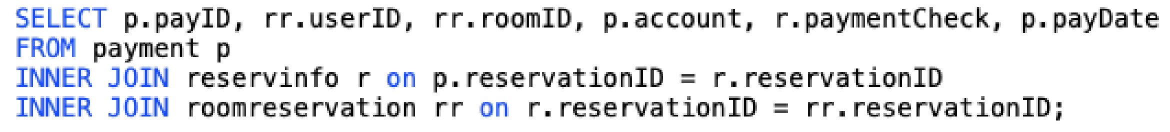
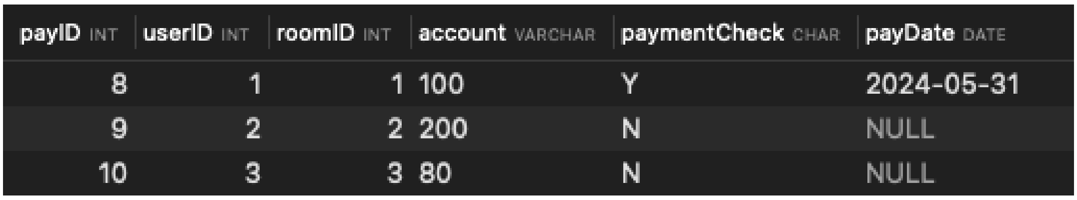
결제

REQ503 결제의 완료, 취소, 환불 여부를 확인

```
SELECT p.payID, rr.userID, rr.roomID, p.account, r.paymentCheck, p.payDate
FROM payment p
INNER JOIN reservinfo r on p.reservationID = r.reservationID
INNER JOIN roomreservation rr on r.reservationID = rr.reservationID;
```



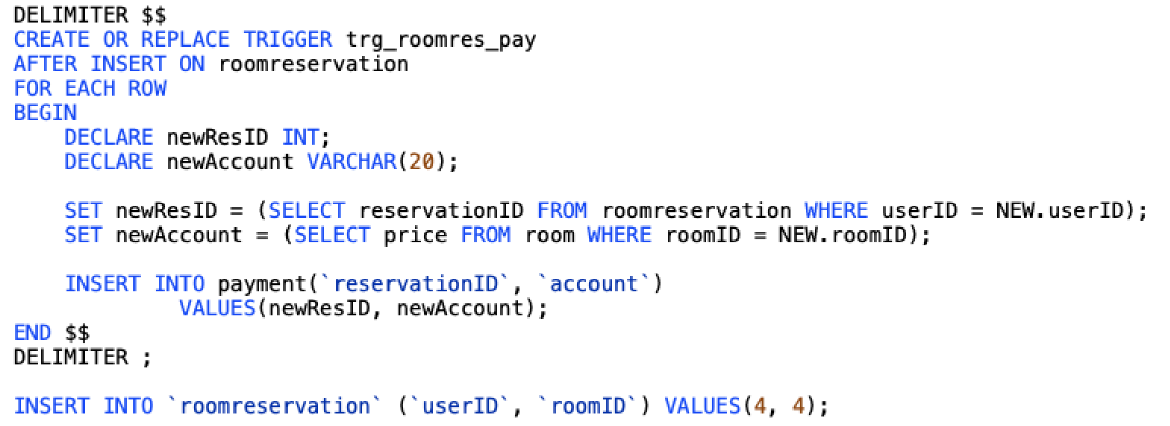
```
SELECT p.payID, rr.userID, rr.roomID, p.account, r.paymentCheck, p.payDate
FROM payment p
INNER JOIN reservinfo r on p.reservationID = r.reservationID
INNER JOIN roomreservation rr on r.reservationID = rr.reservationID;
```



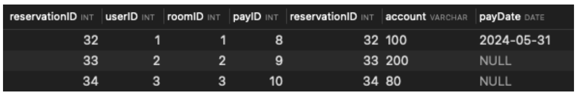
payID INT	userID INT	roomID INT	account VARCHAR	paymentCheck CHAR	payDate DATE
8	1	1	100	Y	2024-05-31
9	2	2	200	N	NULL
10	3	3	80	N	NULL

REQ504 스터디룸을 예약하면 결제 이력이 추가될 수 있어야 한다.

