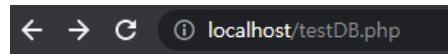


2022 데이터 베이스 Web-Report

숙명매점은 학교 안에 있는 작은 매점이다. 원래는 구매할 상품을 말하면 매점주인이 물건을 꺼내 주고 계산을 했지만, 코로나 사태로 인해 말 대신 웹 키오스크로 주문을 하면 구매한 물건을 받는 방식으로 바뀌었다.

숙명매점의 웹 키오스크 메인 화면



숙명매점

상품번호	상품명	가격	남은수량
1	삼다수	1000	18
2	삼각김밥	1600	12
3	진라면	1000	8
4	왕뚜껑	1000	23
5	불닭볶음면	1200	5
6	가나초콜릿	800	12
7	파워에이드	1900	20
8	제티	700	24
9	오징어땅콩	1300	9
10	하리보젤리	1100	26
11	흰우유	1000	9
12	바나나우유	1200	1
13	커피	900	23
14	에비앙	2000	15
15	포켓몬빵	1500	0
16	초코파이	2400	14
17	백산수	800	6
18	오레오	1900	21
19	진짬뽕	1100	23
20	소세지빵	1000	26
21	크림빵	1000	9
22	자유시간	1400	31
23	전주비빔삼각	1500	9
24	애플잼파이	1100	11
25	빠빠코	1000	52
26	더위사냥	1000	37
27	보석바	1000	28

메인 화면은 아주 간단하게 구성되어 있다.

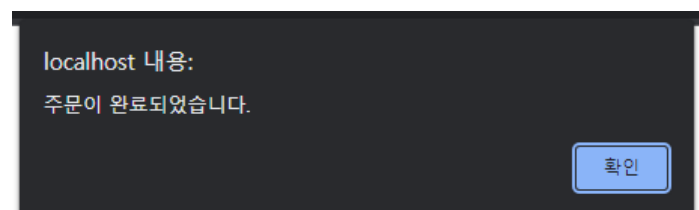
현재 판매하고 있는 품목 27개가 테이블로 정리되어 있고, 아래 텍스트 박스에 구매하고 싶은 상품 이름과 구매할 수량을 입력한 후 버튼을 누르면 다음 단계로 넘어간다.

각 상품들은 고유의 번호를 가지기 때문에 테이블에는 상품 번호, 상품명, 가격, 남은 수량이 표기되어 있다.

구매자는 최대 남은 수량만큼만 상품을 구매할 수 있다.

그럼 물건을 한 번 구매해 보겠다.

보석바 2개를 구매해보겠다.



결제버튼을 누르자 주문이 완료되었다는 알림이 뜬다.

이 상태에서 확인버튼을 누르면

← → ↻ ⓘ localhost/process_order.php?name=보석바&qty=2

주문내역

주문시간: 2022-05-22/21:21:40

상품명: 보석바

주문수량: 2

가격: 2000

현재 남은수량: 26

내가 주문한 시간과 상품명, 주문수량, 총 가격, 현재 남은 수량이 화면에 표시되었다.

내가 주문하기 전에는 보석바의 수량이 28개였는데 26개로 줄어든 것을 볼 수 있다.

← → ↻ ⓘ localhost/testDB.php

숙명매점

상품번호	상품명	가격	남은수량
1	삼다수	1000	18
2	삼각김밥	1600	12
3	진라면	1000	8
4	왕뚜껑	1000	23
5	불닭볶음면	1200	5
6	가나초콜릿	800	12
7	파워에이드	1900	20
8	제티	700	24
9	오징어땅콩	1300	9
10	하리보젤리	1100	26
11	흰우유	1000	9
12	바나나우유	1200	1
13	커피	900	23
14	에비앙	2000	15
15	포켓몬빵	1500	0
16	초코파이	2400	14
17	백산수	800	6
18	오레오	1900	21
19	진짬뽕	1100	23
20	소세지빵	1000	26
21	크림빵	1000	9
22	자유시간	1400	31
23	전주비빔삼각	1500	9
24	애플잼파이	1100	11
25	빠빠코	1000	52
26	더위사냥	1000	37
27	보석바	1000	26

상품명

상품수량

결제

메인 화면에서도 보석바의 수량은 정상적으로 2개가 줄어든 26개로 되어있다.

그런데 만약 내가 남은 수량보다 많이 구매해도 구매가 될까? 한번 해보겠다.

포켓몬빵

5

결제

인기가 많아 품질인 포켓몬 빵을 다섯 개 구매해보겠다.

localhost 내용:

상품 수량이 부족합니다.

