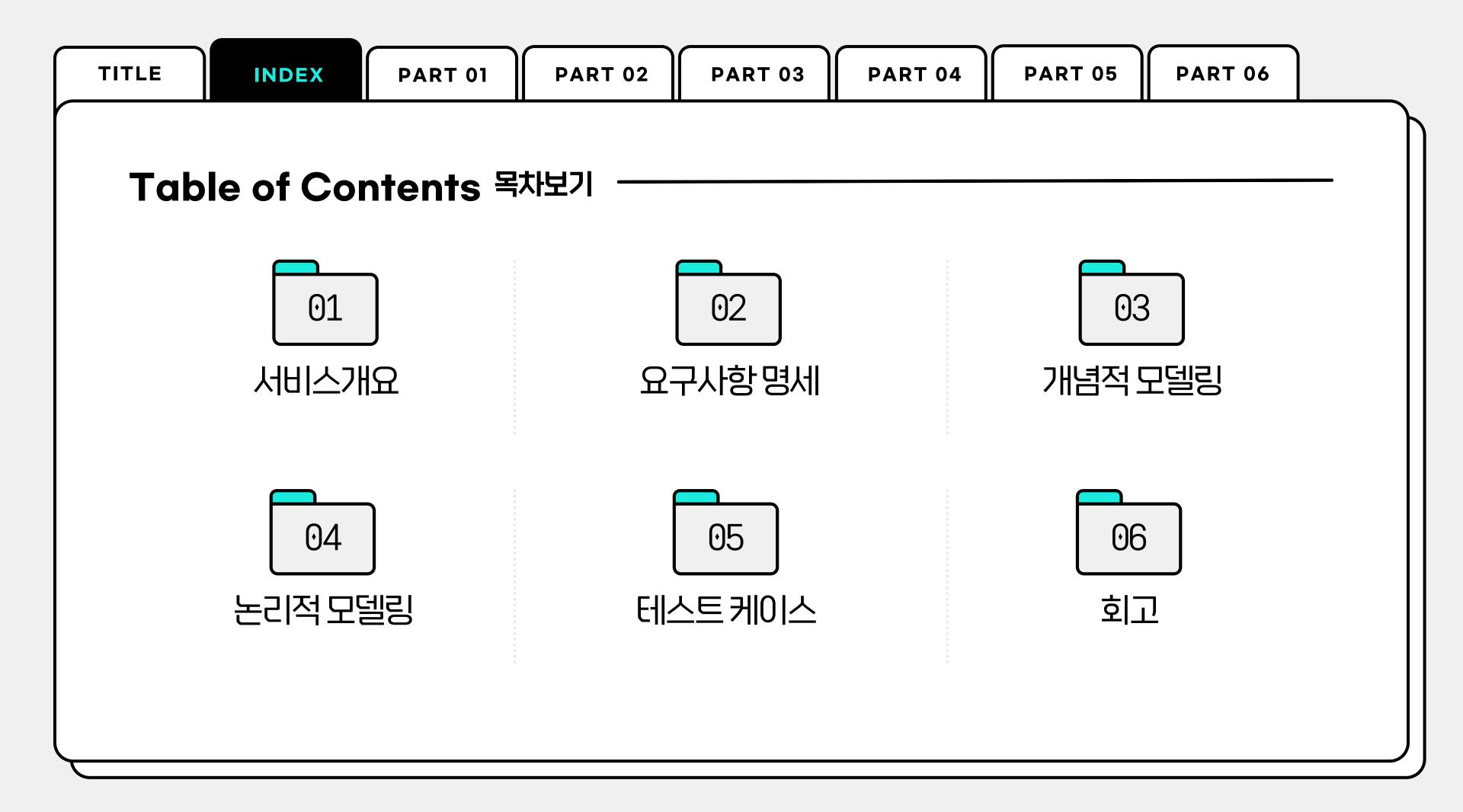
TITLE INDEX PART 01 PART 02 PART 03 PART 04 PART 05 PART 06