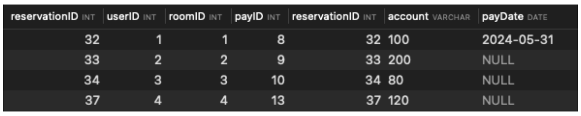
```
DELIMITER $$
CREATE OR REPLACE TRIGGER trg_roomres_pay
AFTER INSERT ON roomreservation
FOR EACH ROW
BEGIN
        DECLARE newResID INT;
        DECLARE newAccount VARCHAR(20);

        SET newResID = (SELECT reservationID FROM roomreservation WHERE userID
= NEW.userID);
        SET newAccount = (SELECT price FROM room WHERE roomID = NEW.roomID);

        INSERT INTO payment(`reservationID`, `account`)
                        VALUES(newResID, newAccount);
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE OR REPLACE TRIGGER trg_roomres_pay
AFTER INSERT ON roomreservation
FOR EACH ROW
BEGIN
    DECLARE newResID INT;
    DECLARE newAccount VARCHAR(20);

    SET newResID = (SELECT reservationID FROM roomreservation WHERE userID = NEW.userID);
    SET newAccount = (SELECT price FROM room WHERE roomID = NEW.roomID);

    INSERT INTO payment(`reservationID`, `account`)
             VALUES(newResID, newAccount);
END $$
DELIMITER ;

INSERT INTO `roomreservation` (`userID`, `roomID`) VALUES(4, 4);
```



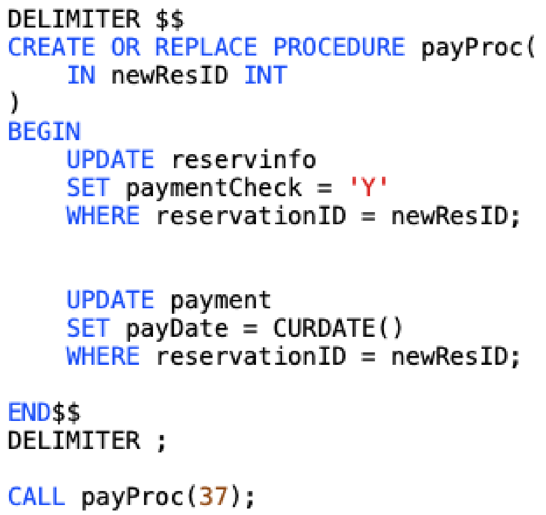
reservationID INT	userID INT	roomID INT	payID INT	reservationID INT	account VARCHAR	payDate DATE
32	1	1	8	32	100	2024-05-31
33	2	2	9	33	200	NULL
34	3	3	10	34	80	NULL



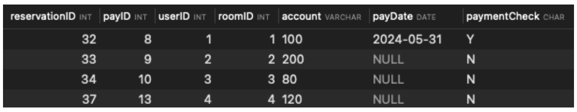
reservationID INT	userID INT	roomID INT	payID INT	reservationID INT	account VARCHAR	payDate DATE
32	1	1	8	32	100	2024-05-31
33	2	2	9	33	200	NULL
34	3	3	10	34	80	NULL
37	4	4	13	37	120	NULL

REQ505 회원이 결제를 하면 결제일과 결제여부를 변경한다.

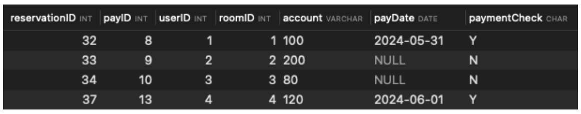
```
DELIMITER $$
CREATE OR REPLACE PROCEDURE payProc(
        IN newResID INT
)
BEGIN
        UPDATE reservinfo
        SET paymentCheck = 'Y'
        WHERE reservationID = newResID;


        UPDATE payment
        SET payDate = CURDATE()
        WHERE reservationID = newResID;

END$$
DELIMITER ;
```



```
DELIMITER $$
CREATE OR REPLACE PROCEDURE payProc(
    IN newResID INT
)
BEGIN
    UPDATE reservinfo
    SET paymentCheck = 'Y'
    WHERE reservationID = newResID;


    UPDATE payment
    SET payDate = CURDATE()
    WHERE reservationID = newResID;

END$$
DELIMITER ;

CALL payProc(37);
```



reservationID INT	payID INT	userID INT	roomID INT	account VARCHAR	payDate DATE	paymentCheck CHAR
32	8	1	1	100	2024-05-31	Y
33	9	2	2	200	NULL	N
34	10	3	3	80	NULL	N
37	13	4	4	120	NULL	N

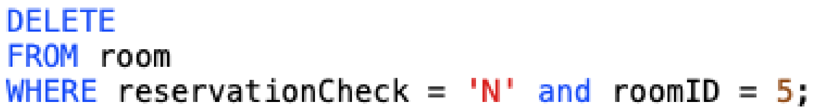


reservationID INT	payID INT	userID INT	roomID INT	account VARCHAR	payDate DATE	paymentCheck CHAR
32	8	1	1	100	2024-05-31	Y
33	9	2	2	200	NULL	N
34	10	3	3	80	NULL	N
37	13	4	4	120	2024-06-01	Y

