

온라인 서점 Database

2021 - 2 데이터베이스 설계 및 활용 기말 과제

6조 김현성 박정현 정민우 조인영

목차

1. 과제 계획

2. 개념 설계 및 ERD

3. SQL SERVER을 통한 DATABASE 구축

4. ASP와 HTML을 활용한 웹 구축

과제 계획

<초안>

데이터베이스 설계 및 활용 [가] 6조

◎ 팀원 구성: 김현성, 박정현, 정민우, 조인영

[역할 분담]

- 1). SQL 설계 팀: 정민우, 조인영
- 2). E-R 모델링 및 ASP 구축 팀: 박정현, 김현성
- 3). PPT(보고서) 제작 팀: 김현성, 조인영
- 4). 발표 영상 녹화 팀: 정민우, 박정현

◎ 선정 주제: 가상의 온라인 서점

◎ 과제 목표: 온라인 서점 내 구비되어 있는 도서 정보를 데이터베이스화 하여 분류하고 고객이 원하는 도서를 조회할 수 있는 정보시스템을 구축한다.

◎ 과제 범위: 온라인 서점이라는 테마 안에서 도서 관련 정보들에 대한 테이블을 중심으로 짜인 데이터베이스를 통해 고객이 원하는 도서를 구매 및 유통하는 프로세스의 전산적 처리 작업을 수행할 수 있도록 설계한다.

가상의 온라인 서점을
주제로 선정 한 뒤
목표와 범위를 설정했고
과제의 목표와
세부적인 과제 수행 범위를 구체화
할 필요가 있다는 피드백을 받았음.

6조 제안서 제출입니다

정민우(20###30) 10월 31일 오후 10:23

- 가상의 온라인 서점을 대상으로 데이터베이스 시스템 구축 과제 제안서를 작성함 - 과제의 목표와 세부적인 과제 수행 범위 및 일정계획을 보다 구체화할 필요가 있음(일정표 상에는 SQL 설계, E-R, ASP 설계만 제시되어 있음) - 기말과제 수행 가이드라인에 제시된 내용을 모두 충족하도록 과제를 수행하기 바람 - 조원간 협력을 통해 제시한 일정계획에 따라 충실하게 과제를 수행하여 결과를 제시하기 바람

김동수 11월 5일 오전 10:43

임력된 점수(배점 0)는 무시하시기 바랍니다.

김동수 11월 5일 오전 11:14

과제 계획

<수정본>

데이터베이스 설계 및 활용 [가] 6조

◎ 팀원 구성: 김현성, 박정현, 정민우, 조인영

[역할 분담]

- 1). SQL 설계 팀: 정민우, 조인영
- 2). E-R 모델링 및 ASP 구축 팀: 박정현, 김현성
- 3). PPT(보고서) 제작 팀: 김현성, 조인영
- 4). 발표 영상 녹화 팀: 정민우, 박정현

◎ 선정 주제: 가상의 온라인 서점

◎ 과제 목표: 온라인 서점 내의 고객과 도서에 관련된 정보들(쿠폰, 주문, 결제, 작가, 출판사, 장바구니 등)을 데이터베이스화 하여 분류하고 정보시스템을 구축한다.

◎ 과제 범위: 온라인 서점이라는 테마 안에서 도서 관련 정보들에 대한 테이블을 중심으로 짜인 데이터베이스를 통해 고객이 도서, 작가, 리뷰 등을 웹을 통해 조회하고, 자신의 고객정보를 고객 목록에 추가하는(회원가입) 기능까지 구현한다.

범위와 목표를 보다 구체화 하였다.

어떤 정보를 다룰 것이고,
SQL SERVER 와 ASP 를 통해
고객이 실제로
웹에서 조회 할 수 있는
범위를 명시하였다.

개념설계 및 ERD

테이블과 그 사이의 관계를 중심으로 한 개념설계 내용

개념설계

ERD 초안

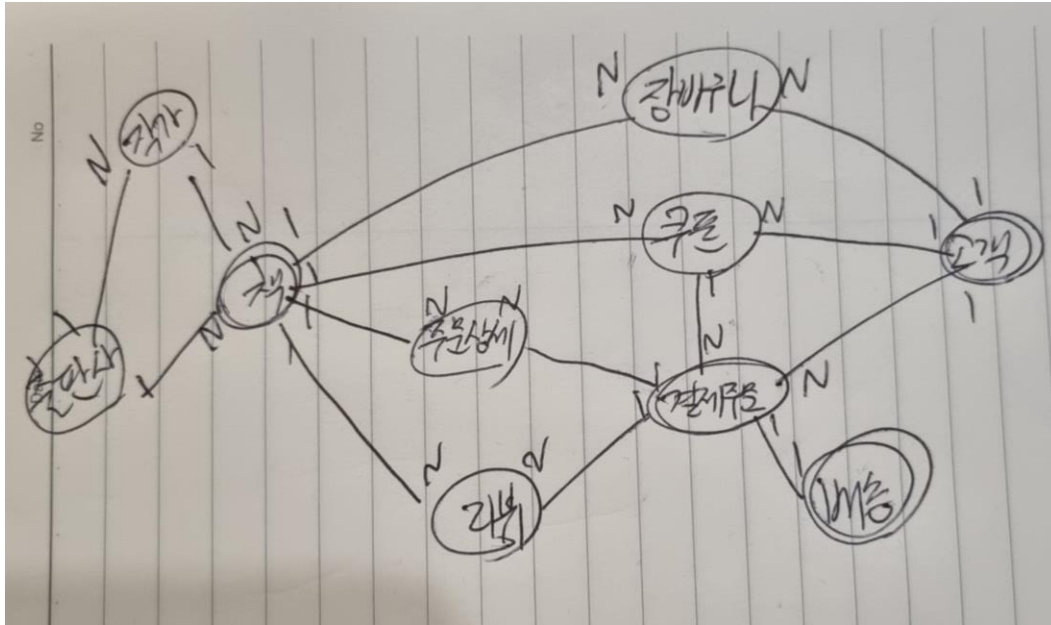


TABLE 초안

고객: 고객ID(PK), 이름, 주소, 우편번호, 전화번호, 메일, 추천인(FK)↵

책: 책ID(PK), 책이름, 출간년도, 작가ID(FK), 가격, 재고, 페이지 수, ebook, 출판사(FK)↵

(WEAK) 주문: 주문ID(PK), 고객ID(FK), 주문일, 결제수단↵

(WEAK) 결제: [결제ID, 주문ID(FK)] (PK), 쿠폰ID(FK), 총 금액↵

작가: 작가ID (PK): surrogate key 작가이름, 이메일 소속출판사(FK), 전화번호↵

출판사: 출판사이름(PK), 전화번호, 주소, 우편번호, 웹사이트↵

(WEAK)이지만 strong 취급 // 장바구니: 장바구니번호(PK), 고객ID(FK), 책ID(FK), 구매수량↵

(WEAK) 주문상세: [주문ID(FK), 책ID(FK)] (PK), 상품별구매수량↵

(WEAK) 리뷰: [주문ID(FK), 책ID(FK)](PK), 평점, 리뷰내용, 등록일↵

(WEAK) 배송: [주문ID(FK), 배송ID](PK), 우편번호, 배송지, 수취자명, 수취자번호↵

쿠폰: 쿠폰번호(PK), 쿠폰명, 할인률, 할인금액 ↵

(WEAK) 쿠폰리스트: [쿠폰ID (FK) 고객ID (FK)] (PK)↵

도서와 고객을 중심으로 들어가야 하는 테이블,내용과
각 테이블 사이의 관계를 스케치를 통해 초안을 생각해 본 내용이다.

PUBLISHER 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	PublisherName (출판사명)	Varchar(30)	UNIQUE, NOT NULL
	OfficeNumber (사무실번호)	Char(25)	NOT NULL
	PublisherAddress (주소)	Varchar(100)	NOT NULL
	PubZIP (우편번호)	Char(30)	NOT NULL
	Website (홈페이지)	Varchar(100)	NOT NULL

PUBLISHER 테이블은 단독으로 존재 할 수 있는 STRONG ENTITY이다.

UNIQUE한 (중복불가능한) Varchar(30) 형식의 PublisherName을 PK로 가진다.

모든 항목을 필수값(NOT NULL)로 설정하였고, 길이가 정해진 번호,우편번호를 제외하고 VarChar 형식을 지정했다.

AUTHOR 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	AuthorID (작가 ID)	Int	IDENTITY(1,1), NOT NULL
	AuthorName (작가 이름)	Char(25)	NOT NULL
	AuthorEmail (작가 Email)	Varchar(100)	NOT NULL
	AuthorPhone (작가 휴대폰번호)	Char(25)	NOT NULL
FK	PublisherName (소속출판사)	Varchar(30)	NULL

AUTHOR 테이블은 단독으로 존재 할 수 있는 STRONG ENTITY이다.

작가 이름만으로 PK를 설정하기엔 동명이인이 있을 수 있다고 생각하였고,
따라서 일련번호(IDENTITY(1,1)) 형식의 AuthorID를 PK로 가진다.

무소속인 작가가 있을 수 있기 때문에 출판사 정보는 NULL을 허용하였다.

BOOK 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	BookID (책 ID)	Int	IDENTITY(1,1) NOT NULL
	BookName (책 이름)	Varchar(50)	NOT NULL
	PubYear (출간 연도)	Date	NOT NULL
FK	AuthorID (작가 ID)	Int	NOT NULL
	Price (가격)	Int	NOT NULL
	Inventory (재고)	Int	NOT NULL
	Pages (페이지 수)	Int	NOT NULL
	Ebook (Ebook지원 여부)	Varchar(30)	NOT NULL
FK	PublisherName (출판사)	Varchar(30)	NOT NULL

BOOK 테이블은 단독으로 존재 할 수 있는 STRONG ENTITY이다.

책 이름이 같은 경우가 있을 수 있기 때문에, 책 이름을 PK로 정하지 않았고,
일련번호(IDENTITY(1,1))형식의 BookID를 PK로 정하였다.

CUSTOMER 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	CustomerID (회원 ID)	Varchar(100)	UNIQUE, NOT NULL
	CustomerPW (회원 암호)	Varchar(100)	NOT NULL
	CustomerName (회원 이름)	Char(30)	NOT NULL
	CsAddress (회원 주소)	Char(100)	NOT NULL
	CsZIP (우편번호)	Char(20)	NOT NULL
	CustomerPhone (전화번호)	Char(20)	NOT NULL
	CustomerEmail (회원Email)	Char(100)	NOT NULL
	ReferredBy (추천인)	Varchar(100)	NULL

CUSTOMER 테이블은 단독으로 존재 할 수 있는 STRONG ENTITY이다.

중복 불가능한 (UNIQUE) 문자열인 CustomerID를 PK로 가진다

회원가입 과정에서 추천인을 입력하지 않을 수 있기 때문에 NULL값을 허용하였다.

COUPON & COUPONLIST(WEAK) 테이블

PK/ FK	열 이름	데이터 형식	특이사항
PK	CouponID (쿠폰ID)	Int	IDENTITY(1,1) NOT NULL
	CouponName (쿠폰 이름)	Varchar (50)	NOT NULL
	DiscountRate (할인률)	Char(10)	NOT NULL
	Validity (유효기간)	Date	NULL

PK/ FK	열 이름	데이터 형식	특이사항
PK	CouponID (주문ID)	Int	UNIQUE, NOT NULL
PK	CustomerID (회원ID)	Varchar(100)	UNIQUE, NOT NULL

COUPON은 단독으로 존재 할 수 있는 STRONG ENTITY이다.

하지만, 개념상 쿠폰 – 회원은 다대다 관계이므로, 중간에 이를 연결해주는 Intersection Table이 필요하다.

COUPONLIST 라는 Intersection Table을 추가로 구성했다.

COUPON 테이블은 일련번호인 CouponID를 PK로 가진다.

COUPONLIST 테이블은 부모 테이블로부터 두 개의 PK를 받아와 COMPOSITE KEY를 PK로 가진다.

유효기간이 없는 쿠폰이 있을 수 있기 때문에 Validity에는 NULL값을 허용하였다.

CART 테이블(WEAK)

PK/FK	열 이름	데이터 형식	특이사항
PK	CartID (카트ID)	Int	IDENTITY(1,1), NOT NULL
FK	CustomerID (회원ID)	Varchar (100)	NOT NULL
FK	BookID (책 ID)	Int	NOT NULL
	Quantity (주문수량)	Int	NOT NULL

CART TABLE은 회원과 책 없이 존재 할 수 없는 ID-Dependent인 Weak Table이다.

부모 테이블로부터 PK를 두개 빌려오고, 일련번호 형식의 카트ID를 추가로 부여하여 COMPOSITE KEY를 PK로 가진다.

모든 항목을 필수라고 생각하여 NOT NULL로 설정하였다.

ORDERS & ORDERINFO(WEAK) 테이블

PK/ FK	열 이름	데이터 형식	특이사항
PK	OrderID (주문 ID)	Int	IDENTITY(1,1) NOT NULL
FK	CustomerID (회원 ID)	Varchar (100)	NOT NULL
	OrderDate (주문 날짜)	Date	NOT NULL

PK/ FK	열 이름	데이터 형식	특이사항
PK	OrderID(주문ID)	Int	NOT NULL
PK	BookID (책ID)	Int	NOT NULL
	QPerProduct (상품별 수량)	Int	NOT NULL

ORDERS는 회원과 책이 존재해야 존재 할 수 있는 ID-Dependent 인 WEAK Entity이다.
단, 개념상 책-주문 은 다대다 관계이므로, 중간에 이를 연결해주는 Intersection Table이 필요하다.
ORDERINFO 라는 Intersection Table을 추가하고, 여기에 상품별 수량을 추가하여 **Associative entity**로 구성했다.

ORDERS 테이블은 부모 테이블로부터 PK를 받아오고, 주문ID를 추가로 PK로 부여하여 COMPOSITE KEY를 PK로 가진다.
ORDERINFO 테이블은 부모 테이블로부터 두 개의 PK를 받아와 COMPOSITE KEY를 PK로 가진다.

두 PK 다 중복되면 안되는 값이기 때문에 **UNIQUE**를 걸어주었고
결제 과정에서 쿠폰이 사용되지 않았을 수 있기 때문에 NULL값을 허용하였다.

PAYMENT(WEAK) 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	PaymentID (결제ID)	Int	UNIQUE, NOT NULL
PK	OrderID (주문ID)	Int	UNIQUE, NOT NULL
FK	CouponID (쿠폰ID)	Int	NULL
	PMethod (결제방법)	Varchar(20)	NOT NULL
	TotalPrice (결제금액)	Int	NOT NULL

결제는 주문이 존재해야 존재 할 수 있기 때문에 ID-Dependent 한 WEAK ENTITY이다.