독서율향상을위한 도서관리시스템

BOOKIFY

[PLAYDATA] 한화시스템 BEYOND SW캠프 / YesterdayWork



Schedule 일정대시보드

5/21-5/22	5/23-5/24	5/25-5/26	5/27-5/28	5/29-5/30	5/31	6/1
주제선정						
	요구사항분석					
		개념적 모델링				
		논리적모달	벨링			
			물리적 모델링			
					테스트케이스작성	

Service Overview 서비스개요

항목	상세내용
서비스	✔ 도서 및 회원 정보를 효율적으로 관리하기 위한 도서 관리 프로그램
기획배경	✔ MZ세대의 문해력 논란 발생 ✔ 줄어들고 있는 독서율을 올려 독서 문화를 활성화
기획목적	✓ 도서의 재고 관리와 도서관 내의 다양한 서비스를 효율적으로 운영할 수 있는 기능을 제공✓ 독서 콘텐츠의 소개와 추천을 통해 사용자들의 독서 활동을 지원하고 독서 문화를 활성화
주요기능	✔ 도서 대여 시스템 ✔ 스터디룸 예약 시스템 ✔ 스터디룸 결제 시스템

Requirements Specification 요구사항명세

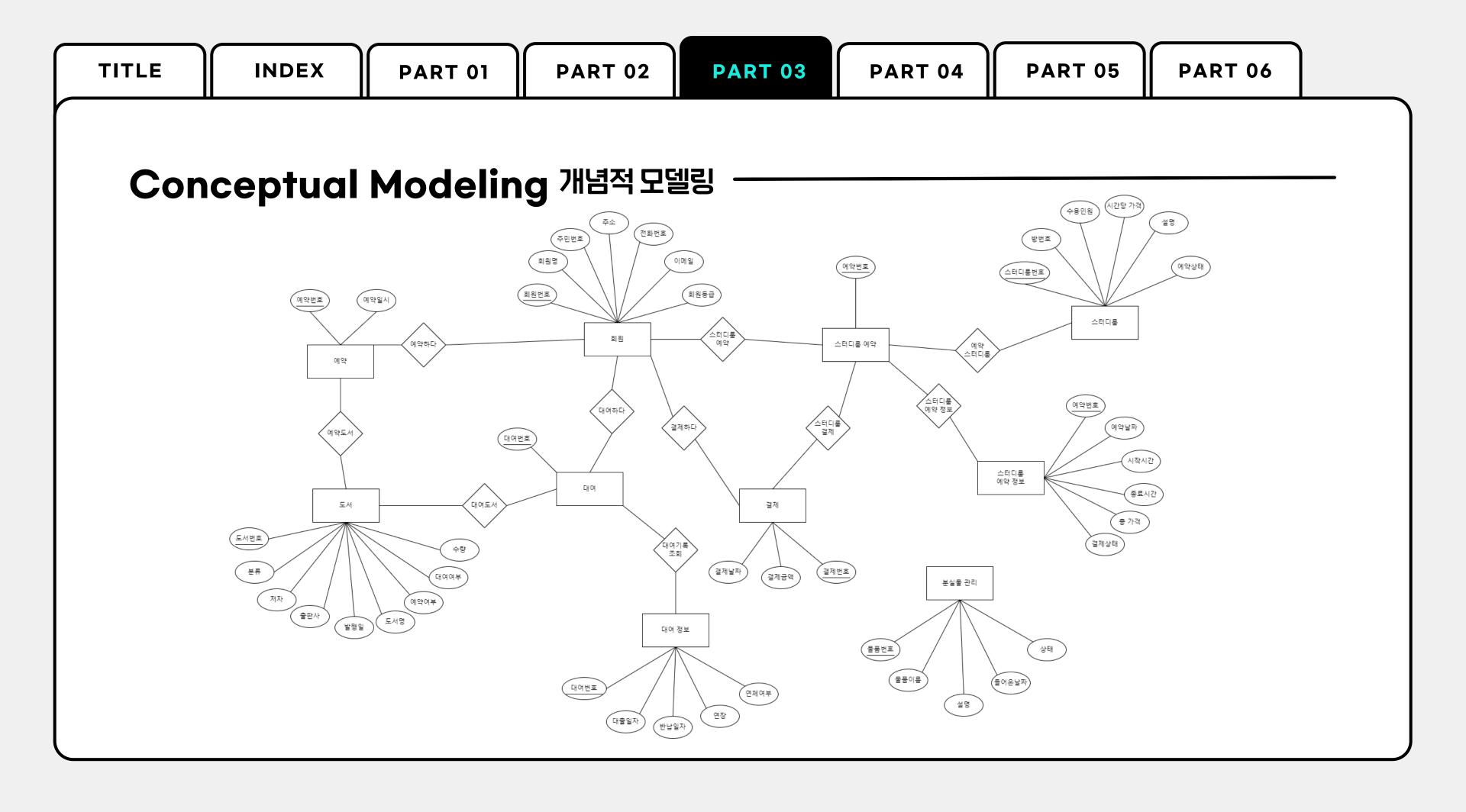
	요	구사항명			
요구사항-ID	대분류	소분류	요구사항 내용	테이블명	사용자
REQ101	회원	회원가입	회원번호, 이메일, 주민번호, 주소, 전화번호를 필수적으로 입력받아 회원에 가입한다.	회원	일반회원, 관리자
REQ102		회원 탈퇴	회원을 탈퇴한다.	회원, 대여	일반회원
REQ103		회원정보 수정	주소, 전화번호, 이름을 수정한다.	회원	일반회원, 관리지
REQ104		회원등급 반영	대출 횟수에 따라 회원 등급 반영한다.	회원, 대여정보	관리자