스터디룸

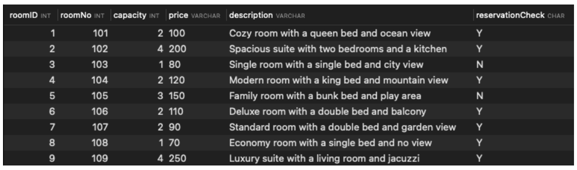
REQ604 특정 스터디룸을 삭제할 수 있다

```
DELETE
FROM room
WHERE reservationCheck = 'N' and roomID = 10;
```



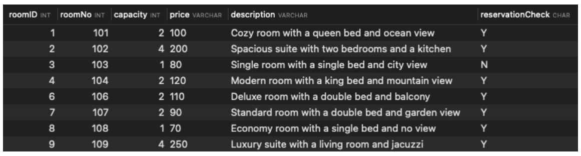
```
DELETE
FROM room
WHERE reservationCheck = 'N' and roomID = 5;
```

select * from room;



roomID INT	roomNo INT	capacity INT	price VARCHAR	description VARCHAR	reservationCheck CHAR
1	101	2	100	Cozy room with a queen bed and ocean view	Y
2	102	4	200	Spacious suite with two bedrooms and a kitchen	Y
3	103	1	80	Single room with a single bed and city view	N
4	104	2	120	Modern room with a king bed and mountain view	Y
5	105	3	150	Family room with a bunk bed and play area	N
6	106	2	110	Deluxe room with a double bed and balcony	Y
7	107	2	90	Standard room with a double bed and garden view	Y
8	108	1	70	Economy room with a single bed and no view	Y
9	109	4	250	Luxury suite with a living room and jacuzzi	Y

select * from room;

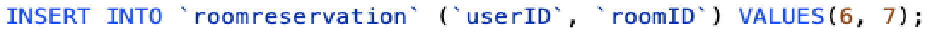


roomID INT	roomNo INT	capacity INT	price VARCHAR	description VARCHAR	reservationCheck CHAR
1	101	2	100	Cozy room with a queen bed and ocean view	Y
2	102	4	200	Spacious suite with two bedrooms and a kitchen	Y
3	103	1	80	Single room with a single bed and city view	N
4	104	2	120	Modern room with a king bed and mountain view	Y
6	106	2	110	Deluxe room with a double bed and balcony	Y
7	107	2	90	Standard room with a double bed and garden view	Y
8	108	1	70	Economy room with a single bed and no view	Y
9	109	4	250	Luxury suite with a living room and jacuzzi	Y

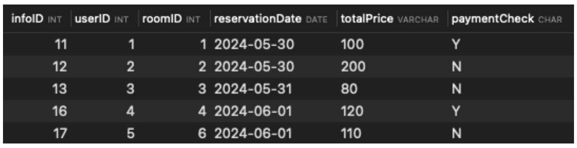
스터디룸예약

REQ701 회원은 당일 날짜에 스터디룸을 예약할 수 있다

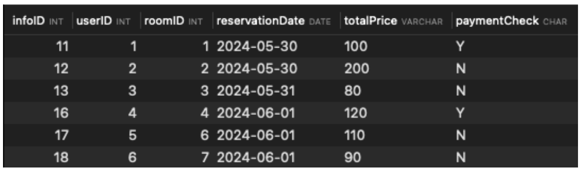
INSERT INTO `roomreservation` into (`userID`, `roomID`) VALUES(6,7);



```
INSERT INTO `roomreservation` (`userID`, `roomID`) VALUES(6, 7);
```