부모 Table에서 PK를 빌려오고, 결제 ID를 추가로 PK로 부여하여 COMPOSITE KEY를 PK로 가진다.

두 PK 다 중복되면 안되는 값이기 때문에 **UNIQUE**를 걸어주었고
결제 과정에서 쿠폰이 사용되지 않았을 수 있기 때문에 NULL값을 허용하였다.

DELIVERY(WEAK) 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	OrderID (결제ID)	Int	UNIQUE, NOT NULL
PK	DeliveryID (배달ID)	Int	IDENTITY(1,1), UNIQUE, NOT NULL
	DZip (우편번호)	Varchar(30)	NOT NULL
	DAddress (배달주소)	Varchar(100)	NOT NULL
	RecieverName (수취자명)	Char(20)	NOT NULL
	RecieverNumber (수취자번호)	Char(20)	NOT NULL

배달은 주문 없이는 존재 할 수가 없다 = WEAK ENTITY

부모 Table에서 PK를 빌려오고, 배달ID를 추가로 PK로 부여하여 COMPOSITE KEY를 PK로 가지게 되었다.

두 PK 다 중복되면 안되는 값이기 때문에 **UNIQUE**를 걸어주었고
배달을 받는데 모든 항목들이 필수적이라고 판단하여 모든 필드에 **NOT NULL**을 설정하였다.

REVIEW(WEAK) 테이블

PK/FK	열 이름	데이터 형식	특이사항
PK	OrderID (주문ID)	Int	NOT NULL
PK	BookID (책 ID)	Int	NOT NULL
	Grade (평점)	Numeric(2,1)	NULL
	ReviewDetail (상세 리뷰)	Varchar(100)	NULL
	EnrollDate (등록일)	Date	NOT NULL

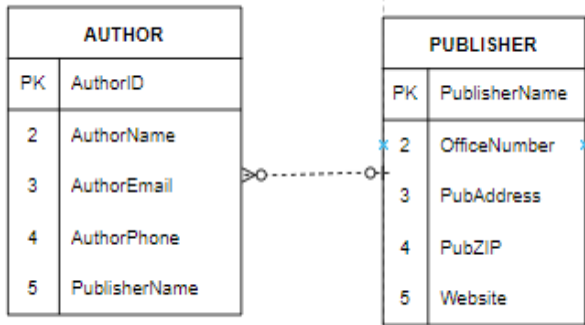
리뷰는 주문과 해당 책이 없으면 존재 할 수 없다. = ID-Dependent 한 WEAK ENTITY

두 부모 Table에서 PK를 빌려와서 COMPOSITE KEY를 PK로 가지는 TABLE이 되었다.

10점 만점의 소수점 첫째 자리를 포함한 숫자 형식으로 표현되는 Grade에 Numeric(2,1) 형식을 주었고,

평점과 상세 리뷰는 필수 사항이 아니기 때문에 NULL 값을 허용했다.

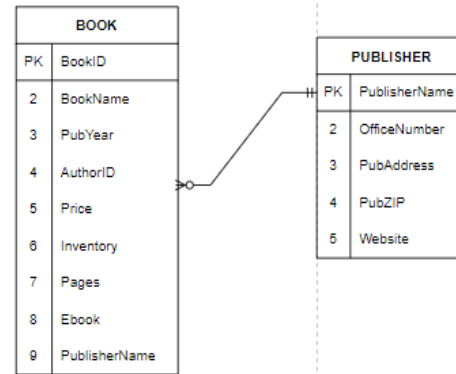
개념설계 -관계



AUTHOR-PUBLISHER

AUTHOR 테이블의
PublisherName (FK)이
PUBLISHER 테이블의
PublisherName (PK)를 참조한다.

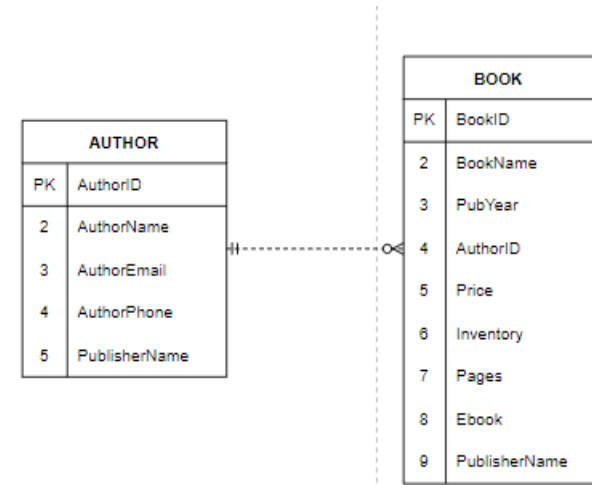
출판사엔 여러 명의 작가가 있을 수도 있고,
한 명도 없을 수도 있기 때문에
Optional-Many 이고,
작가는 출판사가 없거나
최대 하나의 출판사에 소속될 수 있기 때문에
Optional-One 이다.



BOOK-PUBLISHER

BOOK 테이블의
PublisherName (FK)이
PUBLISHER 테이블의
PublisherName (PK)를 참조한다.

한 권의 책은 한 개의 출판사에서 출판 되기
때문에
Mandatory-One 이고,
하나의 출판사는 여러 권의 책을 출판할 수도
있고,
한 권도 출판하지 않을수도 있기 때문에
Optional-Many 이다.

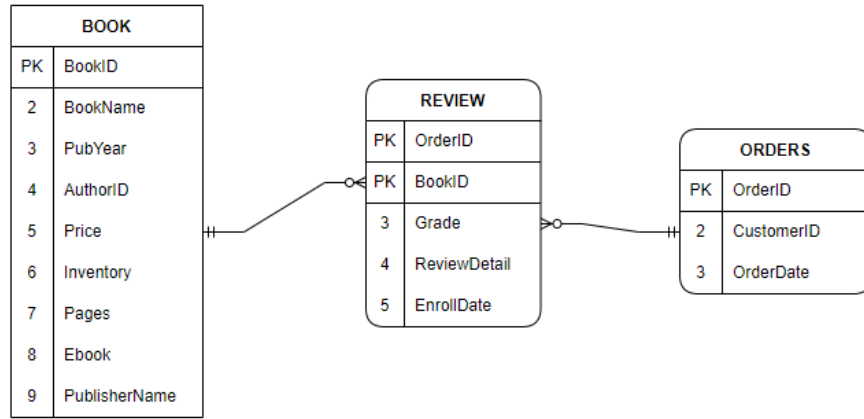


BOOK-AUTHOR

BOOK 테이블의
AuthorID (FK) 가
AUTHOR 테이블의
AuthorID (PK)를 참조한다.

한 작가는 책을 쓰지 않았을 수도 있고,
여러 권의 책을 쓸 수도 있기 때문에
Optional-Many 이고,
한 권의 책은 한 작가에 의해서 쓰여지기 때문에
(공동저자 고려하지 않았음)
Mandatory-One 이다.

개념설계 -관계



BOOK - REVIEW

REVIEW 테이블의
BookID(FK)가
BOOK 테이블의
BookID(PK)를 참조한다.

한 권의 책에는 리뷰가 있을 수도 있고,
여러 개가 있을 수도 있기 때문에

Optional-many이다.

한 개의 리뷰는 책 없이 존재할 수 없고,
무조건 하나의 책에 대해서만 존재하기 때문에
Mandatory-One이다.

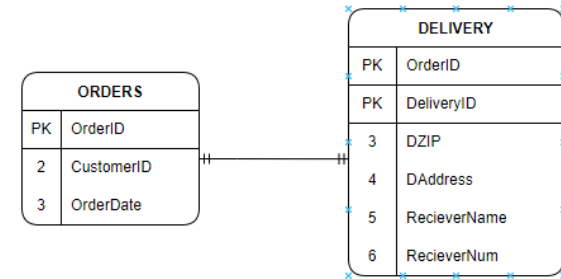
REVIEW - ORDERS

REVIEW 테이블의
OrderID (FK)이
ORDERS 테이블의
OrderID (PK)를 참조한다.

하나의 주문 안에 여러 권의 책이 있을 수 있고,
따라서 리뷰는 없을 수도 있고,
여러 개 존재할 수 있기 때문에

Optional-many이다.

리뷰는 주문이 존재해야 존재 할 수 있기 때문에
Mandatory-one이다.



ORDERS - DELIVERY

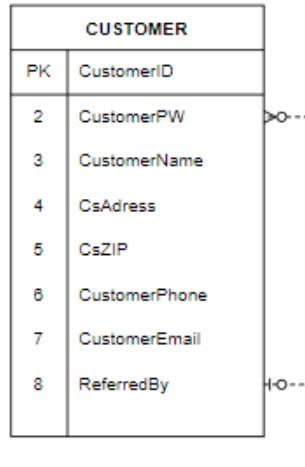
DELIVERY 테이블의
OrderID (FK)가
ORDERS 테이블의
OrderID (PK)를 참조한다.

한 개의 주문이 이루어지면
최대 1개의 배송 정보가 생기고,
배송은 주문 없이 존재 할 수 없기 때문에

양쪽 다

Mandatory-One이다.

개념설계 - 관계



CUSTOMER – CUSTOMER

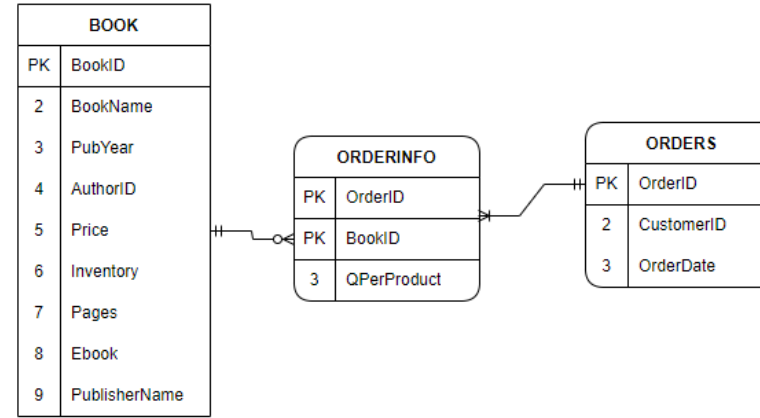
재귀적 관계이다.

한 고객은 추천을 하지 않거나,
최대 한 명을 추천 할 수 있기 때문에

Optional-One이다.

한 고객은 아무에게도 추천을 받지 못하거나
여러 명에게 추천을 받을 수 있기 때문에

Optional-Many이다.



ORDERS – ORDERINFO

ORDERINFO 테이블의
OrderID (FK)이
ORDERS 테이블의
OrderID (PK)를 참조한다.

ORDERINFO는 하나의 주문ID 안에서
주문한 책들의 개별 수량을 보여주는 테이블이다.

따라서 주문은 최소 한 개 최대 여러 개의

ORDERINFO를 가질 수 있기 때문에

Mandatory-Many이고

하나의 ORDERINFO는

하나의 주문에만 해당하기 때문에

Mandatory-One이다.

ORDERINFO – BOOK

ORDERINFO 테이블의
BookID (FK)이
BOOK 테이블의
BookID (PK)를 참조한다.

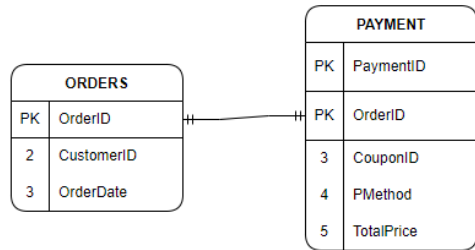
한 개의 주문 상세정보는
책 한권에 대해서만 존재하기 때문에

Mandatory-One이다.

한 권의 책에 대해 주문 상세정보가 없을 수도
있고 여러 개가 존재 할 수도 있기 때문에

Optional-many이다.

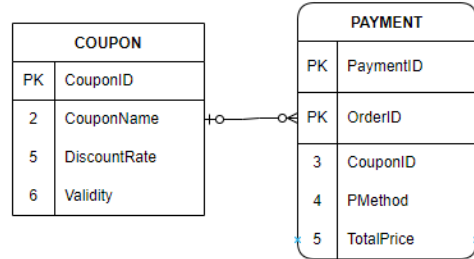
개념설계 -관계



ORDERS – PAYMENT

PAYMENT 테이블의
OrderID (FK)이
ORDERS 테이블의
OrderID (PK)를 참조한다.

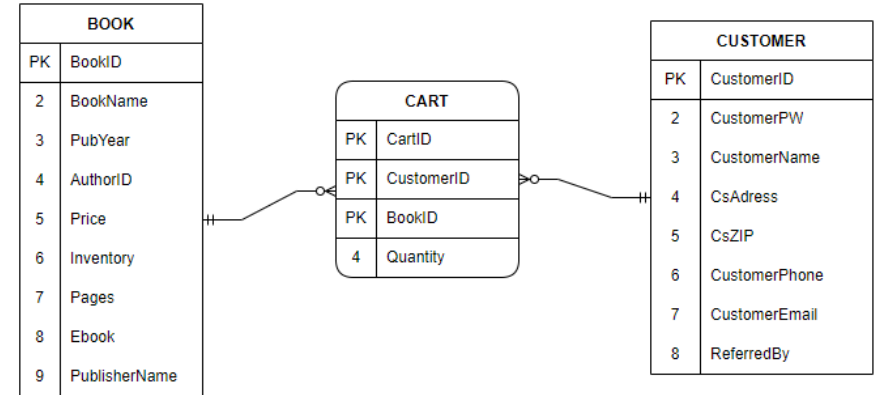
한번의 결제에 한번의 주문이 이루어지기 때문에
1:1 관계이고,
둘 다 **Mandatory-One** 이다.



COUPON - PAYMENT

PAYMENT 테이블의
CouponID (FK) 가
COUPON 테이블의
CouponID (PK)를 참조한다.

하나의 쿠폰은 여러 개의 결제에서 사용될 수 있고
사용되지 않을 수도 있기 때문에,
Optional-Many이다
하나의 결제에서 쿠폰은 사용되지 않을 수도 있고
최대 한 개까지 사용 가능하기 때문에
Optional-One이다.



BOOK – CART

CART 테이블의
BookID (FK이자 PK)가
BOOK 테이블의
BookID (PK)를 참조한다.

