블의 판매 도서에 대한 원가를 계산하는 프로시저를 작성하시오. 단 출판사별 원가는 아래와 같다(예제5-4, 36page참조, 6점, 손으로 계산하여 확인하는 것을 권장함)

굿스포츠: 50%

나무수: 40%

대한미디어: 45%

그외: 55%

```
DELIMITER //
```

```
CREATE PROCEDURE calculateCostByPublisher()
```

```
BEGIN
```

```
DECLARE totalCost DECIMAL(10, 2) DEFAULT 0.0;
```

```
DECLARE salePrice DECIMAL(10, 2);
```

```
DECLARE publisherName VARCHAR(255);
```

```
DECLARE endOfRow BOOLEAN DEFAULT FALSE;
```

```
DECLARE costCursor CURSOR FOR
```

```
SELECT o.saleprice, b.publisher
```

```
FROM Orders o
```

```
JOIN Book b ON o.bookid = b.bookid;
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET endOfRow = TRUE;
```

```
OPEN costCursor;
```

```
cursor_loop: LOOP
```

```
FETCH costCursor INTO salePrice, publisherName;
```

```
IF endOfRow THEN
```

```
LEAVE cursor_loop;
```

```
END IF;

IF publisherName = '대한미디어' THEN
SET salePrice = salePrice * 0.45;
ELSEIF publisherName = '굿스포츠' THEN
SET salePrice = salePrice * 0.5;
ELSEIF publisherName = '나무수' THEN
SET salePrice = salePrice * 0.4;
ELSE
SET salePrice = salePrice * 0.55;
END IF;

SET totalCost = totalCost + salePrice;
END LOOP cursor_loop;

CLOSE costCursor;

SELECT CONCAT('전체 원가 = ', totalCost) AS '원가';

END;

//

DELIMITER ;

CALL calculateCostByPublisher();
```

MySQL Workbench

Administration Schemas Query 1

Limit to 1000 rows

Filter objects

madang

- Tables
- Views
- Stored Procedures
- Functions

```

18  END IF;
19  IF publisherName = '대한미디어' THEN
20    SET salePrice = salePrice * 0.45;
21  ELSEIF publisherName = '굿스포츠' THEN
22    SET salePrice = salePrice * 0.5;
23  ELSEIF publisherName = '나무수' THEN
24    SET salePrice = salePrice * 0.4;
25  ELSE
26    SET salePrice = salePrice * 0.55;
27  END IF;
28  SET totalCost = totalCost + salePrice;
29  END LOOP cursor_loop;
30  CLOSE costCursor;
31  SELECT CONCAT('전체 원가 = ', totalCost) AS '원가';
32  END;
33  //
34  DELIMITER ;
35  CALL calculateCostByPublisher();

```

100% 19:17

Result Grid Filter Rows: Search Export:

원가
전체 원가 = 60000.00

4. 새로운 고객정보를 등록해 자동으로 customer log테이블에 삽입한 내용을 기록하는 트리거를 작성하시오(예제5-5, 39page참조, 3점)

```

set global log_bin_trust_function_creators=on;
use madang;
create table customer_log3 (

custid_l integer,
name_l varchar(20),
address_l varchar(20),
phone_l varchar(20)
);
delimiter //

CREATE TRIGGER AfterInsertcustomer3
AFTER INSERT ON customer FOR EACH ROW
BEGIN
INSERT INTO customer_log3
VALUES(new.custid, new.name, new.address, new.phone);
END;
//
delimiter ;

insert into customer values(15,'김나훈','한기대','010-1234-5678');
select * from customer where custid=15;
select * from customer_log1 where custid_l = '15'

customer

```

Query 1

```
8 phone_l varchar(20)
9 );
10 delimiter //
11 • CREATE TRIGGER AfterInsertcustomer3
12 AFTER INSERT ON customer FOR EACH ROW
13 BEGIN
14 INSERT INTO customer_log3
15 VALUES(new.custid, new.name, new.address, new.phone);
16 END;
17 //
18 delimiter ;
19 • insert into customer values(15,'김나훈','한기대','010-1234-5678')
20 • select * from customer where custid=15;
21 • select * from customer_log1 where custid_l = '15'
```