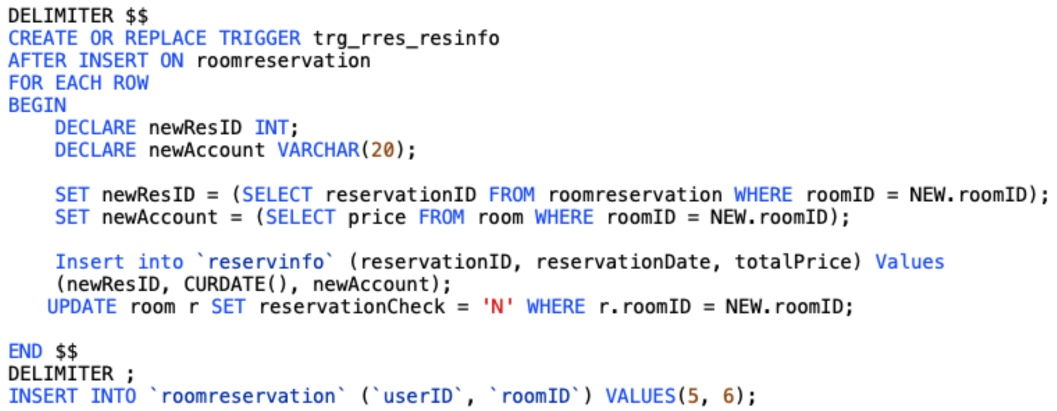
infoID INT	userID INT	roomID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	1	1	2024-05-30	100	Y
12	2	2	2024-05-30	200	N
13	3	3	2024-05-31	80	N
16	4	4	2024-06-01	120	Y
17	5	6	2024-06-01	110	N



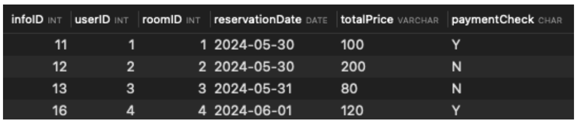
infoID INT	userID INT	roomID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	1	1	2024-05-30	100	Y
12	2	2	2024-05-30	200	N
13	3	3	2024-05-31	80	N
16	4	4	2024-06-01	120	Y
17	5	6	2024-06-01	110	N
18	6	7	2024-06-01	90	N

REQ703 회원이 스터디룸을 예약하면 예약 정보를 생성한다
REQ711 스터디룸을 예약하면 스터디룸의 상태를 변경한다

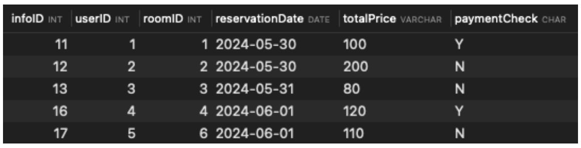
```
DELIMITER $$
CREATE OR REPLACE TRIGGER trg_rres_resinfo
AFTER INSERT ON roomreservation
FOR EACH ROW
BEGIN
        DECLARE newResID INT;
        DECLARE newAccount VARCHAR(20);

        SET newResID = (SELECT reservationID FROM roomreservation WHERE roomID = NEW.roomID);
        SET newAccount = (SELECT price FROM room WHERE roomID = NEW.roomID);

        Insert into `reservinfo` (reservationID, reservationDate, totalPrice) Values
        (newResID, CURDATE(), newAccount);
        UPDATE room r SET reservationCheck = 'N' WHERE r.roomID = NEW.roomID;
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE OR REPLACE TRIGGER trg_rres_resinfo
AFTER INSERT ON roomreservation
FOR EACH ROW
BEGIN
    DECLARE newResID INT;
    DECLARE newAccount VARCHAR(20);

    SET newResID = (SELECT reservationID FROM roomreservation WHERE roomID = NEW.roomID);
    SET newAccount = (SELECT price FROM room WHERE roomID = NEW.roomID);

    Insert into `reservinfo` (reservationID, reservationDate, totalPrice) Values
    (newResID, CURDATE(), newAccount);
   UPDATE room r SET reservationCheck = 'N' WHERE r.roomID = NEW.roomID;

END $$
DELIMITER ;
INSERT INTO `roomreservation` (`userID`, `roomID`) VALUES(5, 6);
```



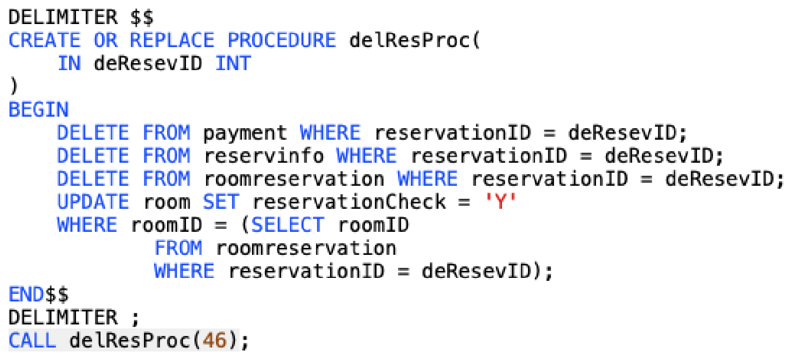
infoID INT	userID INT	roomID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	1	1	2024-05-30	100	Y
12	2	2	2024-05-30	200	N
13	3	3	2024-05-31	80	N
16	4	4	2024-06-01	120	Y