REQ105		회원 이메일(아이디) 검색	회원의 이메일(아이디)을 검색한다.	회원	일반회원, 관리지
REQ106		비밀번호 찾기	회원 이메일과 전화번호를 이용하여 비밀번호를 찾는다.	회원	일반회원, 관리지
REQ107		회원 등급 관리	관리자가 회원의 등급을 변경한다.	회원	관리자
REQ108		대출 횟수 확인	회원의 대출 횟수를 확인한다.	회원, 대여정보	일반회원
REQ201	대여	도서 대여	회원, 도서번호 입력시 도서를 대여한다.	대여, 대여정보, 회원, 도서	일반회원
REQ202		대여 확인	대여중인 회원 정보 조회	대여, 대야정보, 회원	관리자
REQ203		도서 연장	대출 일자를 연장한다.	대여, 대여정보	일반회원
REQ204		연장 확인	예약 여부를 확인하고, 연장 여부 확인한다.	대여, 대여정보, 도서	일반회원
REQ205		반납일 확인	도서 반납일을 확인한다.	대여, 대여정보	일반회원
REQ206		도서 반납	대여 테이블에서 데이터를 삭제한다.	대여, 대여정보, 도서	일반회원
REQ207		연체여부 확인	회원의 연체여부를 확인한다.	회원, 대여, 대여정보	관리자
REQ208		연체일 확인	회원의 연체일을 확인한다.	회원	일반회원
REQ301	예약	도서 예약	도서 정보 조회 후, 도서를 예약한다.	예약, 도서, 도서정보	일반회원
REQ302		예약 여부 변경	도서 예약시 Y로 변경하고, 예약 회원이 더이상 없다면 N으로 변경한다.	예약, 도서	관리자
REQ303		예약 대출 안내	도서 반납시, 예약자에게 대출을 안내한다.	예약, 도서	관리자
REQ304		예약 취소	도서 예약을 취소한다.	예약, 도서	일반회원
REQ305		예약 기간 확인	예약된 도서의 예약 기간을 확인한다.	예약, 도서	일반회원