100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import

	custid	name	address	phone
▶ 15	15	김나훈	한기대	010-1234-5678
	NULL	NULL	NULL	NULL

customer 4 customer\_log1 5

## customer\_log

Query 1

```
8 phone_l varchar(20)
9 );
10 delimiter //
11 • CREATE TRIGGER AfterInsertcustomer3
12 AFTER INSERT ON customer FOR EACH ROW
13 BEGIN
14 INSERT INTO customer_log3
15 VALUES(new.custid, new.name, new.address, new.phone);
16 END;
17 //
18 delimiter ;
19 • insert into customer values(15,'김나훈','한기대','010-1234-5678');
20 • select * from customer where custid=15;
21 • select * from customer_log1 where custid_l = '15'
```

100% 1:1

Result Grid Filter Rows: Search Export:

	custid_l	name_l	address_l	phone_l
▶ 15	15	김나훈	한기대	010-1234-5678



## <파이썬 + mysql 연동 과제>

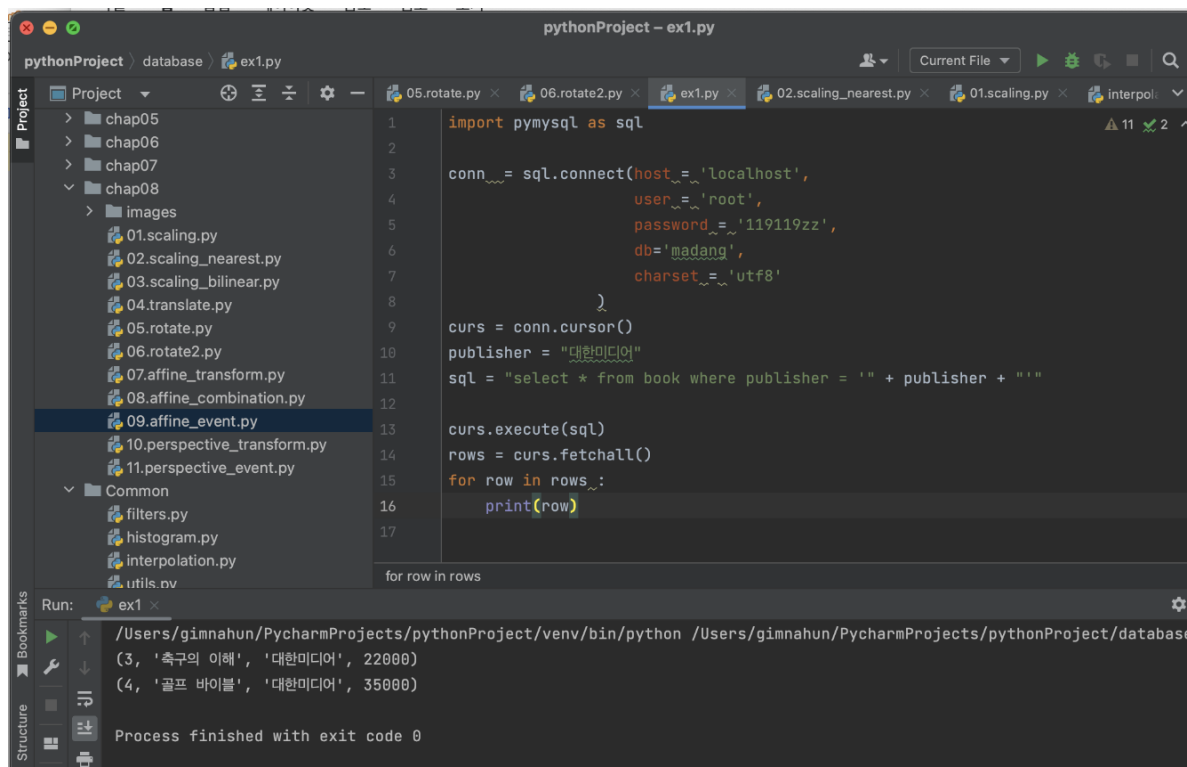
5. “대한미디어”에서 출판한 도서를 검색하는 파이썬 프로그램을 작성하시오(3점)

```
import pymysql as sql

conn = sql.connect(host = 'localhost',
                   user = 'root',
                   password = '119119zz',
                   db='madang',
                   charset = 'utf8'
                   )

curs = conn.cursor()
publisher = "대한미디어"
sql = "select * from book where publisher = '" + publisher + "'"

curs.execute(sql)
rows = curs.fetchall()
for row in rows :
    print(row)
```