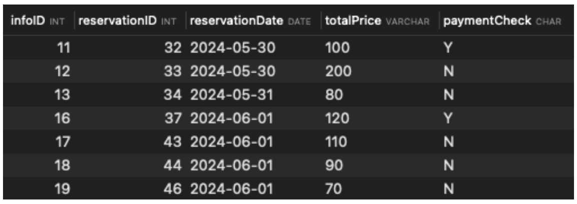
infoID INT	userID INT	roomID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	1	1	2024-05-30	100	Y
12	2	2	2024-05-30	200	N
13	3	3	2024-05-31	80	N
16	4	4	2024-06-01	120	Y
17	5	6	2024-06-01	110	N

REQ707 회원이 예약한 스터디룸을 취소하면 결제와 예약 정보 테이블에서도 삭제되어야 한다

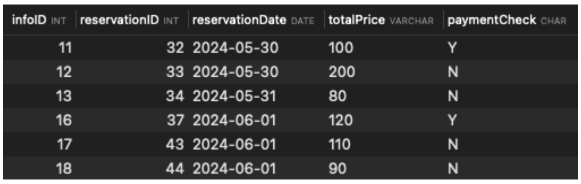
```
-- 결제, 예약 정보, 예약 순서대로 삭제해야 함
DELIMITER $$
CREATE OR REPLACE PROCEDURE delResProc(
        IN deResevID INT
)
BEGIN
        DELETE FROM payment WHERE reservationID = deResevID;
        DELETE FROM reservinfo WHERE reservationID = deResevID;
        DELETE FROM roomreservation WHERE reservationID = deResevID;
        UPDATE room SET reservationCheck = 'Y'
        WHERE roomID = (SELECT roomID
                        FROM roomreservation
                        WHERE reservationID = deResevID);
END$$
DELIMITER ;
```



```
DELIMITER $$
CREATE OR REPLACE PROCEDURE delResProc(
    IN deResevID INT
)
BEGIN
    DELETE FROM payment WHERE reservationID = deResevID;
    DELETE FROM reservinfo WHERE reservationID = deResevID;
    DELETE FROM roomreservation WHERE reservationID = deResevID;
    UPDATE room SET reservationCheck = 'Y'
    WHERE roomID = (SELECT roomID
            FROM roomreservation
            WHERE reservationID = deResevID);
END$$
DELIMITER ;
CALL delResProc(46);
```



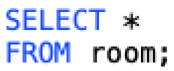
infoID INT	reservationID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	32	2024-05-30	100	Y
12	33	2024-05-30	200	N
13	34	2024-05-31	80	N
16	37	2024-06-01	120	Y
17	43	2024-06-01	110	N
18	44	2024-06-01	90	N
19	46	2024-06-01	70	N



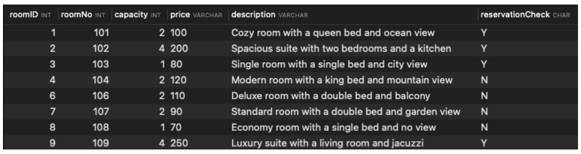
infoID INT	reservationID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	32	2024-05-30	100	Y
12	33	2024-05-30	200	N
13	34	2024-05-31	80	N
16	37	2024-06-01	120	Y
17	43	2024-06-01	110	N
18	44	2024-06-01	90	N

REQ709 스터디룸이 예약이 되었는지 예약이 가능한지 예약 상태 여부를 확인할 수 있다

```
SELECT *
FROM room;
```



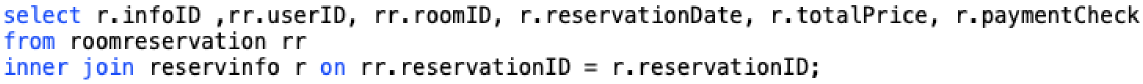
```
SELECT *
FROM room;
```



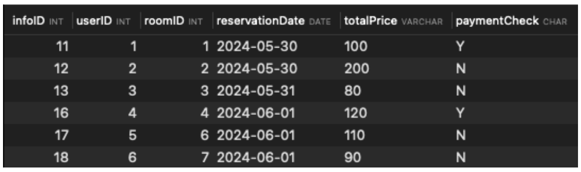
roomID INT	roomNo INT	capacity INT	price VARCHAR	description VARCHAR	reservationCheck CHAR
1	101	2	100	Cozy room with a queen bed and ocean view	Y
2	102	4	200	Spacious suite with two bedrooms and a kitchen	Y
3	103	1	80	Single room with a single bed and city view	Y
4	104	2	120	Modern room with a king bed and mountain view	N
6	106	2	110	Deluxe room with a double bed and balcony	N
7	107	2	90	Standard room with a double bed and garden view	N
8	108	1	70	Economy room with a single bed and no view	N
9	109	4	250	Luxury suite with a living room and jacuzzi	Y