확인

상품 수량이 부족해 구매할 수 없었다.

주문내역

주문시간: 2022-05-22/21:28:18

상품명: 포켓몬빵

주문수량: 5

가격: 7500

현재 남은수량: 0

내가 시도했던 주문내역이 뜬다. 포켓몬 빵의 현재 남은 수량은 0개이다.

만약 구매할 물건의 수량을 0개로 입력하면 그것도 주문이 될지 시도해보겠다.

결제

localhost 내용:
0 이상의 수량을 입력해주세요.

확인

의미 없는 주문을 해보았더니 0 이상의 수량을 입력해야 한다는 알림이 나왔다.

결제

localhost 내용:
올바른 상품이름을 입력해주세요.

확인

올바른 상품이름을 입력하라는 메시지와 함께 주문이 되지 않는다.

결제

localhost 내용:

주문이 완료되었습니다.

확인

주문내역

주문시간: 2022-05-22/21:40:44

상품명: 불닭볶음면

주문수량: 5

가격: 6000

현재 남은수량: 0

마지막으로 5개남은 불닭볶음면을 모두 사보았다. 이로 인해 불닭볶음면의 수량은 0개가 남게 되었다.

재고현황				
	Pnum	Pname	Qty	Default
1	1	삼다수	18	40
2	2	삼각김밥	12	12
3	3	진라면	8	20
4	4	왕뚜껑	23	15
5	5	불닭볶음면	0	15
6	6	가나초콜릿	12	10
7	7	파워에이드	20	15
8	8	제티	24	20
9	9	오징어땅콩	9	15
10	10	하리보젤리	26	20
11	11	흰우유	9	10
12	12	바나나우유	1	10
13	13	커피	23	20
14	14	에비앙	15	15
15	15	포켓몬빵	0	30
16	16	초코파이	14	12
17	17	백산수	6	20
18	18	오레오	21	15
19	19	진짬뽕	23	20
20	20	소세지빵	26	25
21	21	크림빵	9	25
22	22	자유시간	31	20
23	23	전주비빔삼각	9	10
24	24	애플잼파이	11	12

[*** Cross Check 점수 신청: 웹 -->고급 검증: 1점 신청 ***]

0개가 된 수량은 파이썬 프로그램의 재고현황에도 제대로 반영되었다. 나머지 상품 수량들도 모두 웹페이지의 메인 화면과 같게 보인다.

매출현황					
	주문시각	상품번호	상품명	수량	가격
1	2022-05-21/21:07:07	21	크림빵	2	2000
2	2022-05-21/21:07:25	19	진짬뽕	1	1100
3	2022-05-21/21:07:38	14	에비앙	3	6000
4	2022-05-21/22:09:18	5	불닭볶음면	2	2400
5	2022-05-21/22:19:00	17	백산수	3	2400
6	2022-05-21/22:19:12	6	가나초콜릿	1	800
7	2022-05-21/22:39:20	10	하리보젤리	2	2200
8	2022-05-22/10:10:29	22	자유시간	3	4200
9	2022-05-22/10:10:56	13	커피	1	900
10	2022-05-22/10:11:08	16	초코파이	2	4800
11	2022-05-22/10:51:11	20	소세지빵	1	1000
12	2022-05-22/14:57:56	18	오레오	1	1900
13	2022-05-22/15:56:56	10	하리보젤리	2	2200
14	2022-05-22/21:21:40	27	보석바	2	2000
15	2022-05-22/21:40:44	5	불닭볶음면	5	6000

위는 파이썬 프로그램의 매출현황 메뉴이다. 조금 전 구매에 성공했던 보석바 2개와 불닭볶음면 5개의 매출기록이 모두 파이썬 프로그램에 정상적으로 업데이트 되어있다.

<내부 데이터베이스>

매점 데이터베이스 테이블은 더 있지만 웹에서 알아야할 데이터베이스는 다음 세가지이다.