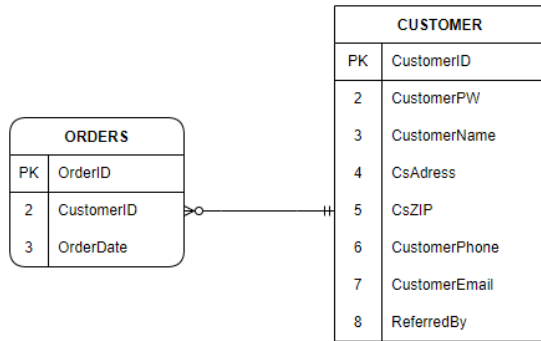
한 장바구니에는 한가지 상품만 담길 수
있다
따라서 **Mandatory-One** 이다
책 한권은 장바구니에 담기지 않을 수도
있고
여러 개의 장바구니에 담길 수도 있다.
따라서 **Optional-Many** 이다.

CART - CUSTOMER

CART 테이블의
BookID (FK이자 PK)가
BOOK 테이블의
BookID (PK)를 참조한다.

한 카트는 한 사람의 고객에게 할당되는 것이다
따라서 **Mandatory-One** 이다
한 사람은 장바구니를 가지지 않을 수 있고,
각기 다른 ID를 가진 카트를 여러 개 가질 수
있다.
따라서 **Optional-Many** 이다.

개념설계 - 관계



CUSTOMER – ORDERS

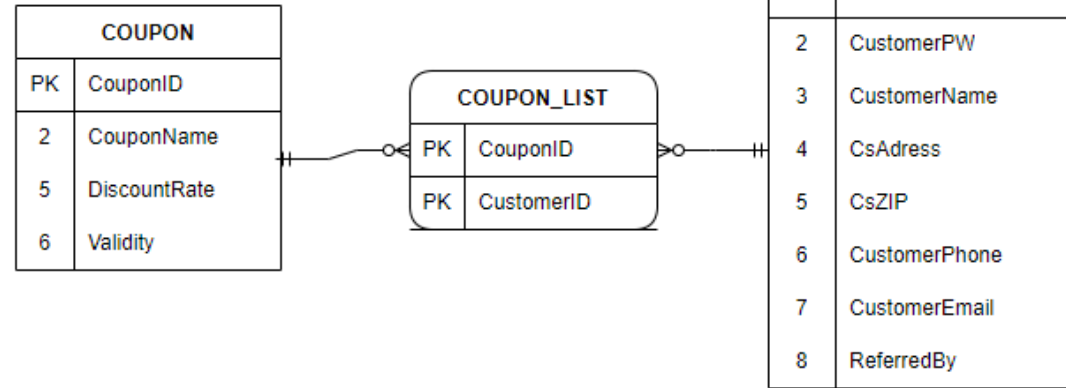
ORDERS 테이블의
CustomerID (FK이자 PK)가
CUSTOMER 테이블의
CustomerID (PK)를 참조한다.

한 사람의 고객은 여러 개의 주문을 할 수도 있고
주문을 안 할 수도 있기 때문에

Optional-Many이고

한 개의 주문은 고객 없이 존재할 수 없고,
한 사람의 고객이 하는 것이기 때문에

Mandatory-One 이다.



COUPON – COUPONLIST

COUPONLIST 테이블의
CouponID (FK)이
COUPON 테이블의
CouponID (PK)를 참조한다.

발급된 하나의 쿠폰은 여러 명의 고객에게
주어질 수도 있고, 아무에게도 주어지지 않을 수
있다

따라서 **Optional-Many**이다.

쿠폰 목록은 쿠폰 없이는 존재할 수 없고
한 개의 쿠폰과 한 명의 고객의 조합을 가진다

따라서 **Mandatory-One**이다.

COUPONLIST – CUSTOMER

COUPONLIST 테이블의
CustomerID (FK)이
CUSTOMER 테이블의
CustomerID (PK)를 참조한다.

한 명의 고객은 여러 개의 쿠폰을 가질 수도 있고
한 개의 쿠폰도 가지지 않을 수 있다.

따라서 **Optional-many** 이다

쿠폰 목록은 고객 없이는 존재할 수 없고
한 개의 쿠폰과 한 명의 고객의 조합을 가진다

따라서 **Mandatory-One**이다.

개념설계 -관계

AUTHOR – BOOK: 한번 출간한 책의 작가는 바뀌지 않고, 작가ID가 삭제 된다 하더라도, 책의 저자 정보는 그대로 남아있어야 하기 때문에 **ON UPDATE CASCADE**와 **ON DELETE CASCADE**는 설정하지 않는다.

AUTHOR – PUBLISHER: 출판사 이름이 바뀌게 되면 이를 refer하는 AUTHOR table의 PublisherName도 바뀌어야 하지만 삭제되게 된다면 해당 작가가 무소속인 작가로 존재해도 되기 때문에 **ON UPDATE CASCADE**는 설정, **ON DELETE CASCADE**는 설정하지 않는다.

PUBLISHER – BOOK: 출간된 책의 출판사의 이름은 바뀌지 않고, 출판사 정보가 삭제된다 하더라도 책의 출판사 정보는 남아있어야 하기 때문에 **ON UPDATE CASCADE**, **ON DELETE CASCADE**를 설정하지 않는다.

BOOK – REVIEW: BOOK table의 BookID가 변하면 이를 refer 하는 리뷰에서의 BookID도 바뀌어야 하므로 **ON UPDATE CASCADE**를 설정한다. BookID가 삭제되어도 Review는 남아있기 때문에 **ON DELETE CASCADE**는 설정하지 않는다.

BOOK – CART: BOOK TABLE의 BookID가 바뀌면 이를 해당 책이 담겨있는 장바구니의 BookID도 함께 변해야 하므로 **ON UPDATE CASCADE**를 설정한다. 또한 책 정보가 삭제되면 해당 책이 담겨있는 장바구니 정보도 삭제되어야 하므로 **ON DELETE CASCADE**를 설정해준다.

ORDERS – ORDERINFO: ORDERS table의 OrderID 가 바뀌게 되면 이를 refer하는 ORDERINFO의 OrderID도 바뀌어야 하기 때문에 **ON UPDATE CASCADE**를 설정한다. OrderID가 삭제되면 ORDERINFO의 관련 정보도 같이 삭제되어야 하므로 **ON DELETE CASCADE**를 설정한다.

ORDERS – REVIEW: ORDERS의 OrderID가 바뀌게 되면 refer하는 Review table의 OrderID도 함께 바뀌어야 하므로 **ON UPDATE CASCADE**를 설정해준다. OrderID가 삭제되어도 review는 사라지지 않으므로 **ON DELETE CASCADE**를 설정하지 않는다.

BOOK – ORDERINFO: BOOK table의 BOOK ID가 바뀌게 되면 refer하는 ORDERINFO 의 BookID도 같이 바뀌어야 하므로 **ON UPDATE CASCADE**를 설정한다. Book id가 삭제되어도 주문상세에 대한 정보는 존재해야 하기 때문에 **ON DELETE CASCADE**를 설정하지 않는다.

개념설계 - 관계

CUSTOMER – ORDERS: 회원 ID가 바뀌게 되면 해당 내용을 참조하는 주문에서의 회원ID 도 변경되어야 하므로 **ON UPDATE CASCADE**를 설정한다. 회원ID가 삭제되어도 주문했던 내용은 남아있어야 하므로 **ON DELETE CASCADE**를 설정하지 않는다.

ORDERS – PAYMENT: ORDERS의 OrderID가 바뀌면 refer 하는PAYMENT의 OrderID도 바뀌어야 하므로 **ON UPDATE CASCADE**를 설정한다.
주문 정보가 삭제되더라도 결제 정보는 삭제되지 않으므로 **ON DELETE CASCADE**를 설정하지 않는다.

ORDERS – DELIVERY: ORDERS의 OrderID 가 변경되면 이를 refer하는 DLEIVERY의 OrderID도 변경해주어야 하므로 **ON UPDATE CASCADE**를 설정한다. 그러나 OrderID가 삭제되어도 배송 정보는 남아있어야하기 때문에 **ON DELETE CASCADE**는 설정하지 않는다.

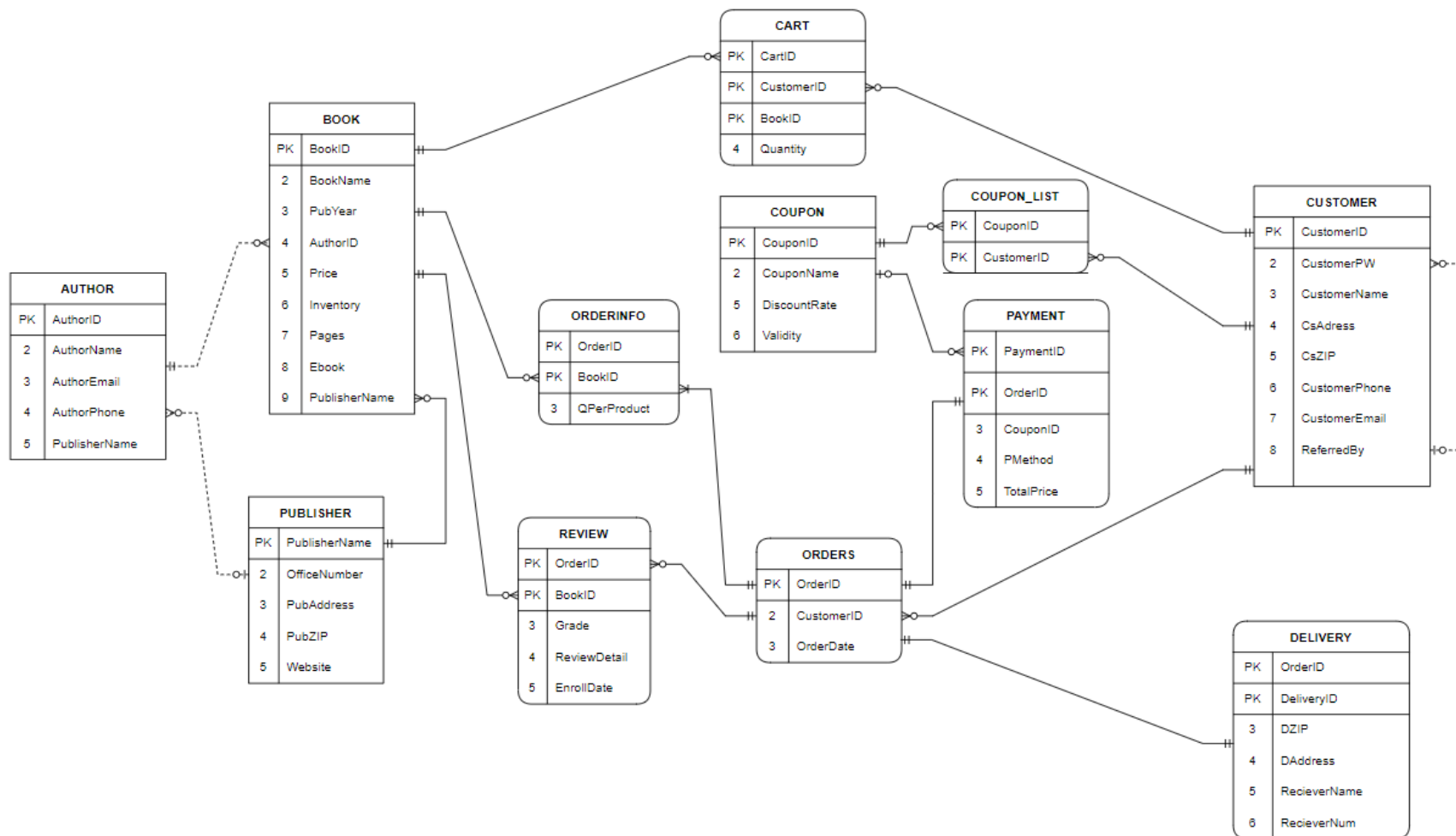
COUPON – COUPONLIST: COUPON의 CouponID가 변경되거나 삭제되면 refer하는 COUPONLIST 의 CouponID를 같이 변경하거나 관련 정보를 삭제해주어야 하므로 **ON UPDATE CASCADE, ON DELETE CASCADE**를 모두 설정한다.

COUPON – PAYMENT: COUPON의 CouponID가 변경되면 refer하는 PAYMENT의 CouponID도 변경되어야 하므로 **ON UPDATE CASCADE**를 설정한다. 하지만 CouponID가 삭제되어도 PAYMENT에서는 결제에 대한 내용이므로 삭제하면 안된다. 따라서 **ON DELETE CASCADE**를 설정하지 않는다.

COUPONLIST – CUSTOMER: CUSTOMER 의 CustomerID가 바뀌면 refer하는 COUPONLIST의 CustomerID가 바뀌거나 삭제되면 COUPONLIST에서도 변경하거나 관련 내용을 삭제해주어야 하므로 **ON UPDATE CASCADE, ON DELETE CASCADE**모두 설정해준다.

CUSTOMER – CART: CUSTOMER table의 CustomerID가 바뀌면 refer하는 Cart table의 CustomerID도 함께 바뀌어야 하므로 **ON UPDATE CASCADE**를 설정해준다. 또한 CustomerId가 삭제되면(회원 탈퇴) 장바구니 내부의 관련 항목도 삭제되어야 하므로 **ON DELETE CASCADE**도 설정해준다

ERD



최종 ERD는 다음과 같다.