REQ710 스터디룸 예약이 되어있는지 정보 확인할 수 있다

```
select r.infoID ,rr.userID, rr.roomID, r.reservationDate, r.totalPrice, r.paymentCheck
from roomreservation rr
inner join reservinfo r on rr.reservationID = r.reservationID;
```



```
select r.infoID ,rr.userID, rr.roomID, r.reservationDate, r.totalPrice, r.paymentCheck
from roomreservation rr
inner join reservinfo r on rr.reservationID = r.reservationID;
```

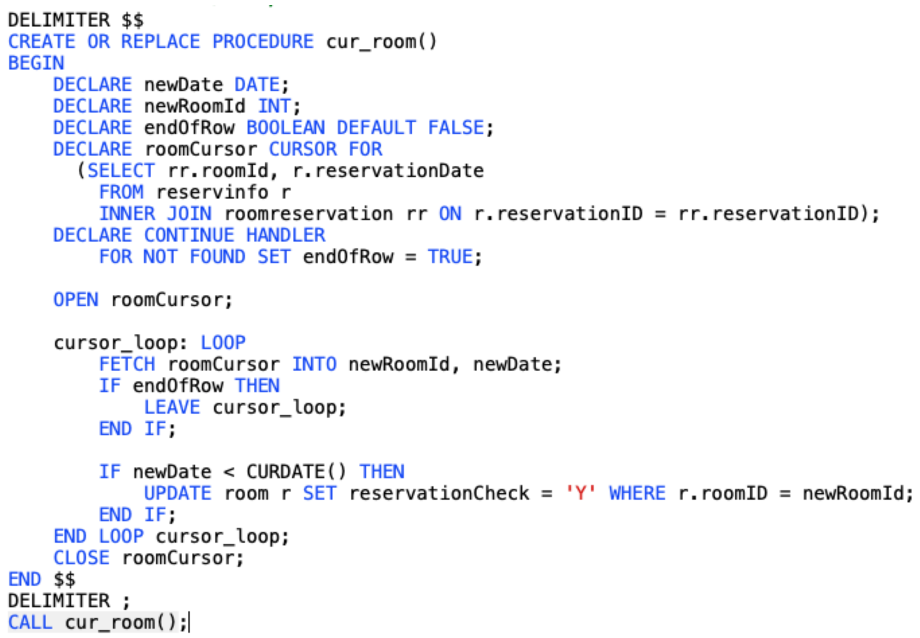


infoID INT	userID INT	roomID INT	reservationDate DATE	totalPrice VARCHAR	paymentCheck CHAR
11	1	1	2024-05-30	100	Y
12	2	2	2024-05-30	200	N
13	3	3	2024-05-31	80	N
16	4	4	2024-06-01	120	Y
17	5	6	2024-06-01	110	N
18	6	7	2024-06-01	90	N

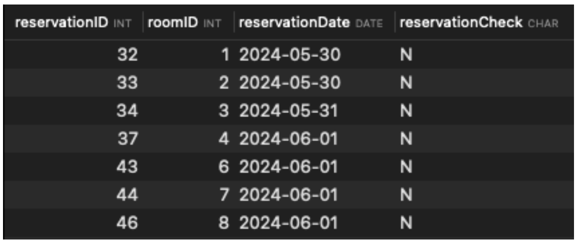
REQ712 스터디룸을 예약이 종료되면 스터디룸의 상태를 변경한다

```
-- 예약 날짜 지난 것 확인하고, 스터디룸 예약 상태 변경하기
DELIMITER $$
CREATE OR REPLACE PROCEDURE cur_room()
BEGIN
   DECLARE newDate DATE;
   DECLARE newRoomId INT;
   DECLARE endOfRow BOOLEAN DEFAULT FALSE;
   DECLARE roomCursor CURSOR FOR
     (SELECT rr.roomId, r.reservationDate
                    FROM reservinfo r
                    INNER JOIN roomreservation rr ON r.reservationID = rr.reservationID);
   DECLARE CONTINUE HANDLER
       FOR NOT FOUND SET endOfRow = TRUE;

   OPEN roomCursor;
```