Product				
	pnum	pname	price	snum
1	1	삼다수	1000	1
2	2	삼각김밥	1600	2
3	3	진라면	1000	3
4	4	왕뚜껑	1000	3
5	5	불닭볶음면	1200	3
6	6	가나초콜릿	800	5
7	7	파워에이드	1900	6
8	8	제티	700	6
9	9	오징어땅콩	1300	5
10	10	하리보젤리	1100	5
11	11	현우유	1000	7
12	12	바나나우유	1200	7
13	13	커피	900	1
14	14	에비앙	2000	8
15	15	포켓몬빵	1500	9
16	16	초코파이	2400	5
17	17	백산수	800	1
18	18	오레오	1900	5
19	19	진짬뽕	1100	3
20	20	소세지빵	1000	10
21	21	크림빵	1000	10
22	22	자유시간	1400	5
23	23	전주비빔삼각	1500	2
24	24	애플잼파이	1100	9
25	25	빠빠코	1000	4
26	26	더위사냥	1000	4
27	27	보석바	1000	4

Product

상품 테이블이다.

상품번호, 상품명, 상품가격, 공급자번호를 저장한다.

웹에서 이 테이블에서는 공급자번호를 제외한 pnum, pname, price 애트리뷰트를 메인 화면에 표시한다.

Stock

	pnum	qty_current	qty_reorder
1	1	18	40
2	2	12	12
3	3	8	20
4	4	23	15
5	5	0	15
6	6	12	10
7	7	20	15
8	8	24	20
9	9	9	15
10	10	26	20
11	11	9	10
12	12	1	10
13	13	23	20
14	14	15	15
15	15	0	30
16	16	14	12
17	17	6	20
18	18	21	15
19	19	23	20
20	20	26	25
21	21	9	25
22	22	31	20
23	23	9	10
24	24	11	12
25	25	52	30
26	26	37	30
27	27	26	30

재고수량 테이블이다.

상품번호, 현재재고수량, 기본재고수량으로 구성되어있다. 기본재고수량인 qty_reorder보다 재고가 낮아지면 주문서를 작성하는 시스템으로 매점이 운영되고있다.

웹에서는 현재재고수량인 qty_current를 사용한다.

현재재고수량은 pnum으로 앞의 product 테이블과 조인되어 웹페이지 메인 화면에 표시되고 있다.

```
SELECT p.pnum, pname, price, qty_current
      FROM product p,
      stock s
      where p.pnum =
      s.pnum
```

웹언어에서 사용된 쿼리이다.

상품번호, 상품명, 가격, 현재재고를 select해 데이터를 웹에 가져오게 된다.

또한 웹에서 주문이 들어오면 업데이트된 재고수량을 stock 테이블에 업데이트 한다.

```
$query = "SELECT qty_current
      FROM stock
      WHERE pnum = '$pnum'";
```

웹언어에서 사용된 쿼리이다.

salesSlip

	ordertime	pnum	qty	totalprice
1	2022-05-21/21:07:07	21	2	2000
2	2022-05-21/21:07:25	19	1	1100
3	2022-05-21/21:07:38	14	3	6000
4	2022-05-21/22:09:18	5	2	2400
5	2022-05-21/22:19:00	17	3	2400
6	2022-05-21/22:19:12	6	1	800
7	2022-05-21/22:39:20	10	2	2200
8	2022-05-22/10:10:29	22	3	4200
9	2022-05-22/10:10:56	13	1	900
10	2022-05-22/10:11:08	16	2	4800
11	2022-05-22/10:51:11	20	1	1000
12	2022-05-22/14:57:56	18	1	1900
13	2022-05-22/15:56:56	10	2	2200
14	2022-05-22/21:21:40	27	2	2000
15	2022-05-22/21:40:44	5	5	6000

매출기록을 남기는 테이블이다. 주문일시, 주문상품품목, 주문수량, 총가격을 포함한다.

이 테이블은 웹에서 조회하지 않지만, 웹의 주문기록을 이 테이블로 insert한다.

```
INSERT INTO salesSlip
VALUES ('$timestamp', '$pnum', '$qty', '$price')
```

따라서 정상 주문이 이루어질 때마다 salesSlip테이블에 새로운 튜플이 추가된다.

여기까지 내부 구조에 대한 설명을 마친다.

<코드>

main.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<?php

$serverName = "DESKTOP-P5PRM6N";
$connectionOptions = array(
    "database" => "ESQL_project",
    "uid" => "user",
    "pwd" => "0000",
    "CharacterSet" => "UTF-8"
);
```

```

// DB 커넥션
$dbconn = sqlsrv_connect($serverName, $connectionOptions);

?>

<!DOCTYPE html>
<html>
    <body>
        <h1> 숙명매점 </h1>
        <table border="1">
            <tr>
                <td>상품번호</td><td>상품명</td><td>가격</td><td>남은수량</td>
                <?php
                    $query = "SELECT p.pnum, pname, price, qty_current
                               FROM product p, stock s
                               where p.pnum = s.pnum";