Requirements Specification 요구사항명세

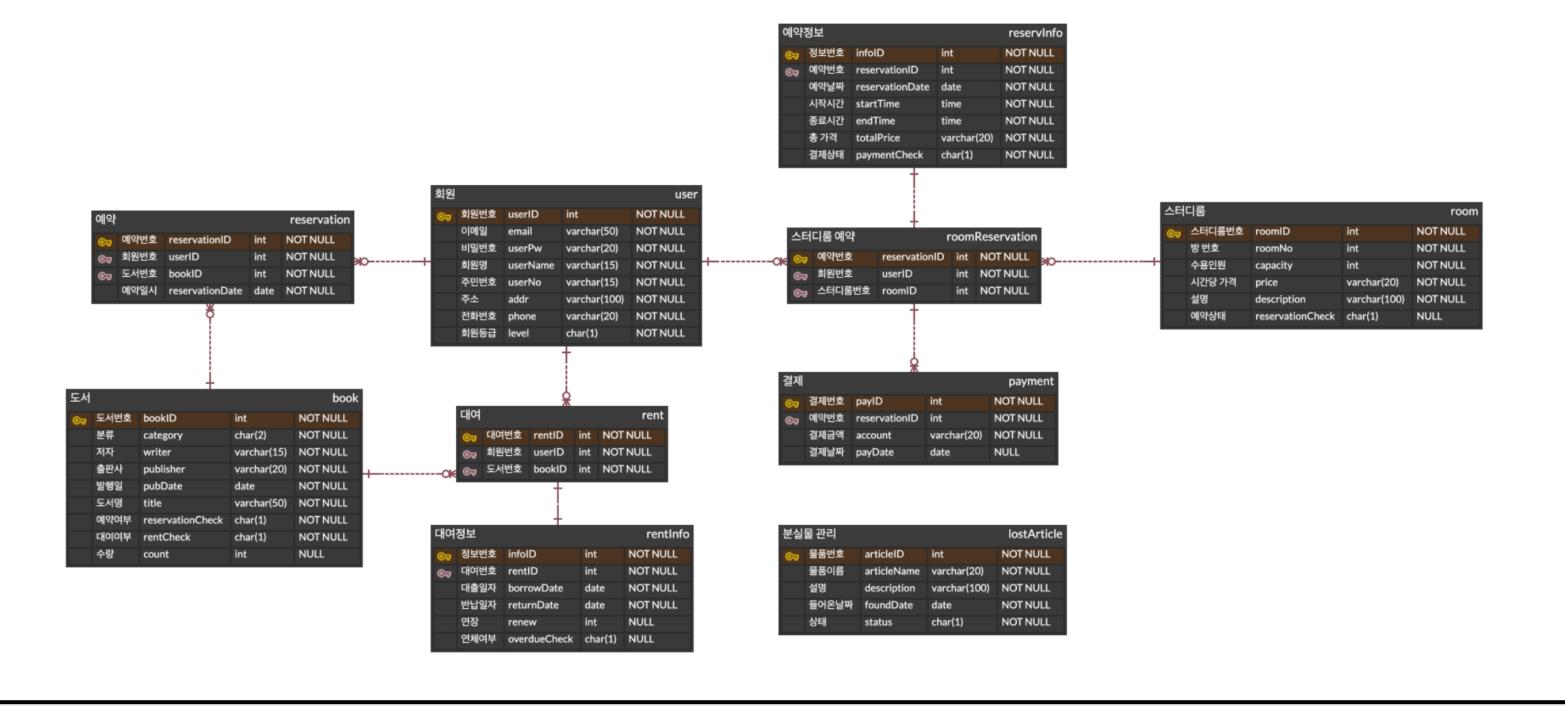
REQ401	도서	도서 찾기	도서명, 저자, 출판사, 발행일, 분류기호, 대출 가능 여부, 예약 여부를 확인한다.	도서, 도서정보	일반회원, 관리지
REQ402		도서 추가	도서명, 저자, 출판사, 발행일, 분류기호, 분류기호로 도서를 추가한다.	도서정보	관리자
REQ403		도서 재고 확인	도서정보로 동일 도서 확인 후 갯수를 확인한다.	도서정보	관리자
REQ404		도서 대출 관리	대출 여부를 확인하거나 반납/대출 누락 도서를 확인한다.	대여, 도서	관리자
REQ405		도서 정보 수정	도서 정보를 수정한다.	도서정보	관리자
REQ406		도서 삭제	파손되거나 분실된 도서인 경우 도서정보를 삭제한다.	도서정보	관리자
REQ501	결제	누적 결제 금액 조회	회원은 누적 결제 금액을 조회 할 수 있어야 한다.	회원, 결제	일반회원, 관리지
REQ502		결제 이력 조회	회원은 지난 결제 이력을 조회 할 수 있어야 한다.	회원, 결제, 스터디룸 예약	일반회원, 관리지
REQ503		결제 상태 확인	결제의 완료, 취소, 환불 여부를 확인	결제, 스터디룸 예약, 예약정보	일반회원, 관리지
REQ504		결제 이력 추가	스터디룸을 예약하면 결제 이력이 추가될 수 있어야 한다.	스터디룸예약, 스터디룸, 결제	관리자
REQ505		결제하기	회원이 결제를 하면 결제일과 결제여부를 변경한다.	결제, 예약정보	관리자
REQ601	스터디룸	스터디룸 추가	새로운 스터디룸(이름, 크기, 수용가능인원, 방번호)을 추가할 수 있다	스터디룸	관리자