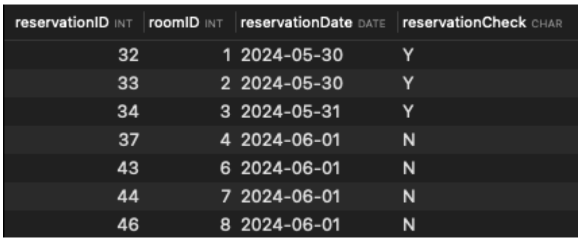
```
    cursor_loop: LOOP
        FETCH roomCursor INTO newRoomId, newDate;
        IF endOfRow THEN
            LEAVE cursor_loop;
        END IF;

        IF newDate < CURDATE() THEN
            UPDATE room r SET reservationCheck = 'Y' WHERE r.roomID = newRoomId;
        END IF;
    END LOOP cursor_loop;
    CLOSE roomCursor;
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE OR REPLACE PROCEDURE cur_room()
BEGIN
    DECLARE newDate DATE;
    DECLARE newRoomId INT;
    DECLARE endOfRow BOOLEAN DEFAULT FALSE;
    DECLARE roomCursor CURSOR FOR
      (SELECT rr.roomId, r.reservationDate
        FROM reservinfo r
        INNER JOIN roomreservation rr ON r.reservationID = rr.reservationID);
    DECLARE CONTINUE HANDLER
        FOR NOT FOUND SET endOfRow = TRUE;

    OPEN roomCursor;

    cursor_loop: LOOP
        FETCH roomCursor INTO newRoomId, newDate;
        IF endOfRow THEN
            LEAVE cursor_loop;
        END IF;

        IF newDate < CURDATE() THEN
            UPDATE room r SET reservationCheck = 'Y' WHERE r.roomID = newRoomId;
        END IF;
    END LOOP cursor_loop;
    CLOSE roomCursor;
END $$
DELIMITER ;
CALL cur_room();
```



reservationID INT	roomID INT	reservationDate DATE	reservationCheck CHAR
32	1	2024-05-30	N
33	2	2024-05-30	N
34	3	2024-05-31	N
37	4	2024-06-01	N
43	6	2024-06-01	N
44	7	2024-06-01	N
46	8	2024-06-01	N

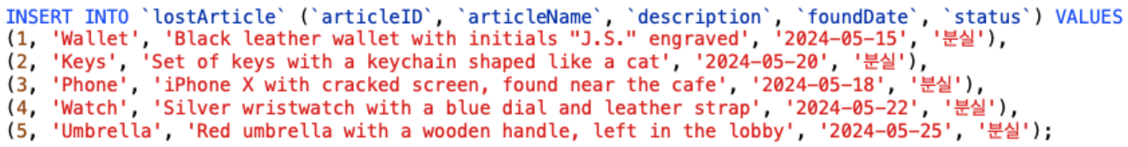


reservationID INT	roomID INT	reservationDate DATE	reservationCheck CHAR
32	1	2024-05-30	Y
33	2	2024-05-30	Y
34	3	2024-05-31	Y
37	4	2024-06-01	N
43	6	2024-06-01	N
44	7	2024-06-01	N
46	8	2024-06-01	N

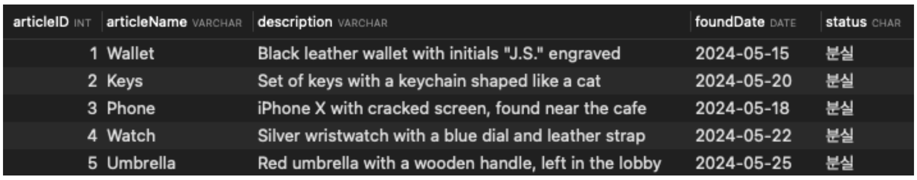
분실물관리

REQ802 분실물 정보를 입력하여 분실물을 추가한다.

```
INSERT INTO `lostArticle` (`articleID`, `articleName`, `description`, `foundDate`, `status`)
VALUES
(1, 'Wallet', 'Black leather wallet with initials "J.S." engraved', '2024-05-15', '분실'),
(2, 'Keys', 'Set of keys with a keychain shaped like a cat', '2024-05-20', '분실'),
(3, 'Phone', 'iPhone X with cracked screen, found near the cafe', '2024-05-18', '분실'),
(4, 'Watch', 'Silver wristwatch with a blue dial and leather strap', '2024-05-22', '분실'),
(5, 'Umbrella', 'Red umbrella with a wooden handle, left in the lobby', '2024-05-25', '분실');
```