                    $stmt = sqlsrv_query($dbconn, $query);

                    while ($row = sqlsrv_fetch_array($stmt, SQLSRV_FETCH_ASSOC)){

                        $filtered = array(

                            'pnum'=>htmlspecialchars($row['pnum']),
                            'pname'=>htmlspecialchars($row['pname']),
                            'price'=>htmlspecialchars($row['price']),
                            'qty_current'=>htmlspecialchars($row['qty_current']));
                        ?>
                        <tr>
                            <td><?=$filtered['pnum']?></td>
                            <td><?=$filtered['pname']?></td>
                            <td><?=$filtered['price']?></td>
                            <td><?=$filtered['qty_current']?></td>
                        </tr>
                    <?php
                    }
                ?>
            </tr>
        </table>
        <form action = "process_order.php" method="get">
            <p><input type="text" name="name" placeholder="상품명"</p>
            <p><input type="text" name="qty" placeholder="상품수량"</p>
            <p><input type="submit" value="결제"></p>
        </form>
    </body>
</html>

<?php

```

```
// statement 해제
sqlsrv_free_stmt($stmt);
// 데이터베이스 접속 해제
sqlsrv_close($dbconn);
?>
```

첫번째 파일에서는 php로 데이터를 불러와 html언어로 테이블을 만들고, text로 상품명과 상품 수량을 입력 받아 Process_order.php파일로 넘기는 것이 핵심기능이다.

Process_order.php

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<?php
$serverName = "DESKTOP-P5PRM6N";
$connectionOptions = array(
    "database" => "ESQL_project",
    "uid" => "user",
    "pwd" => "0000",
    "CharacterSet" => "UTF-8"
);

$name = $_GET["name"];
$qty = $_GET["qty"];
//echo $name;
//echo $qty;

$dbconn = sqlsrv_connect($serverName, $connectionOptions);

$query = "SELECT *
          FROM product
          WHERE pname = '$name'";

$result = sqlsrv_query($dbconn, $query);
$stmt = sqlsrv_query($dbconn, $query);
$row = sqlsrv_fetch_array($stmt, SQLSRV_FETCH_ASSOC);

if($qty>0){
if($row["pnum"]){
    $pnum = $row["pnum"];
    $price = $row["price"] * $qty;
    $query = "SELECT qty_current
              FROM stock
              WHERE pnum = '$pnum'";

$result = sqlsrv_query($dbconn, $query);
$stmt = sqlsrv_query($dbconn, $query);
$row = sqlsrv_fetch_array($stmt, SQLSRV_FETCH_ASSOC);
$qty_current = $row["qty_current"];
```

```

if(($qty_current - $qty)>=0){
    $qty_current = $qty_current - $qty;

    $query = "UPDATE stock
              SET qty_current = '$qty_current'
              WHERE pnum = '$pnum'";
    $result = sqlsrv_query($dbconn, $query);
    $stmt = sqlsrv_query($dbconn, $query);
    if($stmt){

        echo "<script>alert('주문이 완료되었습니다.');

```

이 파일에서는 상품명과 상품수량을 받아 연산을 수행한다.

먼저 입력받은 상품명으로 select하여 상품 번호와 가격 등의 정보를 데이터로 받는다. 받은 pnum이 유효한지 체크하고 총가격\$price를 계산하고, pnum을 이용해 stock테이블에도 현재재고

를 업데이트한다.

코드발췌

```
if($row["pnum"]){
    $pnum = $row["pnum"];
    $price = $row["price"] * $qty;
    $query = "SELECT qty_current
              FROM stock
              WHERE pnum = '$pnum'";

    $result = sqlsrv_query($dbconn, $query);
    $stmt = sqlsrv_query($dbconn, $query);
    $row = sqlsrv_fetch_array($stmt, SQLSRV_FETCH_ASSOC);
    $qty_current = $row["qty_current"];

    if(($qty_current - $qty)>=0){
        $qty_current = $qty_current - $qty;

        $query = "UPDATE stock
                  SET qty_current = '$qty_current'
                  WHERE pnum = '$pnum'";
        $result = sqlsrv_query($dbconn, $query);
        $stmt = sqlsrv_query($dbconn, $query);
    }
}
```

\$stmt가 유효하다면 연산이 제대로 실행된 것이므로 주문이 완료되었다고 본다. 현재 주문완료 시각을 \$timestamp변수로 받고, 주문일시, 상품번호, 수량, 총가격을 salesSlip테이블에 insert하는 연산을 진행한다.

```
if($stmt){
    echo "<script>alert('주문이 완료되었습니다.');
```

else문을 이용해 상단에서 설명한 예외처리도 하였다.