SQL

테이블 생성과 디자인/데이터 삽입 쿼리문

SQL- PUBLISHER TABLE

```
CREATE TABLE PUBLISHER(  
    PublisherName VarChar(30) NOT NULL UNIQUE,  
    OfficeNumber Char(25) NOT NULL,  
    PublisherAddress VarChar(100) NOT NULL,  
    PubZIP Char(30) NOT NULL,  
    WebSite VarChar(100) NOT NULL,  
    CONSTRAINT PUBLISHER_PK PRIMARY KEY(PublisherName)  
);
```



PUBLISHER TABLE 생성 쿼리문

열 이름	데이터 형식	Null 허용
PublisherName	varchar(30)	<input type="checkbox"/>
OfficeNumber	char(25)	<input type="checkbox"/>
PublisherAddress	varchar(100)	<input type="checkbox"/>
PubZIP	char(30)	<input type="checkbox"/>
WebSite	varchar(100)	<input type="checkbox"/>



PUBLISHER TABLE 디자인

```
INSERT INTO PUBLISHER VALUES('미래의창', '02-2026-5777', '서울특별시 금천구 가산디지털 1로 168 우림라이온스밸리 B동', '12345', 'www.book.co.kr')  
,('좋은땅', '02-374-8616', '서울특별시 마포구 양화로 12길 26 지월드빌딩', '12346', 'www.g-world.co.kr')  
,('생각의 길', '02-322-3409', '서울특별시 마포구 서교동 442-1 성광빌딩 202호', '12347', 'www.trustbook.co.kr')  
,('마음산책', '02-362-1452', '서울특별시 마포구 잔다리로 3안길 20 마음산책빌딩 5층', '12348', 'www.maumsan.co.kr')  
,('아이이북', '02-338-3690', '서울특별시 마포구 동교로36', '12349', 'www.iebook.co.kr')  
,('북로망스', '02-123-4567', '서울특별시 마포구 동교로 38', '12340', 'www.bookro.co.kr')  
,('더퀘스트', '02-330-9704', '서울특별시 마포구 서교동 467-9', '12356', 'www.theQuest.co.kr')  
,('북스톤', '02-6499-1706', '서울특별시 성동구 연무장 7길 11 우리문화재단 8층', '12357', 'www.book-stone.co.kr')  
,('힘찬북스', '02-123-4567', '서울특별시 마포구 동교로 41', '23458', 'www.himchanbooks.co.kr')  
,('지식과 감성', '070-4651-3730', '서울특별시 금천구 가산동 50-3 대륭포스트타워', '34567', 'www.knsbookup.com')  
,('바른북스', '070-7610-9820', '서울특별시 성동구 연무장 5길 9-16', '45678', 'www.barunbooks.co.kr')  
,('문학동네', '031-906-5079', '경기도 고양시 일산서구 중앙로 1456 서현프라자', '14567', 'www.culturezon.co.kr')  
,('코리아닷컴', '031-459-8830', '경기도 군포시 군포로 697번길 5', '17890', 'www.koreadot.co.kr')  
,('미디어숲', '010-7933-3826', '경기도 남양주시 진건오남로 스튜디오', '16890', 'www.midforest.co.kr')  
,('비즈니스북스', '062-716-8533', '서울특별시 강동구 올림픽로 660 천호엘크루주상 복합 408호', '24567', 'www.businessbooks.co.kr')
```



PUBLISHER TABLE 에 데이터 삽입 쿼리문

SQL – PUBLISHER TABLE

```
SELECT *  
FROM PUBLISHER;
```

→ PUBLISHER TABLE 데이터 조회
쿼리문

결과 메시지					
	PublisherName	OfficeNumber	PublisherAddress	PubZIP	WebSite
1	더퀘스트	02-330-9704	서울특별시 마포구 서교동 467-9	12356	www.theQuest.co.kr
2	마음산책	02-362-1452	서울특별시 마포구 잔다리로 3안길 20 마음산책빌딩 5층	12348	www.maumsan.co.kr
3	문학동네	031-906-5079	경기도 고양시 일산서구 중앙로 1456 서현프라자	14567	www.culturezon.co.kr
4	미디어숲	010-7933-3826	경기도 남양주시 진건오남로 스튜디오	16890	www.midforest.co.kr
5	미래의창	02-2026-5777	서울특별시 금천구 가산디지털 1로 168 우림라이온스 밸리 B동	12345	www.book.co.kr
6	바른북스	070-7610-9820	서울특별시 성동구 연무장 5길 9-16	45678	www.barunbooks.co.kr
7	북로망스	02-123-4567	서울특별시 마포구 동교로 38	12340	www.bookro.co.kr
8	북스톤	02-6499-1706	서울특별시 성동구 연무장 7길 11 우리문화재단 8층	12357	www.book-stone.co.kr
9	비즈니스북스	062-716-8533	서울특별시 강동구 올림픽로 660 천호엘크루주상 복합 408호	24567	www.businessbooks.co.kr
10	생각의 길	02-322-3409	서울특별시 마포구 서교동 442-1 성광빌딩 202호	12347	www.trustbook.co.kr
11	아이이북	02-338-3690	서울특별시 마포구 동교로36	12349	www.iebook.co.kr
12	좋은땅	02-374-8616	서울특별시 마포구 양화로 12길 26 지월드빌딩	12346	www.g-world.co.kr
13	지식과 감성	070-4651-3730	서울특별시 금천구 가산동 50-3 대륭포스트타워	34567	www.knsbookup.com
14	코리아닷컴	031-459-8830	경기도 군포시 군포로 697번길 5	17890	www.koreadot.co.kr
15	힘찬북스	02-123-4567	서울특별시 마포구 동교로 41	23458	www.himchanbooks.co.kr

→ PUBLISHER TABLE 데이터 조회 결과

SQL-AUTHOR TABLE

```
CREATE TABLE AUTHOR(  
    AuthorID int NOT NULL IDENTITY(1,1),  
    AuthorName Char(25) NOT NULL,  
    AuthorEmail VarChar(100) NULL,  
    AuthorPhone Char(25) NOT NULL,  
    PublisherName VarChar(30) NULL,  
    CONSTRAINT AUTHOR_PK PRIMARY KEY(AuthorID),  
    CONSTRAINT AUTHOR_FK FOREIGN KEY (PublisherName)  
        REFERENCES PUBLISHER (PublisherName)  
        ON UPDATE CASCADE,  
    CONSTRAINT AUTHOR_Phone UNIQUE (AuthorPhone)  
);
```

➡ AUTHOR TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
▶	AuthorID	int	<input type="checkbox"/>
	AuthorName	char(25)	<input type="checkbox"/>
	AuthorEmail	varchar(100)	<input checked="" type="checkbox"/>
	AuthorPhone	char(25)	<input type="checkbox"/>
	PublisherName	varchar(30)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

➡ AUTHOR TABLE 디자인

```
INSERT INTO AUTHOR VALUES('김난도', 'nando123@naver.com', '010-1234-5678', '미래의창')  
, ('김지수', 'jisu123@gmail.com', '010-1234-5679', '좋은땅')  
, ('유시민', 'simin123@gmail.com', '010-1234-5670', '생각의 길')  
, ('임종현', 'jonghyun123@naver.com', '010-1234-6789', '마음산책')  
, ('김초엽', 'cho345@gmail.com', '010-1234-6780', '아이이북')  
, ('한강', 'hangang345@naver.com', '010-1234-7890', '북로망스')  
, ('강기태', 'gitae456@naver.com', '010-1234-3456', '더퀘스트')  
, ('소윤', 'soyoon12@gmail.com', '010-1234-3457', '마음산책')  
, ('성소라', 'sora12@naver.com', '010-1234-3458', '북스톤')  
, ('김이섭', 'esub12@naver.com', '010-1234-3459', '힘찬북스')  
, ('송길영', 'gilyoung12@naver.com', '010-1234-3450', '지식과 감성')  
, ('김주리', 'juli12@gmail.com', '010-1234-4567', '문학동네')  
, ('박영숙', 'youngsuk12@gmail.com', '010-1234-4560', '북스톤')  
, ('오은영', 'eunyoung12@gmail.com', '010-1234-4561', '코리아닷컴')  
, ('김도윤', 'doyoon12@gmail.com', '010-1234-4562', '코리아닷컴')
```

➡ AUTHOR TABLE 데이터 삽입 쿼리문

SQL-AUTHOR TABLE

```
SELECT *  
FROM AUTHOR;
```



AUTHOR TABLE 데이터 조회 쿼리문

결과		메시지			
	AuthorID	AuthorName	AuthorEmail	AuthorPhone	PublisherName
1	1	김난도	nando123@naver.com	010-1234-5678	미래의창
2	2	김지수	jisu123@gmail.com	010-1234-5679	좋은땅
3	3	유시민	simin123@gmail.com	010-1234-5670	생각의 길
4	4	임종현	jonghyun123@naver.com	010-1234-6789	마음산책
5	5	김초엽	cho345@gmail.com	010-1234-6780	아이이북
6	6	한강	hangang345@naver.com	010-1234-7890	북로망스
7	7	강기태	gitae456@naver.com	010-1234-3456	더퀘스트
8	8	소윤	soyoon12@gmail.com	010-1234-3457	마음산책
9	9	성소라	sora12@naver.com	010-1234-3458	북스톤
10	10	김이섭	esub12@naver.com	010-1234-3459	힘찬북스
11	11	송길영	gilyoung12@naver.com	010-1234-3450	지식과 감성
12	12	김주리	juli12@gmail.com	010-1234-4567	문학동네
13	13	박영숙	youngsuk12@gmail.com	010-1234-4560	북스톤
14	14	오은영	eunyoung12@gmail.com	010-1234-4561	코리아닷컴
15	15	김도윤	doyoon12@gmail.com	010-1234-4562	코리아닷컴



AUTHOR TABLE 데이터 조회 결과

SQL-CUSTOMER TABLE

```
CREATE TABLE CUSTOMER(
  CustomerID VARCHAR(100) NOT NULL UNIQUE,
  CustomerPW VARCHAR(100) NOT NULL,
  CustomerName Char(30) NOT NULL,
  CsAddress Char(100) NOT NULL,
  CsZIP Char(20) NOT NULL,
  CustomerPhone Char(20) NOT NULL,
  CustomerEmail Char(100) NULL,
  ReferredBy VARCHAR(100),
  CONSTRAINT CUSTOMER_PK PRIMARY KEY(CustomerID),
  CONSTRAINT FK_Recursive FOREIGN KEY(ReferredBy)
    REFERENCES CUSTOMER(CustomerID)
);
```

→ CUSTOMER TABLE 생성 쿼리문

열 이름	데이터 형식	Null 허용
CustomerID	varchar(100)	<input type="checkbox"/>
CustomerName	char(30)	<input type="checkbox"/>
CsAddress	char(100)	<input type="checkbox"/>
CsZIP	char(20)	<input type="checkbox"/>
CustomerPhone	char(20)	<input type="checkbox"/>
CustomerEmail	char(100)	<input checked="" type="checkbox"/>
ReferredBy	varchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

→ CUSTOMER TABLE 디자인

```
INSERT INTO CUSTOMER VALUES('hongildong123','홍길동','경기도 성남시 분당구 신수로 791','13635','010-1112-3456','honggil@naver.com','hongildong123')
,('snk2020','김선남','경기도 성남시 분당구 탄천상로 95','13637','010-1113-3457','sunnam@gmail.com','snk2020')
,('sunh2201','곽선희','경기도 수원시 영통구 덕영대로 1483번길 16-4','16680','010-1114-3458','sunhee12@naver.com','sunh2201')
,('yiuli22','이율리','경기도 수원시 권선구 평동로 52-1','16601','010-1115-3456','leewool12@naver.com','yiuli22')
,('jhee103','박지희','경기도 파주시 검산로 173번길 78','10848','010-1116-3456','jihee1@gmail.com','jhee103')
,('honga223','홍가진','서울특별시 강남구 선릉로 8길 19','06325','010-1117-3456','gajin12@gmail.com','honga223')
,('hyungsuk12','이형석','서울특별시 강남구 테헤란로 6길9','06233','010-1118-3456','hyungsuk@naver.com','hyungsuk12')
,('ina89','홍이나','인천광역시 강화군 대월로 12-2','23025','010-1119-3456','lina123@naver.com','ina89')
,('yunghee103','김영희','인천광역시 동구 금곡로 62','22562','010-1110-3456','younghee12@gmail.com','yunghee103')
,('misun182','박미선','서울특별시 강서구 양천로 55길 55','07528','010-1122-3457','misun12@naver.com','misun182')
,('minsu88','김민수','서울특별시 성북구 길음로 92','02724','010-1135-3457','minsu12@naver.com','minsu88')
,('minho89','장민호','부산광역시 해운대구 반송로 886','48000','010-1138-3458','minho34@gmail.com','minho89')
,('jiho99','박지호','대전광역시 대덕구 대청호수로 1533','34316','010-1140-3450','jiho12@naver.com','jiho99')
,('gihun2','이기훈','대전광역시 유성구 대덕대로 523','34131','010-1145-6789','gihun12@naver.com','gihun2')
,('somang88','이소망','경기도 용인시 기흥구 원고매로 56-8','17086','010-1146-6789','somang12@gmail.com','somang88')
```

→ CUSTOMER TABLE 데이터 삽입 쿼리문

SQL-CUSTOMER TABLE

SELECT * FROM CUSTOMER

→ CUSTOMER TABLE 데이터 조회
쿼리문

결과	메시지	CustomerID	CustomerPW	CustomerName	CsAddress	CsZIP	CustomerPhone	CustomerEmail	ReferredBy
		gihun2	khl3909	이기훈	대전광역시 유성구 대덕대로 523	34131	010-1145-6789	gihun12@naver.com	gihun2
		hong223	hgj321	홍가진	서울특별시 강남구 선릉로 8길 19	06325	010-1117-3456	gajin12@gmail.com	hong223
		hongildong123	hgd123	홍길동	경기도 성남시 분당구 신수로 791	13635	010-1112-3456	honggil@naver.com	hongildong123
		hyungsuk12	lhs345	이형석	서울특별시 강남구 테헤란로 6길9	06233	010-1118-3456	hyungsuk@naver.com	hyungsuk12
		ina89	hina123	홍이나	인천광역시 강화군 대월로 12-2	23025	010-1119-3456	lina123@naver.com	ina89
		jhee103	pjh123	박지희	경기도 파주시 검산로 173번길 78	10848	010-1116-3456	jihee1@gmail.com	jhee103
		jiho99	jhpp2323	박지호	대전광역시 대덕구 대청호수로 1533	34316	010-1140-3450	jiho12@naver.com	jiho99
		minho89	mhj53	장민호	부산광역시 해운대구 반송로 886	48000	010-1138-3458	minho34@gmail.com	minho89
		minsu88	msms23	김민수	서울특별시 성북구 길음로 92	02724	010-1135-3457	minsu12@naver.com	minsu88
		misun182	msh3232	박미선	서울특별시 강서구 양천로 55길 55	07528	010-1122-3457	misun12@naver.com	misun182
		snk2020	snk123	김선남	경기도 성남시 분당구 탄천상로 95	13637	010-1113-3457	sunnam1@gmail.com	snk2020
		somang88	smsm12323	이소망	경기도 용인시 기흥구 원고매로 56-8	17086	010-1146-6789	somang12@gmail.com	somang88
		sunh2201	ksh123	곽선희	경기도 수원시 영통구 덕영대로 1483번길 16-4	16680	010-1114-3458	sunhee12@naver.com	sunh2201
		yiuli22	iuli123	이율리	경기도 수원시 권선구 평동로 52-1	16601	010-1115-3456	leewool12@naver.com	yiuli22
		yunghee103	yh122	김영희	인천광역시 동구 금곡로 62	22562	010-1110-3456	younghee12@gmail.com	yunghee103