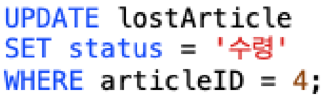
```
INSERT INTO `lostArticle` (`articleID`, `articleName`, `description`, `foundDate`, `status`) VALUES
(1, 'Wallet', 'Black leather wallet with initials "J.S." engraved', '2024-05-15', '분실'),
(2, 'Keys', 'Set of keys with a keychain shaped like a cat', '2024-05-20', '분실'),
(3, 'Phone', 'iPhone X with cracked screen, found near the cafe', '2024-05-18', '분실'),
(4, 'Watch', 'Silver wristwatch with a blue dial and leather strap', '2024-05-22', '분실'),
(5, 'Umbrella', 'Red umbrella with a wooden handle, left in the lobby', '2024-05-25', '분실');
```



articleID INT	articleName VARCHAR	description VARCHAR	foundDate DATE	status CHAR
1	Wallet	Black leather wallet with initials "J.S." engraved	2024-05-15	분실
2	Keys	Set of keys with a keychain shaped like a cat	2024-05-20	분실
3	Phone	iPhone X with cracked screen, found near the cafe	2024-05-18	분실
4	Watch	Silver wristwatch with a blue dial and leather strap	2024-05-22	분실
5	Umbrella	Red umbrella with a wooden handle, left in the lobby	2024-05-25	분실

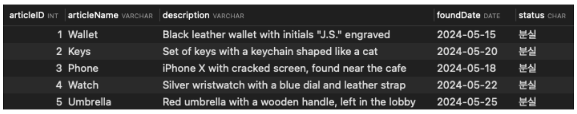
REQ803 분실물을 찾아갔을 경우, 분실물 상태를 변경한다.

```
UPDATE lostArticle
SET status = '수령'
WHERE articleID = 4;
```



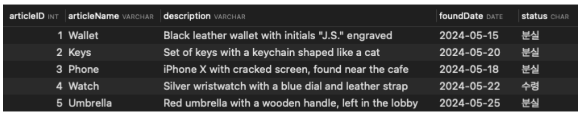
```
UPDATE lostArticle
SET status = '수령'
WHERE articleID = 4;
```

SELECT * FROM lostArticle;



articleID INT	articleName VARCHAR	description VARCHAR	foundDate DATE	status CHAR
1	Wallet	Black leather wallet with initials "J.S." engraved	2024-05-15	분실
2	Keys	Set of keys with a keychain shaped like a cat	2024-05-20	분실
3	Phone	iPhone X with cracked screen, found near the cafe	2024-05-18	분실
4	Watch	Silver wristwatch with a blue dial and leather strap	2024-05-22	분실
5	Umbrella	Red umbrella with a wooden handle, left in the lobby	2024-05-25	분실

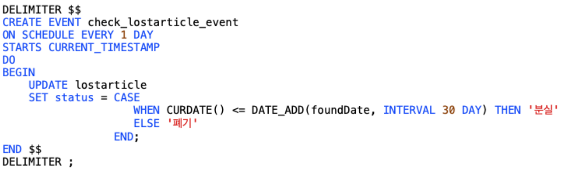
SELECT * FROM lostArticle;



articleID INT	articleName VARCHAR	description VARCHAR	foundDate DATE	status CHAR
1	Wallet	Black leather wallet with initials "J.S." engraved	2024-05-15	분실
2	Keys	Set of keys with a keychain shaped like a cat	2024-05-20	분실
3	Phone	iPhone X with cracked screen, found near the cafe	2024-05-18	분실
4	Watch	Silver wristwatch with a blue dial and leather strap	2024-05-22	수령
5	Umbrella	Red umbrella with a wooden handle, left in the lobby	2024-05-25	분실

REQ804 분실물 발견 후 30일이 지나면 상태를 '폐기'로 변경한다

```
DELIMITER $$
CREATE EVENT check_lostarticle_event
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP
DO
BEGIN
    UPDATE lostarticle
    SET status = CASE
                WHEN CURDATE() <= DATE_ADD(foundDate, INTERVAL 30 DAY) THEN '분실'
                ELSE '폐기'
              END;
END $$
DELIMITER ;
```



```
DELIMITER $$
CREATE EVENT check_lostarticle_event
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP
DO
BEGIN
    UPDATE lostarticle
    SET status = CASE
                WHEN CURDATE() <= DATE_ADD(foundDate, INTERVAL 30 DAY) THEN '분실'
                ELSE '폐기'
              END;
END $$
DELIMITER ;
```