<진행일지>-고급, 웹 언어 합쳐져 있습니다.

언어 & 주제 정하기

매점관리 프로그램을 만들기로 정하였다. 먼저 웹에서는 데이터베이스에서 데이터를 받아 메뉴를 만들고 주문을 받는 것을 구현한다. 고급언어는 매점 관리자용 프로그램으로, 매출전표와 재고 공급자 등을 확인할 수 있다. 또한 재고나 신상품 입고 기능도 프로그램으로 구현할 수 있을 것 같다. 웹언어는 많이 다뤄보지 않아서 잘 모르겠지만 php나 javascript중에서 정할 것 같고 고급언어로는 python을 사용할 예정이다. python으로는 tkinter 모듈을 사용해 단순한 GUI를 구현할 것이다.

5/14

데이터 작성

상품(상품번호, 상품명, 가격, 공급자번호)
product(pnum,pname,price,snum)

공급자(공급자번호, 공급자이름, 전화번호)
supplier(snum,sname,phone)

물품재고리스트(상품번호,현재재고수량, 재주문 재고수량)
stock(pnum,qty_current,qty_reorder)

매출전표(상품번호, 수량, 총가격) -> 웹에서 주문 들어오면 상품 수량 업데이트
salesSlip(pnum,qty,totalprice)

도매가격(상품번호, 도매가격, 도매주문수량)
wholesale(pnum,wprice,qty)

부 -> 재고주문서(상품번호, 상품명, 주문수량, 도매가격, 공급자이름, 공급자전화번호)

데이터베이스 스키마를 간략히 메모장에 작성하였다. mssql에서 DDL언어로 테이블과 인스턴스를 구현하였다. 나중에 아마 수정이 필요할 것 같지만 일단 매출전표 테이블인 salesSlip을 제외하고는 데이터를 모두 채워두었다.

5/17

Pymssql 모듈을 설치하고 조교님이 올려 주신 자료를 참고해 파이썬에서 데이터를 불러와보았다. 서버 이름 때문에 오류가 많이 났지만 가능한 서버이름을 모조리 입력해보다가 해결했다.

5/18

파이썬에서 메인 페이지를 만들었다. Tkinter를 사용해 간단한 GUI를 구현할 것이다. 파이썬에서는 매점 관리자용 프로그램을 구현할 것이라, 재고현황, 매출현황, 재고주문서, 재고입고, 신상품입고 다섯가지의 버튼을 우선 만들었다. 먼저 메인 윈도우를 만들고 ttk.Button구문을 사용해 버튼을 추가하였다. 아직 연결은 하지 않았다. 재고현황, 매출현황, 재고주문서 버튼을 누르면 테이블이 조회되게 select구문을 쿼리로 날릴 것이다. 재고 입고는 만들어 놓은 stock테이블에 현재 재고를

추가하는 update set 구문을 사용할 것 같고, 신상품입고는 새 튜플을 추가해야되니까 insert into 구문을 사용하게 될 것이다.

5/19

파이썬에서 버튼을 누르면 실행되는 함수 get_data를 만들었다. 버튼마다 가지는 코드에 따라 sql 쿼리가 달라지게 해 데이터를 받는 함수이다. 재고현황, 매출현황, 재고주문서 3가지 버튼을 누르면 각각의 select 쿼리가 실행된다. 새 창을 여는 클래스 Make_newWindow도 만들었다. Get_data 함수에서 데이터를 받아 리스트형태로 바꿔 Make_newWindow로 보내 tk.treeview를 사용해 테이블을 작성할 것이다. 오늘은 데이터를 받아 클래스로 보내는 것까지 진행하였다.

5/20

전에 재고주문서를 stock_view로 작성해두고 그걸 get_data에서 select 하게 했었는데, HLL에서 4개의 테이블 조인을 포함해야 할 것 같아서 뷰 대신 직접 조인하는 select구문이 파이썬에서 실행되게 하였다. Make_newWindow안에 있는 make_table함수에서 각각의 테이블이 만들어진다. if 문으로 각 버튼에 맞는 메시지가 조절된다.