REQ602		스터디룸 조회	데이터베이스에 저장된 모든 스터디룸을 조회할 수 있다	스터디룸	일반회원, 관리지
REQ603		스터디룸 수정	기존의 스터디룸 정보를 수정할 수 있다	스터디룸	관리자
REQ604		스터디룸 삭제	특정 스터디룸을 삭제할 수 있다	스터디룸	관리자
REQ701	스터디룸 예약	예약 생성	회원은 당일 날짜에 스터디룸을 예약할 수 있다	스터디룸 예약, 회원,결제	일반회원
REQ702		예약 조회	데이터베이스에 저장된 모든 예약정보를 조회할 수 있다	스터디룸 예약, 회원	일반회원, 관리자
REQ703		예약 정보 생성	회원이 스터디룸을 예약하면 예약 정보를 생성한다	스터디룸 예약, 예약정보	
REQ704		예약 수정	기존의 예약 내역을 수정할 수 있다(인원이나 예약시간변경 다른 스터디룸 등등)	스터디룸 예약, 예약정보, 회원, 결제	일반회원
REQ706		예약 취소	회원은 예약한 스터디룸을 취소할 수 있다	스터디룸 예약, 회원, 결제	일반회원
REQ707		예약 정보 삭제	회원이 예약한 스터디룸을 취소하면 결제와 예약 정보 테이블에서도 삭제되어야 한다	스터디룸 예약, 예약정보, 결제	일반회원
REQ709		예약 상태 확인	스터디룸이 예약이 되었는지 예약이 가능한지 예약 상태 여부를 확인할 수 있다	스터디룸 예약	일반회원, 관리자
REQ710		예약 정보 확인	스터디룸 예약이 되어있는지 정보 확인할 수 있다.	스터디룸	관리자
REQ711		예약 상태 변경 - 1	스터디룸을 예약하면 스터디룸의 상태를 변경한다	스터디룸 예약, 스터디룸	관리자
REQ712		예약 상태 변경 - 2	스터디룸을 예약이 종료되면 스터디룸의 상태를 변경한다	스터디룸 예약, 스터디룸	관리자