6. Book 테이블에 임의의 행을 추가하고 그 행을 삭제하는 파이썬 프로그램을 작성하

시오(3점)

-행 추가 -

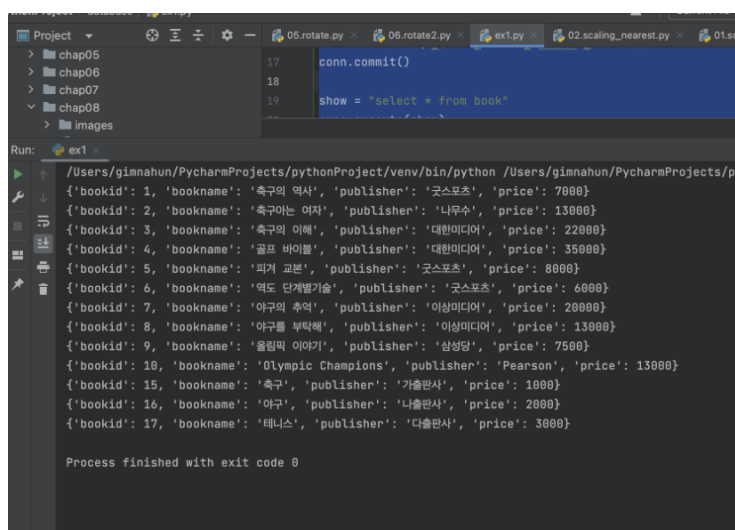
```
import pymysql as sql
import pymysql.cursors

conn = sql.connect(host = 'localhost',
                   user = 'root',
                   password = '119119zz',
                   db='madang',
                   charset = 'utf8'
                   )
curs = conn.cursor(pymysql.cursors.DictCursor)

sql = "insert into book(bookid,bookname,publisher,price) values(%s,%s,%s,%s);"

curs.execute(sql,('15','축구','가출판사','1000'))
curs.execute(sql,('16','야구','나출판사','2000'))
curs.execute(sql,('17','테니스','다출판사','3000'))
conn.commit()

show = "select * from book"
curs.execute(show)
rows = curs.fetchall()
for row in rows :
    print(row)
```



```
conn.commit()
show = "select * from book"

/Users/gimnahun/PycharmProjects/pythonProject/venv/bin/python /Users/gimnahun/PycharmProjects/p
{'bookid': 1, 'bookname': '축구의 역사', 'publisher': '굿스포츠', 'price': 7000}
{'bookid': 2, 'bookname': '축구하는 여자', 'publisher': '나우루', 'price': 13000}
{'bookid': 3, 'bookname': '축구의 이해', 'publisher': '대한미디어', 'price': 22000}
{'bookid': 4, 'bookname': '골프 바이블', 'publisher': '대한미디어', 'price': 35000}
{'bookid': 5, 'bookname': '피겨 교본', 'publisher': '굿스포츠', 'price': 8000}
{'bookid': 6, 'bookname': '역도 단계별기술', 'publisher': '굿스포츠', 'price': 6000}
{'bookid': 7, 'bookname': '야구의 추억', 'publisher': '이상미디어', 'price': 20000}
{'bookid': 8, 'bookname': '야구를 부탁해', 'publisher': '이상미디어', 'price': 13000}
{'bookid': 9, 'bookname': '올림픽 이야기', 'publisher': '삼성당', 'price': 7500}
{'bookid': 10, 'bookname': 'Olympic Champions', 'publisher': 'Pearson', 'price': 13000}
{'bookid': 15, 'bookname': '축구', 'publisher': '가출판사', 'price': 1000}
{'bookid': 16, 'bookname': '야구', 'publisher': '나출판사', 'price': 2000}
{'bookid': 17, 'bookname': '테니스', 'publisher': '다출판사', 'price': 3000}

Process finished with exit code 0
```

- 행 삭제 -

```
import pymysql as sql
import pymysql.cursors
```

```
conn = sql.connect(host = 'localhost',
                   user = 'root',
                   password = '119119zz',
                   db='madang',
                   charset = 'utf8'
                   )
curs = conn.cursor(pymysql.cursors.DictCursor)
```

```
sql = "delete from book where bookid = %d;"
curs.execute(sql % 16)
curs.execute(sql % 17)
conn.commit()
```

```
show = "select * from book"
curs.execute(show)
rows = curs.fetchall()
for row in rows:
    print(row)
```