5/21

Php와 sqlsrv를 설치했다. Apache도 설치했다. php.ini와 httpd.conf를 수정해 php와 apache를 연결하는 것은 했지만 데이터베이스와 연동이 되지 않아 고생했다. Vscode를 설치하니까 확실히 편리했다. php파일을 다시 수정하고 데이터베이스와 연동을 했지만 이번엔 한글이 깨졌다. 한참동안 시도한 결과 "CharacterSet" => "UTF-8"을 추가하고 한글이 나왔다. 테이블을 만들려면 html이 있어야 해서 또 html 확장을 설치하고 테이블을 만들었다. 상품 주문을 위한 textbox를 넣는 것은 생각보다 간단했다. 데이터를 처음에는 post로 받으려고 했는데 자꾸 오류가 나서 get으로 받는 것으로 변경했다. 상품명과 수량을 입력 받아 상품명으로 select해 key인 pnum을 먼저 찾고 오류사항 처리를 시작했다. 주문수량이 0일 때, 상품번호가 없을 때(상품이름을 잘못 입력했을 때), 재고수량이 부족할 때 등을 염두에 두고 코드를 작성하였다. 올바른 입력일 시 상품번호에 맞는 pnum을 이용해 재고 테이블인 stock에 입력 받은 만큼의 qty를 빼는 update연산을 실시했다. 연산이 정상적으로 시행되면 주문이 정상적으로 완료된 것으로 하고 현재 날짜시간, 상품번호, 구매수량, 총가격을 매출전표인 salesSlip 테이블에 Insert한다. salesSlip테이블은 처음에 작성한 스키마에서 약간 수정해 현재날짜/시간과 주문한 상품번호를 key로 가지게 하였다. 총 가격은 앞서 진행한 select구문에서 상품의 price를 받아 구매수량과 곱해 얻었다. 각종 예외 알림 alert와 echo등을 한 후 주문내역기록을 뜨게 하였다. 상품을 주문하면 재고수량이 줄어드는 것을 확인하고 웹 페이지 구현은 마무리했다.

그 다음 해야 할 것은 파이썬이었다. 4번 버튼인 재고입고 버튼의 기능을 만드는 차례였다. 입고할 재고의 상품번호를 입력하는 방법은 combobox가 좋을 것 같아서 4번버튼도 get_data함수에서 pnum리스트를 받은 뒤 Make_newWindow 클래스로 넘겨받아 get_stock함수가 호출되게 하였

다. 콤보박스과 텍스트박스를 만들고 확인버튼을 만들었다. 오늘은 여기까지.

그리고 만들어진 웹페이지에서 재고와 매출전표가 업데이트 되는 것은 파이썬에서도 확인 가능했다.

5/22

어제 콤보박스과 텍스트에서 입력 받은 pnum, qty로 재고수량이 업데이트 되게 하였다. Update set 구문을 적어 재고 테이블에서 현재재고인 qty_current에 입력 받은 qty가 더해지게 하였다. 웹 페이지와는 반대되는 연산이다. 파이썬 프로그램에서 재고를 추가해주면 웹 페이지에도 재고가 업데이트 되어 주문할 수 있게 되었다. 이제 마지막 신상품 추가 기능이 문제였다. 신상품을 추가 하면 상품테이블인 product테이블에 추가가 되어야 한다. 그 뿐만 아니라 연관된 테이블인 stock(재고), wholesale(도매), supplier(공급자) 테이블에도 모두 새로운 정보가 들어가야 한다. 그래서 고민하다가 필요한 모든 정보를 결국 입력을 받기로 하였다. 그런데 또 문제가 공급자는 새로운 공급자일수도 있고, 기존의 공급자일수도 있다는 것이었다. 따라서 먼저 공급자번호를 받아서 앞선 여부를 체크하고, 새로운 공급자이면 나머지 공급자이름, 전화번호 정보를 받고 아니면 그대로 진행되게 하였다. 우려했던 여러 insert 구문들은 잘 실행되는듯 보였다. 하지만 삽입한 데이터에서만 또 한글이 깨지기 시작했다. 이 문제로 네다섯시간을 고민했다. 그리고 아직 진행중이다. - > 생각해보니까 꼭 한글로 insert할 필요는 없는 것 같다.

5/23

보고서를 쓰다가 갑자기 재고주문서를 내보내는 기능을 넣어보고 싶어서 추가해보았다. 데이터 프레임을 내보내는 게 생각보다 더 간단하여서 금방 했다.

<reference>

<https://github.com/web-n/web3-PHP-mysql/commit/cfd82024c7333523357982a48248b772245ee1b7>

<https://github.com/web-n/web3-PHP-mysql/commit/33ae5459b5cc27d6f387040a40a0e66d6732ab6e>