→ CUSTOMER TABLE 데이터 조회 결과

SQL-BOOK TABLE

```
CREATE TABLE BOOK(  
  BookID int NOT NULL IDENTITY(1,1),  
  BookName VarChar(50) NOT NULL,  
  PubYear date NOT NULL,  
  AuthorID int NOT NULL,  
  Price int NOT NULL,  
  Inventory int NOT NULL,  
  Pages int NOT NULL,  
  Ebook VarChar(30) NOT NULL,  
  PublisherName VarChar(30) NOT NULL,  
  CONSTRAINT BOOK_PK PRIMARY KEY(BookID),  
  CONSTRAINT BOOK_FK FOREIGN KEY(AuthorID)  
    REFERENCES AUTHOR(AuthorID),  
  CONSTRAINT BOOK_FK2 FOREIGN KEY(PublisherName)  
    REFERENCES PUBLISHER(PublisherName)  
);
```

BOOK TABLE 생성 쿼리문

열 이름	데이터 형식	Null 허용
BookID	int	<input type="checkbox"/>
BookName	varchar(50)	<input type="checkbox"/>
PubYear	date	<input type="checkbox"/>
AuthorID	int	<input type="checkbox"/>
Price	int	<input type="checkbox"/>
Inventory	int	<input type="checkbox"/>
Pages	int	<input type="checkbox"/>
Ebook	varchar(30)	<input type="checkbox"/>
PublisherName	varchar(30)	<input type="checkbox"/>

BOOK TABLE 디자인

```
INSERT INTO BOOK VALUES('트랜드 코리아 2022', '2021', '2', '16200', '5', '452', 'YES', '미래의창')  
, ('피아니스트', '2020', '2', '13500', '1', '380', 'YES', '좋은땅')  
, ('유럽 도시 기행', '2019', '3', '14850', '2', '324', 'YES', '생각의 길')  
, ('바보', '2013', '4', '10800', '0', '248', 'NO', '지식과 감성')  
, ('행성어 서점', '2021', '5', '13050', '1', '216', 'YES', '마음산책')  
, ('작별하지 않는다', '2021', '6', '12600', '2', '332', 'NO', '문학동네')  
, ('180일간의 트랙터 다이어리', '2011', '7', '9100', '0', '200', 'YES', '북스톤')  
, ('작은 별이지만 빛나고 있어', '2021', '8', '13500', '1', '274', 'YES', '북로망스')  
, ('NFT 레볼루션', '2021', '9', '16200', '2', '324', 'YES', '더퀘스트')  
, ('인생의 답은 내 안에 있다', '2021', '10', '11060', '0', '300', 'NO', '미디어숲')  
, ('그냥 하지 말라', '2021', '11', '15300', '3', '284', 'YES', '북스톤')  
, ('사이다 공식으로 톡 쓰는 글쓰기 비법', '2021', '12', '13500', '4', '272', 'YES', '힘찬북스')  
, ('세계미래보고서 2021', '2020', '13', '16020', '1', '372', 'YES', '비즈니스북스')  
, ('오은영의 화해', '2019', '14', '14400', '3', '320', 'YES', '코리아닷컴')  
, ('럭키', '2021', '15', '14400', '2', '280', 'YES', '북로망스')
```

BOOK TABLE 데이터 삽입 쿼리문

SQL-BOOK TABLE

```
SELECT *  
FROM BOOK;
```

BOOK TABLE 데이터 조회 쿼리문

결과 메시지

	BookID	BookName	PubYear	AuthorID	Price	Inventory	Pages	Ebook	PublisherName
1	1	트랜드 코리아 2022	2021-01-01	2	16200	5	452	YES	미래의창
2	2	피아니스트	2020-01-01	2	13500	1	380	YES	좋은땅
3	3	유럽 도시 기행	2019-01-01	3	14850	2	324	YES	생각의 길
4	4	바보	2013-01-01	4	10800	0	248	NO	지식과 감성
5	5	행성어 서점	2021-01-01	5	13050	1	216	YES	마음산책
6	6	작별하지 않는다	2021-01-01	6	12600	2	332	NO	문학동네
7	7	180일간의 트랙터 다이어리	2011-01-01	7	9100	0	200	YES	북스톤
8	8	작은 별이지만 빛나고 있어	2021-01-01	8	13500	1	274	YES	북로망스
9	9	NFT 레볼루션	2021-01-01	9	16200	2	324	YES	더퀘스트
10	10	인생의 답은 내 안에 있다	2021-01-01	10	11060	0	300	NO	미디어숲
11	11	그냥 하지 말라	2021-01-01	11	15300	3	284	YES	북스톤
12	12	사이다 공식으로 톡 쓰는 글쓰기 비법	2021-01-01	12	13500	4	272	YES	힘찬북스
13	13	세계미래보고서 2021	2020-01-01	13	16020	1	372	YES	비즈니스북스
14	14	오은영의 화해	2019-01-01	14	14400	3	320	YES	코리아닷컴
15	15	럭키	2021-01-01	15	14400	2	280	YES	북로망스

BOOK TABLE 데이터 조회 결과

SQL-CART TABLE

```
CREATE TABLE CART(  
    CartID      int NOT NULL IDENTITY(1,1),  
    CustomerID  VARCHAR(100) NOT NULL,  
    BookID      int NOT NULL,  
    Quantity    int NOT NULL,  
    CONSTRAINT CART_PK PRIMARY KEY(CartID, CustomerID, BookID),  
    CONSTRAINT CART_FK FOREIGN KEY (CustomerID)  
        REFERENCES CUSTOMER (CustomerID)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    CONSTRAINT CART_FK2 FOREIGN KEY (BookID)  
        REFERENCES BOOK (BookID)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

→ CART TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
▼	CartID	int	<input type="checkbox"/>
▼	CustomerID	varchar(100)	<input type="checkbox"/>
▼	BookID	int	<input type="checkbox"/>
▶	Quantity	int	<input checked="" type="checkbox"/>

→ CART TABLE 디자인

```
INSERT INTO CART VALUES  
( '1', 'hongildong123', '2', '31' ),  
( '2', 'snk2020', '16', '36' ),  
( '3', 'sunh2201', '3', '24' ),  
( '4', 'yiuli22', '4', '33' ),  
( '5', 'jhee103', '5', '4' ),  
( '6', 'honga223', '6', '1' ),  
( '7', 'hyungsuk12', '7', '31' ),  
( '8', 'ina89', '9', '2' ),  
( '9', 'yunghee103', '8', '5' ),  
( '10', 'misun182', '12', '12' ),  
( '11', 'minsu88', '10', '7' ),  
( '12', 'minho89', '11', '5' ),  
( '13', 'jiho99', '14', '9' ),  
( '14', 'gihun2', '15', '2' ),  
( '15', 'somang88', '13', '1' )
```

→ CART TABLE 데이터 삽입 쿼리문

SQL-CART TABLE

SELECT * FROM CART



CART TABLE 데이터 조회 쿼리문

	CartID	CustomerID	BookID	Quantity
1	1	hongildong123	2	31
2	2	snk2020	1	36
3	3	sunh2201	3	24
4	4	yiuli22	4	33
5	5	jhee103	5	4
6	6	honga223	6	1
7	7	hyungsuk12	7	31
8	8	ina89	9	2
9	9	yunghee103	8	5
10	10	misun182	12	12
11	11	minsu88	10	7
12	12	minho89	11	5
13	13	jiho99	14	9
14	14	gihun2	15	2
15	15	somang88	13	1



CART TABLE 데이터 조회 결과

SQL-ORDERS TABLE

```
CREATE TABLE ORDERS (  
    OrderID      int NOT NULL IDENTITY(1,1),  
    CustomerID   VARCHAR(100) NOT NULL,  
    OrderDate    Date NOT NULL,  
    CONSTRAINT ORDERS_PK PRIMARY KEY(OrderID),  
    CONSTRAINT ORDERS_FK2 FOREIGN KEY(CustomerID)  
        REFERENCES CUSTOMER(CustomerID)  
        ON UPDATE CASCADE  
);
```

➡ ORDERS TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
▶	OrderID	int	<input type="checkbox"/>
	CustomerID	varchar(100)	<input type="checkbox"/>
	OrderDate	date	<input type="checkbox"/>
			<input type="checkbox"/>

➡ ORDERS TABLE 디자인

```
INSERT INTO ORDERS VALUES  
( 'hongildong123', '09/02/2021' ),  
( 'snk2020', '09/12/2021' ),  
( 'sunh2201', '09/14/2021' ),  
( 'yiuli22', '09/21/2021' ),  
( 'jhee103', '09/22/2021' ),  
( 'honga223', '09/23/2021' ),  
( 'hyungsuk12', '09/24/2021' ),  
( 'ina89', '09/25/2021' ),  
( 'yunghee103', '10/02/2021' ),  
( 'misun182', '10/03/2021' ),  
( 'minsu88', '10/05/2021' ),  
( 'minho89', '11/05/2021' ),  
( 'jiho99', '11/10/2021' ),  
( 'gihun2', '11/11/2021' ),  
( 'somang88', '11/12/2021' );
```

➡ ORDERS TABLE 데이터 삽입 쿼리문

SQL-ORDERS TABLE

SELECT * FROM ORDERS



ORDERS TABLE 데이터 조회 쿼리문

결과 메시지

	OrderID	CustomerID	OrderDate
1	1	hongildong123	2021-09-02
2	2	snk2020	2021-09-12
3	3	sunh2201	2021-09-14
4	4	yiuli22	2021-09-21
5	5	jhee103	2021-09-22
6	6	honga223	2021-09-23
7	7	hyungsuk12	2021-09-24
8	8	ina89	2021-09-25
9	9	yunghee103	2021-10-02
10	10	misun182	2021-10-03
11	11	minsu88	2021-10-05
12	12	minho89	2021-11-05
13	13	jiho99	2021-11-10
14	14	gihun2	2021-11-11
15	15	somang88	2021-11-12



ORDERS TABLE 데이터 조회 결과

SQL-ORDER INFO TABLE

```
CREATE TABLE ORDERINFO (  
  OrderID int NOT NULL,  
  BookID int NOT NULL,  
  QperProduct int NOT NULL,  
  CONSTRAINT ORDER_PK PRIMARY KEY(OrderID,BookID),  
  
  CONSTRAINT ORDER_FK FOREIGN KEY(BookID)  
    REFERENCES BOOK(BookID)  
    ON UPDATE CASCADE,  
  CONSTRAINT ORDER_FK2 FOREIGN KEY(OrderID)  
    REFERENCES ORDERS(OrderID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

➡ ORDERINFO TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
🔍	OrderID	int	<input type="checkbox"/>
🔍	BookID	int	<input type="checkbox"/>
	QperProduct	int	<input type="checkbox"/>
			<input type="checkbox"/>

➡ ORDERINFO TABLE 디자인

```
INSERT INTO ORDERINFO VALUES  
(1, 1, 2),  
(1, 2, 1),  
(1, 5, 1),  
(2, 6, 1),  
(2, 9, 3),  
(5, 10, 2),  
(7, 5, 1),  
(7, 2, 1),  
(4, 3, 1),  
(4, 2, 2),  
(10, 7, 1),  
(12, 10, 2),  
(12, 2, 1),  
(14, 9, 4),  
(14, 12, 2)
```

➡ ORDERINFO TABLE 데이터 삽입 쿼리문

SQL-ORDER INFO TABLE

```
SELECT *  
FROM ORDERINFO;
```



ORDERINFO TABLE 데이터 조회
쿼리문

결과 메시지

	OrderID	BookID	QperProduct
1	1	1	2
2	1	2	1
3	1	5	1
4	2	6	1
5	2	9	3
6	4	2	2
7	4	3	1
8	5	10	2
9	7	2	1
10	7	5	1
11	10	7	1
12	12	2	1
13	12	10	2
14	14	9	4
15	14	12	2



ORDERINFO TABLE 데이터 조회 결과

SQL-PAYMENT TABLE

```
CREATE TABLE PAYMENT(  
    PaymentID int NOT NULL UNIQUE,  
    OrderID int NOT NULL UNIQUE,  
    CouponID int NULL,  
    PMethod Varchar(20) NOT NULL,  
    TotalPrice int NOT NULL,  
    CONSTRAINT PAYMENT_PK PRIMARY KEY(PaymentID,OrderID),  
    CONSTRAINT PAYMENT_FK FOREIGN KEY (OrderID)  
        REFERENCES ORDERS(OrderID)  
        ON UPDATE CASCADE,  
    CONSTRAINT PAYMENT_FK2 FOREIGN KEY (CouponID)  
        REFERENCES COUPON (CouponID)  
        ON UPDATE CASCADE  
);
```

➡ PAYMENT TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
🔑	PaymentID	int	<input type="checkbox"/>
🔑	OrderID	int	<input type="checkbox"/>
	CouponID	int	<input checked="" type="checkbox"/>
	PMethod	varchar(20)	<input type="checkbox"/>
	TotalPrice	int	<input type="checkbox"/>
			<input type="checkbox"/>

➡ PAYMENT TABLE 디자인

```
INSERT INTO PAYMENT VALUES  
(1,1,2, 'Card', '30000'),  
(2,3,2, 'Card', '15000'),  
(3,2,3, 'ATM', '24000'),  
(4,6,1, 'KAKAOPAY', '17000'),  
(5,7,6, 'Card', '25000'),  
(6,4,8, 'NAVERPAY', '12000'),  
(7,8,9, 'Card', '16500'),  
(8,5,10, 'ATM', '14500'),  
(9,9,11, 'Card', '34000'),  
(10,11,13, 'KAKAOPAY', '9800'),  
(11,10,7, 'Card', '45000'),  
(12,13,4, 'ATM', '37000'),  
(13,15,1, 'Card', '25500'),  
(14,12,9, 'Card', '30000'),  
(15,14,10, 'Card', '14000')
```

➡ PAYMENT TABLE 데이터 삽입 쿼리문

SQL-PAYMENT TABLE

```
SELECT *  
FROM PAYMENT;
```



PAYMENT TABLE 데이터 조회 쿼리문

	PaymentID	OrderID	CouponID	PMethod	TotalPrice
1	1	1	2	Card	30000
2	2	3	2	Card	15000
3	3	2	3	ATM	24000
4	4	6	1	KAKAOPAY	17000
5	5	7	6	Card	25000
6	6	4	8	NAVERPAY	12000
7	7	8	9	Card	16500
8	8	5	10	ATM	14500
9	9	9	11	Card	34000
10	10	11	13	KAKAOPAY	9800
11	11	10	7	Card	45000
12	12	13	4	ATM	37000
13	13	15	1	Card	25500
14	14	12	9	Card	30000
15	15	14	10	Card	14000