Requirements Specification 요구사항명세

REQ801	분실물 관리	분실물 조회	분실물 종류로 분실물의 정보를 찾을 수 있다.	분실물 관리	일반회원, 관리자
REQ802		분실물 추가	분실물 정보를 입력하여 분실물을 추가한다.	분실물 관리	관리자
REQ803		수령 분실물 정보 변경	분실물을 찾아갔을 경우, 분실물 상태를 변경한다.	분실물 관리	관리자
REQ804		폐기 분실물 정보 변경	분실물 발견 후 30일이 지나면 상태를 '폐기'로 변경한다	분실물 관리	관리자



Logical Modeling 논리적모델링



Test Case 테스트케이스1 ——

```
DELIMITER $$
CREATE TRIGGER add_rental_record
AFTER INSERT ON rent
FOR EACH ROW
BEGIN
    INSERT INTO rentInfo (rentID, borrowDate, returnDate)
    VALUES (NEW.rentID, NOW(), DATE_ADD(NOW(), INTERVAL 7 DAY));
END$$
DELIMITER;
```

테이블	내용
대여정보	도서 대여시 대여정보 테이블에 자동으로 데이터를 추가할 수 있다

Test Case 테스트케이스1 -



Test Case 테스트케이스2 ——

```
SET GLOBAL event_scheduler = ON; -- 이벤트 자동실행은 위한 변수설정

SHOW VARIABLES WHERE variable_name = 'event_scheduler'; -- 변수설정 확인

DELIMITER $$
CREATE EVENT check_overdue_event
ON SCHEDULE EVERY 1 DAY
STARTS CURRENT_TIMESTAMP
DO
BEGIN
    UPDATE rentinfo
    SET overdueCheck = case when returnDate >= CURDATE() then 'N' ELSE 'Y' END;
END $$
DELIMITER;
```

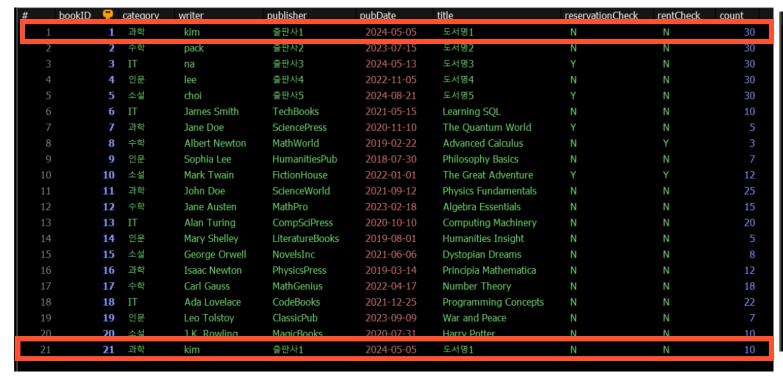
테이블	내용
대여정보	도서 반납일이 지나면 대여 정보 의 <u>연체여부를 'Y'</u> 로 변경할 수 있다

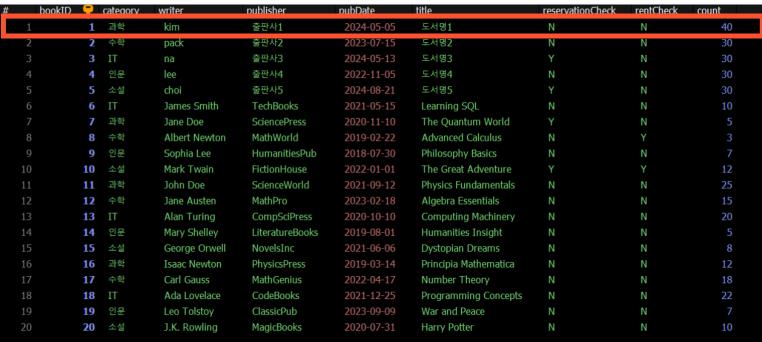
Test Case 테스트케이스3 —

```
DELIMITER $$
CREATE PROCEDURE UpdateSameBookCounts()
BEGIN
    -- 카운트를 합산
   UPDATE book t1
   JOIN (
       SELECT writer, title, SUM(count) AS total_count
       FROM book
       GROUP BY writer, title
    ) t2 ON t1.writer = t2.writer AND t1.title = t2.title
    SET t1.count = t2.total_count;
    -- 중복 제거
    DELETE FROM book
   WHERE (writer, title, bookID) NOT IN (
       SELECT writer, title, MIN(bookID)
       FROM book
       GROUP BY writer, title
   );
END$$
DELIMITER;
```

테이블	내용
도서	<u>동일한</u> 제목과 작가명을 가진 도서 이면 수량을 합친다

Test Case 테스트케이스3





Test Case 테스트케이스4 -

```
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);
```

테이블	내용
결제	스터디룸 예약 을 하면 결제 테이블에 데이터를 추가할 수 있다

Test Case 테스트케이스4 ———

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

Test Case 테스트케이스5

```
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);
```

테이블	내용
결제	회원이 결제 를 하면 결제정보 테이블의 <u>결제여부를 'Y', 결제일자를 오늘날짜</u> 로 변경한다

Test Case 테스트케이스5

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

Test Case 테스트케이스6 ——

```
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);
```