PAYMENT TABLE 데이터 조회 결과

SQL-COUPON TABLE

```
CREATE TABLE COUPON(  
    CouponID int NOT NULL IDENTITY(1,1),  
    CouponName VarChar(50) NOT NULL,  
    DiscountRate Char(10) NOT NULL,  
    Validity Date NULL,  
    CONSTRAINT COUPON_PK PRIMARY KEY(CouponID)  
);
```

→ COUPON TABLE 생성 쿼리문

열 이름	데이터 형식	Null 허용
CouponID	int	<input type="checkbox"/>
CouponName	varchar(50)	<input type="checkbox"/>
DiscountRate	char(10)	<input type="checkbox"/>
Validity	date	<input checked="" type="checkbox"/>
	nchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

→ COUPON TABLE 디자인

```
INSERT INTO COUPON VALUES  
(  
    '크리스마스 기념 이벤트', '10%', '01/07/2021'),  
    ('10주년 기념 이벤트', '15%', '12/22/2020'),  
    ('15000원 이상 구매 시 할인', '5%', '04/21/2021'),  
    ('20000원 이상 구매 시 할인', '10%', '04/21/2021'),  
    ('30000원 이상 구매 시 할인', '15%', '04/21/2021'),  
    ('실버 회원 우대', '15%', '11/12/2020'),  
    ('골드 회원 우대', '20%', '08/17/2021'),  
    ('신학기 대학생 지원 이벤트', '20%', '06/09/2018'),  
    ('수험생을 위한 선물', '30%', '03/05/2020'),  
    ('신규 가입자 환영', '25%', '02/12/2019'),  
    ('1년 이상 장기 고객 우대', '15%', '07/16/2018'),  
    ('즐거운 한가위는 독서와 함께!', '5%', '11/15/2019'),  
    ('새해 복 많이 받으세요!', '15%', '03/01/2021'),  
    ('50000원 이상 구매 시 할인', '20%', '03/11/2020'),  
    ('할로윈 데이 깜짝 이벤트', '10%', '11/30/2021')
```

→ COUPON TABLE 데이터 삽입 쿼리문

SQL-COUPON TABLE

```
SELECT *  
FROM COUPON;
```



COUPON TABLE 데이터 조회 쿼리문

결과

메시지

	CouponID	CouponName	DiscountRate	Validity
1	1	크리스마스 기념 이벤트	10%	2021-01-07
2	2	10주년 기념 이벤트	15%	2020-12-22
3	3	15000원 이상 구매 시 할인	5%	2021-04-21
4	4	20000원 이상 구매 시 할인	10%	2021-04-21
5	5	30000원 이상 구매 시 할인	15%	2021-04-21
6	6	실버 회원 우대	15%	2020-11-12
7	7	골드 회원 우대	20%	2021-08-17
8	8	신학기 대학생 지원 이벤트	20%	2018-06-09
9	9	수험생을 위한 선물	30%	2020-03-05
10	10	신규 가입자 환영	25%	2019-02-12
11	11	1년 이상 장기 고객 우대	15%	2018-07-16
12	12	즐거운 한가위는 독서와 함께!	5%	2019-11-15
13	13	새해 복 많이 받으세요!	15%	2021-03-01
14	14	50000원 이상 구매 시 할인	20%	2020-03-11
15	15	할로윈 데이 깜짝 이벤트	10%	2021-11-30



COUPON TABLE 데이터 조회 결과

SQL-COUPONLIST TABLE

```
CREATE TABLE COUPONLIST(  
    CouponID    int NOT NULL UNIQUE,  
    CustomerID  VARCHAR(100) NOT NULL UNIQUE,  
    CONSTRAINT COUPONLIST_PK PRIMARY KEY(CouponID, CustomerID),  
    CONSTRAINT COUPONLIST_FK FOREIGN KEY (CouponID)  
        REFERENCES COUPON (CouponID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT COUPONLIST_FK2 FOREIGN KEY (CustomerID)  
        REFERENCES CUSTOMER (CustomerID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

→ COUPONLIST TABLE 생성
쿼리문

	열 이름	데이터 형식	Null 허용
🔑	CouponID	int	<input type="checkbox"/>
🔑	CustomerID	varchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

→ COUPONLIST TABLE 디자인

```
INSERT INTO COUPONLIST VALUES  
(1, 'hongildong123'),  
(2, 'snk2020'),  
(3, 'sunh2201'),  
(4, 'yiuli22'),  
(5, 'jhee103'),  
(6, 'honga223'),  
(7, 'hyungsuk12'),  
(8, 'ina89'),  
(9, 'yunghee103'),  
(10, 'misun182'),  
(11, 'minsu88'),  
(12, 'minho89'),  
(13, 'jiho99'),  
(14, 'gihun2'),  
(15, 'somang88')
```

→ COUPONLIST TABLE 데이터 삽입
쿼리문

SQL-COUPONLIST TABLE

SELECT * FROM COUPONLIST



COUPONLIST TABLE 데이터 조회 쿼리문

	CouponID	CustomerID
1	1	hongildong123
2	2	snk2020
3	3	sunh2201
4	4	yiuli22
5	5	jhee103
6	6	honga223
7	7	hyungsuk12
8	8	ina89
9	9	yunghee103
10	10	misun182
11	11	minsu88
12	12	minho89
13	13	jiho99
14	14	gihun2
15	15	somang88



COUPONLIST TABLE 데이터 조회
결과

SQL-DELIVERY TABLE

```
CREATE TABLE DELIVERY (
  OrderID      int NOT NULL UNIQUE,
  DeliveryID   int NOT NULL IDENTITY(1,1) UNIQUE,
  DZip         varchar(30) NOT NULL,
  DAddress     varchar(100) NOT NULL,
  RecieverName char(20) NOT NULL,
  RecieverNumber char(20) NOT NULL,
  CONSTRAINT DELIVERY_PK PRIMARY KEY (OrderID, DeliveryID),
  CONSTRAINT DELIVERY_FK FOREIGN KEY (OrderID)
    REFERENCES ORDERS (OrderID)
    ON UPDATE CASCADE
);
```

DELIVERY TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
🔑	OrderID	int	<input type="checkbox"/>
🔑	DeliveryID	int	<input type="checkbox"/>
	DZip	varchar(30)	<input type="checkbox"/>
	DAddress	varchar(100)	<input type="checkbox"/>
	RecieverName	char(20)	<input type="checkbox"/>
	RecieverNumber	char(20)	<input type="checkbox"/>
			<input type="checkbox"/>

DELIVERY TABLE 디자인

INSERT INTO DELIVERY VALUES

```
(1, '16680', '경기도 수원시 영통구 덕영대로 1483번길 16-4', '곽선희', '010-1114-3458'),
(2, '13637', '경기도 성남시 분당구 탄천상로 95', '김선남', '010-1113-3457'),
(3, '13635', '경기도 성남시 분당구 신수로 791', '홍길동', '010-1112-3456'),
(4, '16601', '경기도 수원시 권선구 평동로 52-1', '이율리', '010-1115-3456'),
(5, '10848', '경기도 파주시 검산로 173번길 78', '박지희', '010-1116-3456'),
(6, '06325', '서울특별시 강남구 선릉로 8길 19', '홍가진', '010-1117-3456'),
(7, '06233', '서울특별시 강남구 테헤란로 6길9', '이형석', '010-1118-3456'),
(8, '23025', '인천광역시 강화군 대월로 12-2', '홍이나', '010-1119-3456'),
(9, '22562', '인천광역시 동구 금곡로 62', '김영희', '010-1110-3456'),
(10, '07528', '서울특별시 강서구 양천로 55길 55', '박미선', '010-1122-3457'),
(11, '02724', '서울특별시 성북구 길음로 92', '김민수', '010-1135-3457'),
(12, '48000', '부산광역시 해운대구 반송로 886', '장민호', '010-1138-3458'),
(13, '34316', '대전광역시 대덕구 대청호수로 1533', '박지호', '010-1140-3450'),
(14, '34131', '대전광역시 유성구 대덕대로 523', '이기훈', '010-1145-6789'),
(15, '17086', '경기도 용인시 기흥구 원고매로 56-8', '이소망', '010-1146-6789')
```

DELIVERY TABLE 데이터 삽입 쿼리문

SQL-DELIVERY TABLE

```
SELECT *  
FROM DELIVERY;
```



DELIVERY TABLE 데이터 조회 쿼리문

결과 메시지

	OrderID	DeliveryID	DZip	DAddress	RecieverName	RecieverNumber
1	1	1	16680	경기도 수원시 영통구 덕영대로 1483번길 16-4	곽선희	010-1114-3458
2	2	2	13637	경기도 성남시 분당구 탄천상로 95	김선남	010-1113-3457
3	3	3	13635	경기도 성남시 분당구 신수로 791	홍길동	010-1112-3456
4	4	4	16601	경기도 수원시 권선구 평동로 52-1	이율리	010-1115-3456
5	5	5	10848	경기도 파주시 검산로 173번길 78	박지희	010-1116-3456
6	6	6	06325	서울특별시 강남구 선릉로 8길 19	홍가진	010-1117-3456
7	7	7	06233	서울특별시 강남구 테헤란로 6길9	이형석	010-1118-3456
8	8	8	23025	인천광역시 강화군 대월로 12-2	홍이나	010-1119-3456
9	9	9	22562	인천광역시 동구 금곡로 62	김영희	010-1110-3456
10	10	10	07528	서울특별시 강서구 양천로 55길 55	박미선	010-1122-3457
11	11	11	02724	서울특별시 성북구 길음로 92	김민수	010-1135-3457
12	12	12	48000	부산광역시 해운대구 반송로 886	장민호	010-1138-3458
13	13	13	34316	대전광역시 대덕구 대청호수로 1533	박지호	010-1140-3450
14	14	14	34131	대전광역시 유성구 대덕대로 523	이기훈	010-1145-6789
15	15	15	17086	경기도 용인시 기흥구 원고매로 56-8	이소망	010-1146-6789



DELIVERY TABLE 데이터 조회 결과

SQL-REVIEW TABLE

```
CREATE TABLE REVIEW(  
    OrderID int NOT NULL,  
    BookID int NOT NULL,  
    Grade Numeric(2,1) NULL,  
    ReviewDetail VarChar(100) NULL,  
    EnrollDate Date NOT NULL,  
    CONSTRAINT REVIEW_PK PRIMARY KEY(OrderID, BookID),  
    CONSTRAINT REVIEW_FK FOREIGN KEY(OrderID)  
        REFERENCES ORDERS(OrderID)  
        ON UPDATE CASCADE,  
    CONSTRAINT REVIEW_FK2 FOREIGN KEY(BookID)  
        REFERENCES BOOK(BookID)  
        ON UPDATE CASCADE  
);
```

➡ REVIEW TABLE 생성 쿼리문

	열 이름	데이터 형식	Null 허용
PK	OrderID	int	<input type="checkbox"/>
PK	BookID	int	<input type="checkbox"/>
	Grade	numeric(2, 1)	<input checked="" type="checkbox"/>
	ReviewDetail	varchar(100)	<input checked="" type="checkbox"/>
	EnrollDate	date	<input type="checkbox"/>
			<input type="checkbox"/>

➡ REVIEW TABLE 디자인

```
INSERT INTO REVIEW VALUES  
(2, 5, '4.2', '한 번쯤 읽어봐야 할 자기계발서이다', '11/23/2021'),  
(1, 4, '3.6', '그저 그런 책', '10/13/2021'),  
(5, 8, '3.2', '장르와 관련이 있는 지 모르겠다', '04/12/2021'),  
(3, 6, '4.2', '배경지식을 쌓기에 유용한 도서라고 생각함!', '08/20/2021'),  
(8, 11, '4.8', '왜 베스트 셀러인지 첫 챕터에서 바로 알 수 있는 책', '07/08/2021'),  
(9, 12, '2.6', '돈이 아깝다. 읽지 마세요.', '11/12/2021'),  
(13, 1, '4.9', '서재에 보관하고 계속 읽어야 할 책인 것 같다.', '08/04/2021'),  
(14, 2, '4.0', '아이들에게도 읽게 해주면 좋을 만한 소재를 다룬 듯 하다.', '02/20/2020'),  
(11, 14, '4.0', '', '05/22/2019'),  
(10, 13, '3.5', '누구나 한 번쯤 고민하는 것들을 잘 정리해서 쓴 책임. 추천함.', '07/10/2021'),  
(4, 7, '2.8', '독자의 성향에 따라 완전히 다르게 해석될 수 있는 오묘한 책이다.', '09/27/2013'),  
(6, 9, '1.9', '차례의 구성이 이해하기 너무 어렵게 되어있다.', '08/17/2021'),  
(7, 10, '3.0', '', '09/02/2021'),  
(12, 15, '4.3', '', '12/22/2021'),  
(15, 3, '3.0', '', '10/22/2020')
```

➡ REVIEW TABLE 데이터 삽입 쿼리문

SQL-REVIEW TABLE

```
SELECT *  
FROM REVIEW;
```



REVIEW TABLE 데이터 조회 쿼리문

결과

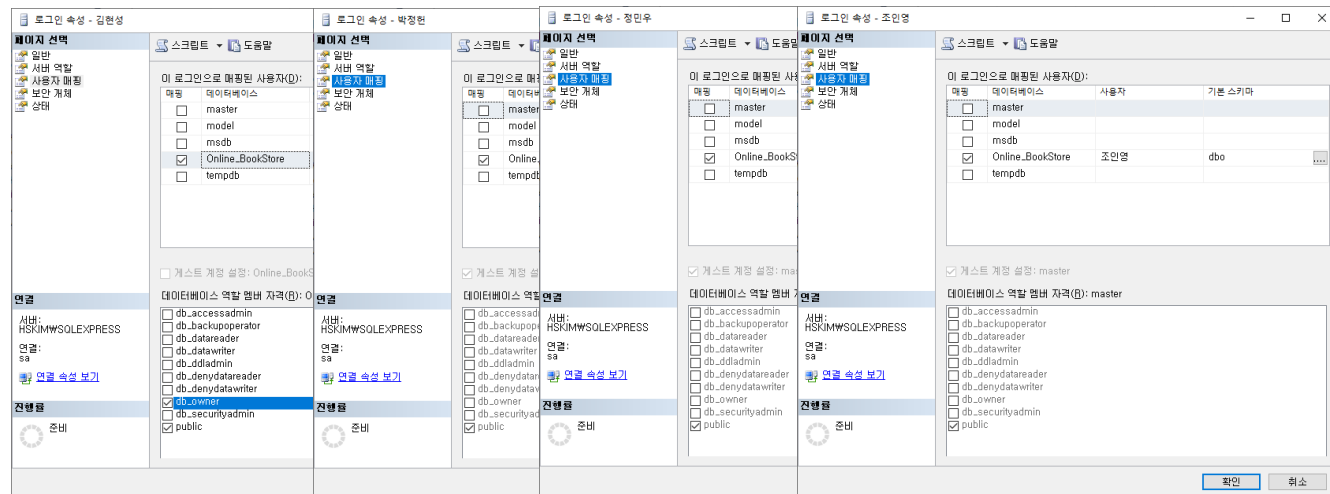
메시지

	OrderID	BookID	Grade	ReviewDetail	EnrollDate
1	1	4	3.6	그저 그런 책	2021-10-13
2	2	5	4.2	한 번쯤 읽어봐야 할 자기계발서이다	2021-11-23
3	3	6	4.2	배경지식을 쌓기에 유용한 도서라고 생각함!	2021-08-20
4	4	7	2.8	독자의 성향에 따라 완전히 다르게 해석될 수 있는 오묘한 책이다.	2013-09-27
5	5	8	3.2	장르와 관련이 있는 지 모르겠다	2021-04-12
6	6	9	1.9	차례의 구성이 이해하기 너무 어렵게 되어있다.	2021-08-17
7	7	10	3.0		2021-09-02
8	8	11	4.8	왜 베스트 셀러인지 첫 챕터에서 바로 알 수 있는 책	2021-07-08
9	9	12	2.6	돈이 아깝다. 읽지 마세요.	2021-11-12
10	10	13	3.5	누구나 한 번쯤 고민하는 것들을 잘 정리해서 쓴 책임. 추천함.	2021-07-10
11	11	14	4.0		2019-05-22
12	12	15	4.3		2021-12-22
13	13	1	4.9	서재에 보관하고 계속 읽어야 할 책인 것 같다.	2021-08-04
14	14	2	4.0	아이들에게도 읽게 해주면 좋을 만한 소재를 다룬 듯 하다.	2020-02-20
15	15	3	3.0		2020-10-22

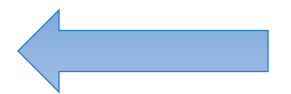
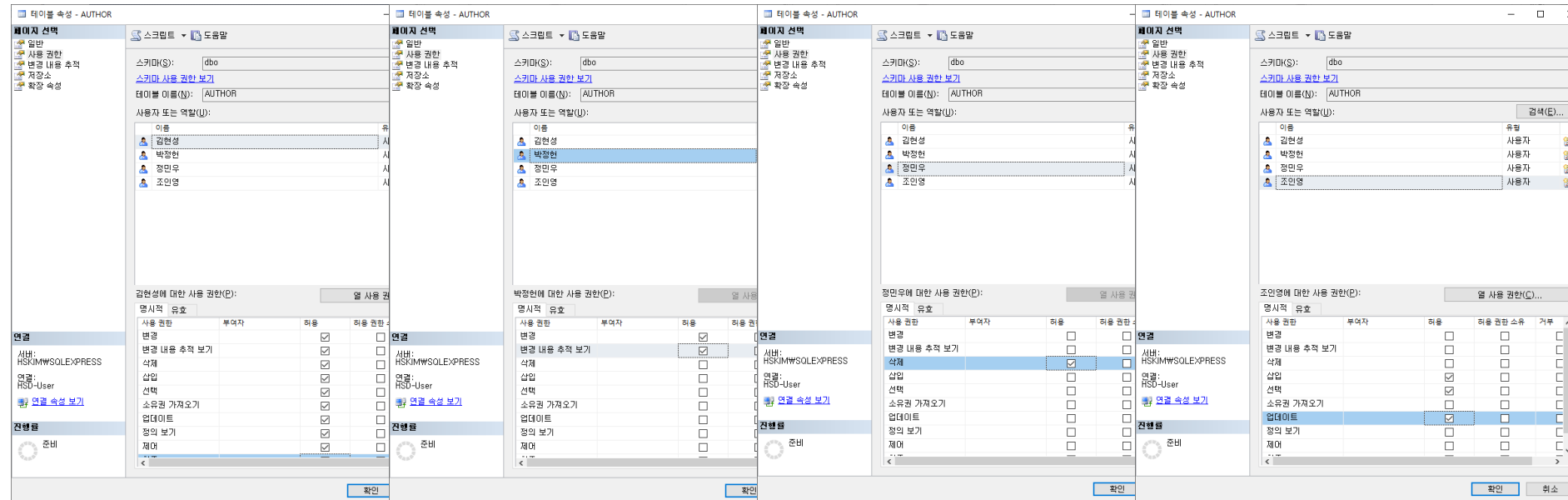


REVIEW TABLE 데이터 조회 결과

관리자 권한



4명의 사용자(김현성, 박정현, 정민우, 조인영)를 추가하고
 ‘김현성’에게만 DB_OWNER 권한을 주고
 나머지는 PUBLIC으로 설정하였다.
 추가로 테이블 별로
 김현성: 전체 권한
 박정현: 변경/ 변경 내용 추적 보기
 정민우: 삭제
 조인영: 선택 삽입 업데이트
 권한을 주었다.

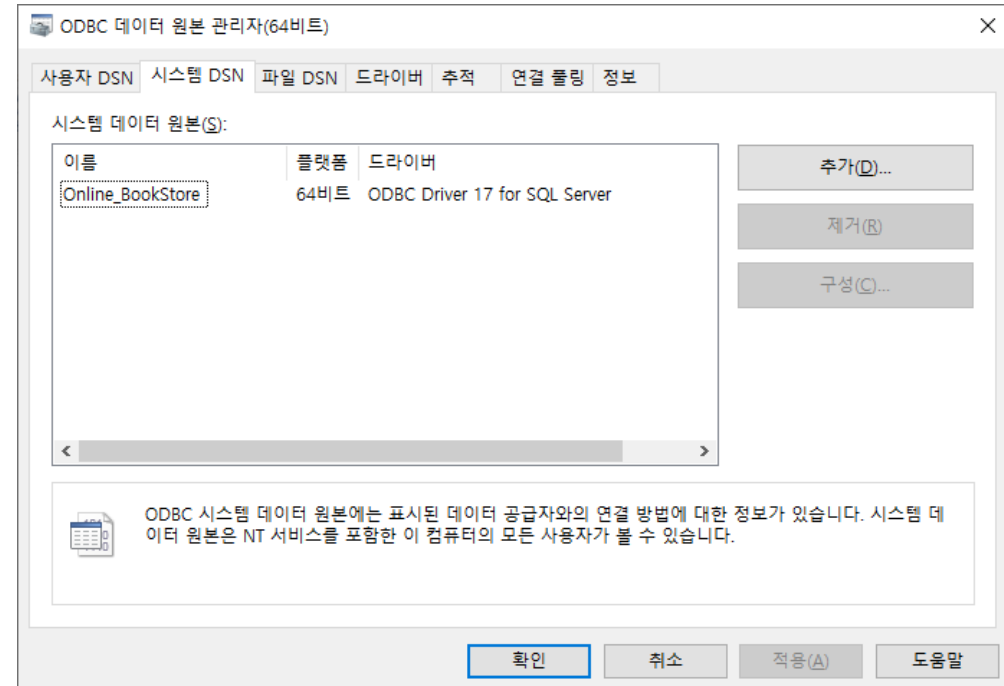
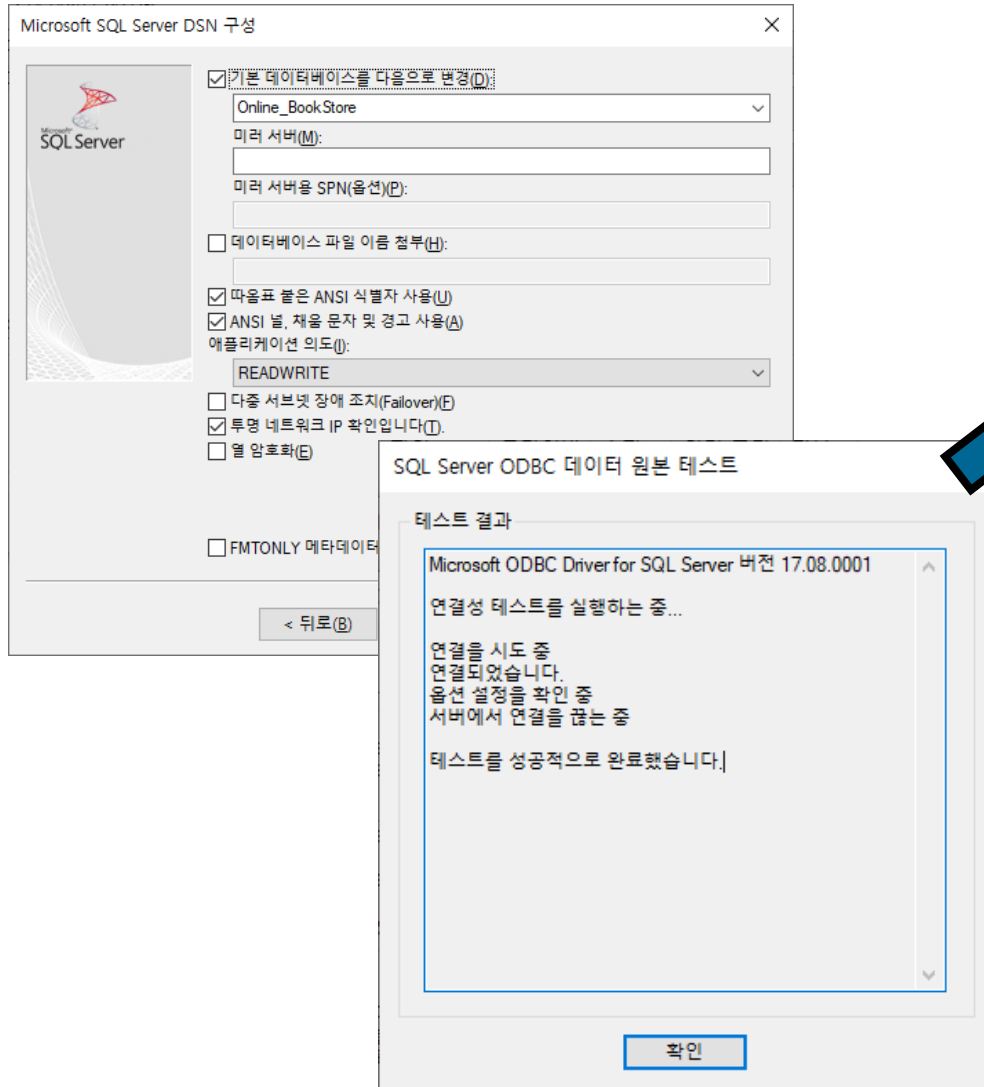


AUTHOR 테이블에
 먼저 설정한 모습이다.
 추후에 모든 테이블에
 동일하게 설정 해 주었음.

ASP

ODBC 연결부터 어플리케이션 구축까지

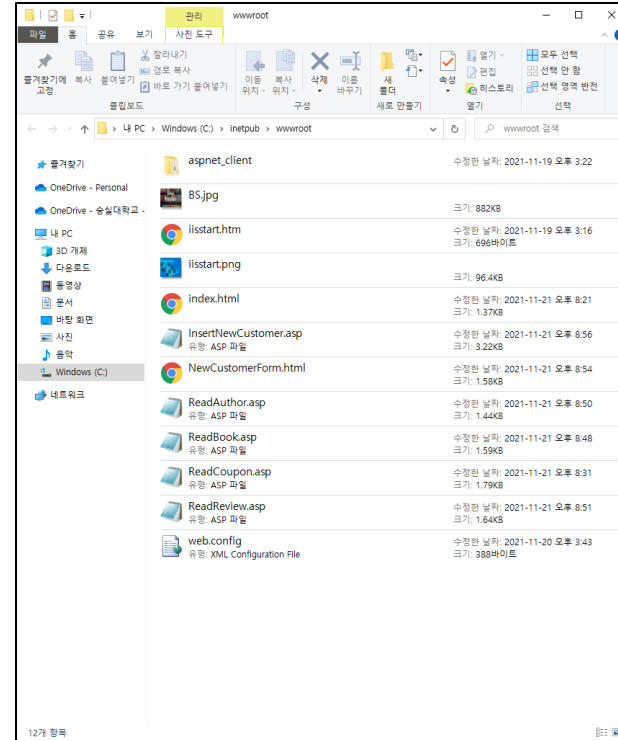
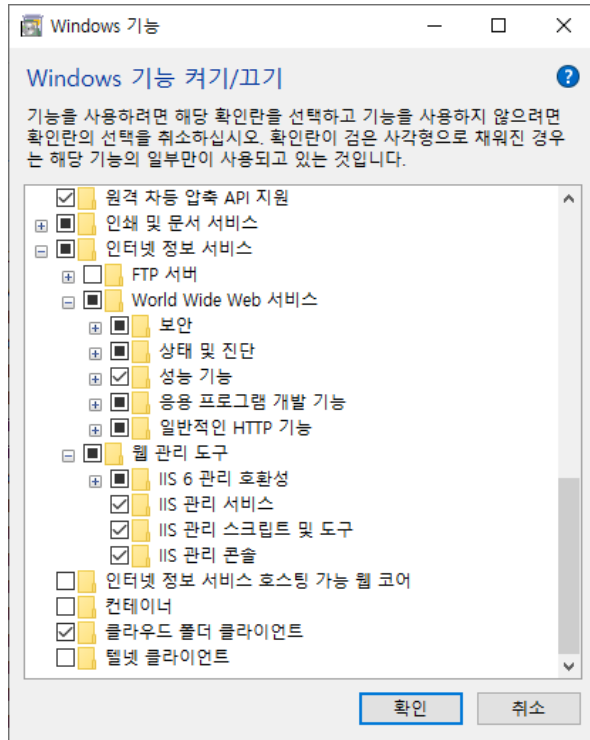
ODBC 연결



ODBC를 통해 SQL SERVER과 연결을 하고,
데이터 원본을 Online_Bookstore로 설정 한 뒤
데이터 원본 테스트를 성공적으로 완료 하였다.

시스템DSN에 추가된 모습을 확인 할 수 있다.

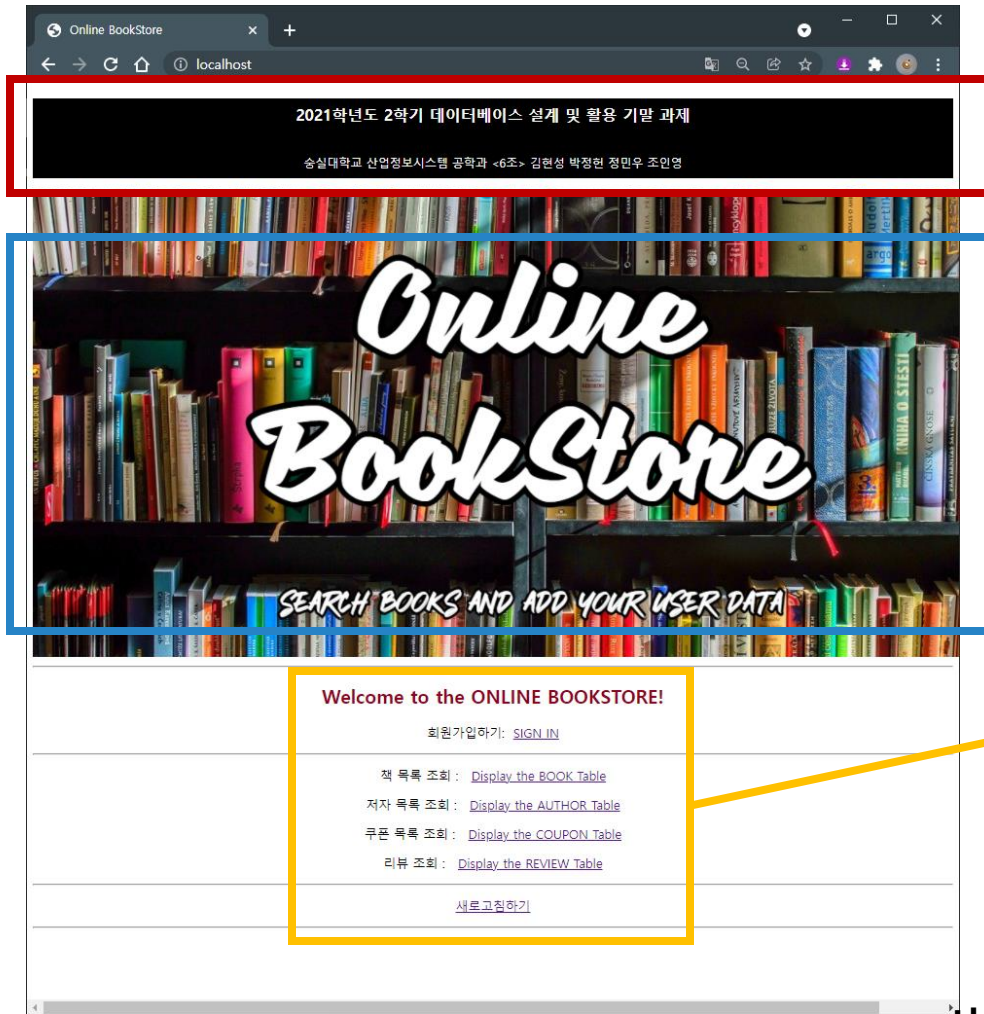
index.html



WINDOWS 기능에서
인터넷 정보 서비스 항목을 활성화하고,
메모장을 관리자 기능으로 실행하여 작성한

Asp, html 파일들을 inetpub/wwwroot 경로에 저장하였다.

index.html