테이블	내용
예약정보	스터디룸 을 예약하면 스터디룸예약정보 데이터를 생성한다

Test Case 테스트케이스6 ——

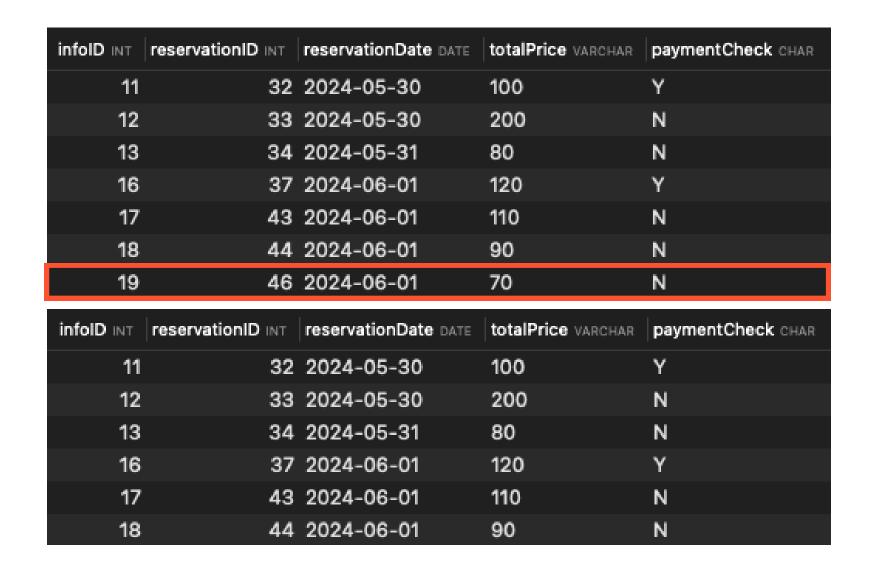


infolD INT	userID INT re	oomID INT reservati	ionDate DATE totalPri	ice varchar paymentChe	ck char
11	1	1 2024-0	5-30 100	Y	
12	2	2 2024-0	5-30 200	N	
13	3	3 2024-0	5-31 80	N	
16	4	4 2024-0	6-01 120	Y	
17	5	6 2024-0	6-01 110	N	

Test Case 테스트케이스7 ———

테이블	내용
스터디룸 예약정보	회원이 예약한 스터디룸을 취소하면 스터디룸 <u>예약 상태를 'N'</u> 으로 변경하고 결제, 예약정보, 스터디룸예약 테이블에서도 <u>삭제</u> 되어야 한다

Test Case 테스트케이스7 —

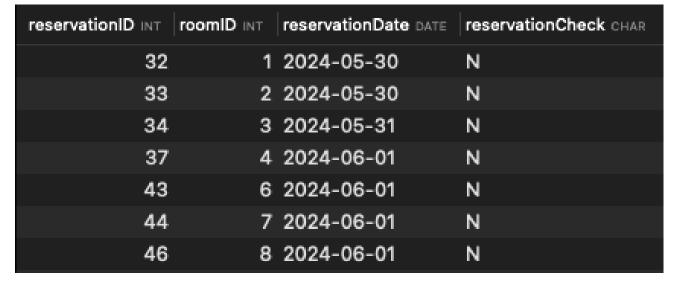


Test Case 테스트케이스8 -

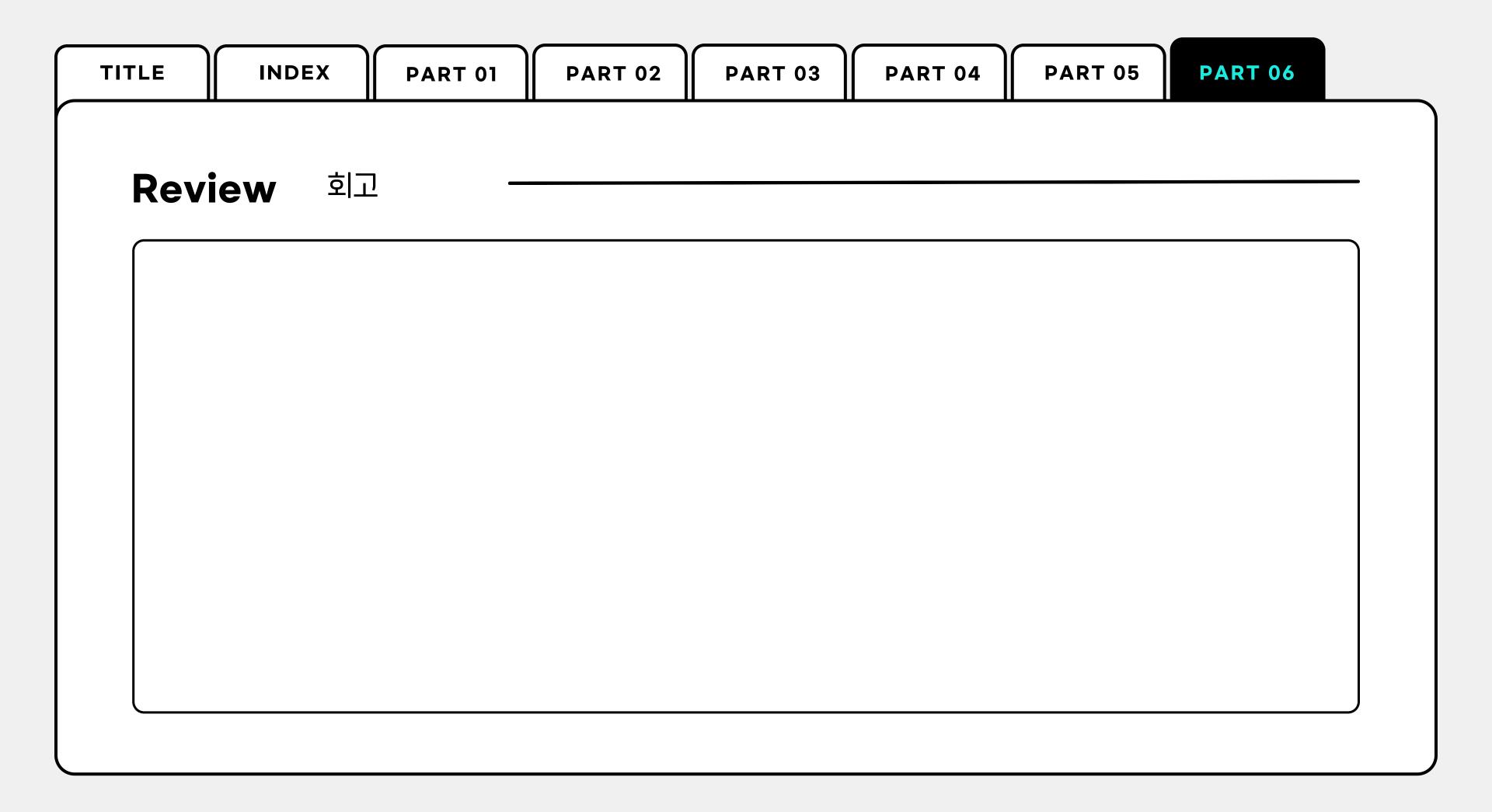
```
CREATE OR REPLACE PROCEDURE cur_room()
BEGIN
    DECLARE newDate DATE;
    DECLARE newRoomId INT;
    DECLARE endOfRow BOOLEAN DEFAULT FALSE;
    DECLARE roomCursor CURSOR FOR
      (SELECT rr.roomId, r.reservationDate
        FROM reservinfo r
        INNER JOIN roomreservation rr ON r.reservationID = 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</pre>
            UPDATE room r SET reservationCheck = 'Y' WHERE r.roomID = newRoomId;
        END IF:
    END LOOP cursor_loop;
    CLOSE roomCursor;
END $$
DELIMITER;
CALL cur_room();
```

터이블	내용
스터디룸	스터디룸 예약 종료시 예약 날짜가 지난 것 확인하고, 스터디룸 예약 상태 변경하기

Test Case 테스트케이스8 -



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



감사합니다

BOOKIFY

[PLAYDATA] 한화시스템 BEYOND SW캠프 / YesterdayWork