```
*index.html - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!DOCTYPE html>
<html>
<head>
<title>Online BookStore</title>
<meta charset="UTF-8">
<div class="textBanner">
<style>
.textBanner { background:#000; text-align:center; }
.textBanner p { padding:6px 14px 5px; color:#fff; font-size:20px; line-height:30px; font-weight:bold; }
</style>
<p>2021학년도 2학기 데이터베이스 설계 및 활용 기말 과제</p>
<p style="text-align: center; font-size:15px; font-weight: normal">
숭실대학교 산업정보시스템 공학과
<6조> 김현성 박정현 정민우 조인영
</p>
</div>
<body>
<a href="https://iise.ssu.ac.kr/">
<center>
</a>
<hr />
<h2 style="text-align: center; color: #900020">
Welcome to the ONLINE BOOKSTORE!
</h2>
<p>회원가입하기:< a href="NewCustomerForm.html">
SIGN IN
</a>
</p>
<p>책 목록 조회:< a href="ReadBook.asp">
Display the BOOK Table
</a>
</p>
<p>저자 목록 조회:< a href="ReadAuthor.asp">
Display the AUTHOR Table
</a>
</p>
<p>쿠폰 목록 조회:< a href="ReadCoupon.asp">
Display the COUPON Table
</a>
</p>
<p>리뷰 조회:< a href="ReadReview.asp">
Display the REVIEW Table
</a>
</p>
<hr />
<p style="text-align: center; color: #900020">
<a href="">
새로고침하기
</a>
</p>
<hr />
</body>
</html>
```

상단 배너에 해당하는 코드이다.

```
<a href="https://iise.ssu.ac.kr/">
<center>
</a>
<hr />
<h2 style="text-align: center; color: #900020">
Welcome to the ONLINE BOOKSTORE!
</h2>
<p>회원가입하기:< a href="NewCustomerForm.html">
SIGN IN
</a>
</p>
```

```
*index.html - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<hr />
<h2>
<p>회원가입하기:< a href="NewCustomerForm.html">
SIGN IN
</a>
</p>
<hr />
<p>책 목록 조회:< a href="ReadBook.asp">
Display the BOOK Table
</a>
</p>
<p>저자 목록 조회:< a href="ReadAuthor.asp">
Display the AUTHOR Table
</a>
</p>
<p>쿠폰 목록 조회:< a href="ReadCoupon.asp">
Display the COUPON Table
</a>
</p>
<p>리뷰 조회:< a href="ReadReview.asp">
Display the REVIEW Table
</a>
</p>
<hr />
<p style="text-align: center; color: #900020">
<a href="">
새로고침하기
</a>
</p>
<hr />
</body>
</html>
```

사진과 사진을 눌렀을 때
이동하는 링크에 해당하는
코드이다.

NewCustomerForm.html

NewCustomerForm HTML Page x +

localhost/NewCustomerForm.html

회원가입

회원정보를 입력해 주십시오.

아이디: hskim127777

비밀번호: password

이름: 김원성

주소: 서울특별시 강남구

우편번호: 06330

휴대폰번호: 01075159231

Email: hskim1277@gmail.com

추천인:

[홈페이지로 돌아가기](#)

해당 Form에서 입력 받은
정보를
InsertNewCustomer.asp로
연결하는 코드

```

NewCustomerForm.html - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<html>
<head>
<meta charset="UTF-8">
<title>NewCustomerForm HTML Page</title>
<style type="text/css">
h1 {text-align: center; color: #900020}
h2 {font-family: Arial, sans-serif; text-align: left; color: #900020}
p.footer {text-align: center}
</style>
</head>
<body>
<form action="InsertNewCustomer.asp" method="POST">
<!-- Page Headers -->
<h1>회원가입</h1>
<hr />
<br />
<p>
<center> <b>회원정보를 입력해 주십시오. </b>
</p>
<table>
<tr>
<td>&nbsp;&nbsp;&nbsp;&아이디:&nbsp;&nbsp;&nbsp;&</td>
<td><input type="text" name="CustomerID" size="20" /></td>
</tr>
<tr>
<td>&nbsp;&nbsp;&비밀번호:&nbsp;&nbsp;&nbsp;&</td>
<td><input type="text" name="CustomerPW" size="20" /></td>
</tr>
<tr>
<td>&nbsp;&nbsp;&이름:&nbsp;&nbsp;&nbsp;&</td>
<td><input type="text" name="CustomerName" size="20" /></td>
</tr>
<tr>
<td colspan="2">
<input type="submit" value="회원가입하기" />
<input type="reset" value="초기화" />
</td>
</tr>
</table>
<p class="footer">
<a href="index.html">
홈페이지로 돌아가기
</a>
</p>
</body>
</html>

```

InsertNewCustomer.asp

회원가입 장

localhost/InsertNewCustomer.asp

회원가입이 성공적으로 완료되었습니다.

추가된 회원 정보

CustomerID:	hskim127744
CustomerName:	김현성
CsAddress:	서울특별시 강남구 개포로
CsZIP:	06330

ONLINE BOOKSTORE 회원 리스트

회원 아이디	회원 이름	회원 주소	우편번호	회원 전화번호	회원 E-mail	추천인
gihun2	이기훈	대전광역시 유성구 대덕대로 523	34131	010-1145-6789	gihun12@naver.com	gihun2
hong223	홍가진	서울특별시 강남구 선릉로 8길 19	06325	010-1117-3456	gajin12@gmail.com	hong223
hongildong123	홍길동	경기도 성남시 분당구 신수로 791	13635	010-1112-3456	honggil@naver.com	hongildong123
hskim127744	김현성	서울특별시 강남구 개포로	06330	01075159231	hskim1277@gmail.com	
hyungsuk12	이형석	서울특별시 강남구 테헤란로 6길9	06233	010-1118-3456	hyungsuk@naver.com	hyungsuk12
ina89	홍이나	인천광역시 강화군 대월로 12-2	23025	010-1119-3456	ina123@naver.com	ina89
jhee103	박지희	경기도 파주시 김산로 173번길 78	10848	010-1116-3456	jhee1@gmail.com	jhee103
jiho99	박지호	대전광역시 대덕구 대청호수로 1533	34316	010-1140-3456	jiho12@naver.com	jiho99
minho89	장민호	부산광역시 해운대구 반송로 886	48000	010-1138-3458	minho34@gmail.com	minho89
minsu88	김민수	서울특별시 성북구 길음로 92	02724	010-1135-3457	minsu12@naver.com	minsu88
misun182	박미선	서울특별시 강서구 양천로 55길 55	07528	010-1122-3457	misun12@naver.com	misun182
snk2020	김선남	경기도 성남시 분당구 탄현상로 95	13637	010-1113-3457	sunnam1@gmail.com	snk2020
somang88	이소망	경기도 용인시 기흥구 원곡대로 56-8	17086	010-1146-6789	somang12@gmail.com	somang88
sunh2201	곽선희	경기도 수원시 영통구 억영대로 1483번길 16-4	16680	010-1114-3458	sunhee12@naver.com	sunh2201
yiul122	이율리	경기도 수원시 권선구 평통로 52-1	16601	010-1115-3456	leewool12@naver.com	yiul122
yunghee103	김영희	인천광역시 동구 급곡로 62	22562	010-1110-3456	younghee12@gmail.com	yunghee103

[홈페이지로 돌아가기](#)

HSKIMWSQLEXPRESS.Online_BookStore - dbo.CUSTOMER - Microsoft SQL Server Management Studio

파일(F) 편집(E) 보기(V) 프로젝트(P) 디버깅(D) 쿼리 디자인(Q) 도구(T) 창(W) 도움말(H)

개체 탐색기

HSKIMWSQLEXPRESS -> dbo.CUSTOMER

CustomerID	CustomerPW	CustomerName	CsAddress	CsZIP	CustomerPhone	CustomerEmail	ReferredBy
gihun2	kh13909	이기훈	대전광역시 유...	34131	...	gihun12@nave...	gihun2
hong223	hgi321	홍가진	서울특별시 강...	06325	...	gajin12@gmail...	hong223
hongildong123	hgd123	홍길동	경기도 성남시 ...	13635	...	honggil@naver...	hongildong123
hskim127744	password	김현성	서울특별시 강...	06330	...	hskim1277@g...	
hyungsuk12	lhs345	이형석	서울특별시 강...	06233	...	hyungsuk@nav...	hyungsuk12
ina89	hina123	홍이나	인천광역시 강...	23025	...	lina123@naver...	ina89
jhee103	pjh123	박지희	경기도 파주시 ...	10848	...	jhee1@gmail.c...	jhee103
jiho99	jhpp2323	박지호	대전광역시 대...	34316	...	jiho12@naver...	jiho99
minho89	mjh53	장민호	부산광역시 해...	48000	...	minho34@gm...	minho89
minsu88	msms23	김민수	서울특별시 성...	02724	...	minsu12@nav...	minsu88
misun182	msp3232	박미선	서울특별시 강...	07528	...	misun12@nav...	misun182
snk2020	snk123	김선남	경기도 성남시 ...	13637	...	sunnam1@gm...	snk2020
somang88	smsm12323	이소망	경기도 용인시 ...	17086	...	somang12@g...	somang88
sunh2201	ksh123	곽선희	경기도 수원시 ...	16680	...	sunhee12@nav...	sunh2201
yiul122	liul123	이율리	경기도 수원시 ...	16601	...	leewool12@na...	yiul122
yunghee103	yh122	김영희	인천광역시 동...	22562	...	younghee12@...	yunghee103
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

추가한 정보까지 출력된 회원 리스트에서 확인 할 수 있다.

원본 DATABASE에도 역시 성공적으로 추가된 것을 확인 할 수 있다.

InsertNewCustomer.asp

```
<%
CustomerID = Request.Form("CustomerID")
CustomerPW = Request.Form("CustomerPW")
CustomerName =
Request.Form("CustomerName")
CsAddress = Request.Form("CsAddress")
CsZIP = Request.Form("CsZIP")
CustomerPhone =
Request.Form("CustomerPhone")
CustomerEmail =
Request.Form("CustomerEmail")
ReferredBy = Request.Form("ReferredBy")
Dim dbConCmd, dbCon, rs, rsList
dbConCmd= "DSN=Online_BookStore;
Database=Online_BookStore; UID=HSD-User;
PWD= 123456;"
Set dbCon =
Server.CreateObject("ADODB.Connection")
dbCon.Open dbConCmd
SQL = "INSERT INTO
CUSTOMER(CustomerID,
CustomerPW,CustomerName,
CsAddress, CsZIP, CustomerPhone,
CustomerEmail, ReferredBy)
VALUES "
SQL = SQL & "(" & CustomerID & ""
SQL = SQL & "," & CustomerPW & ""
SQL = SQL & "," & CustomerName & ""
SQL = SQL & "," & CsAddress & ""
SQL = SQL & "," & CsZIP & ""
SQL = SQL & "," & CustomerPhone & ""
SQL = SQL & "," & CustomerEmail & ""
SQL = SQL & "," & ReferredBy & """)"
dbCon.Execute SQL
%>
<%
Session.CodePage="65001"
Response.CharSet="utf-8"
%>
```

```
<h1><center>회원가입이 성공적으로
완료 되었습니다.</h1>
<hr />
<h2><center>추가된 회원 정보</h2>
<center><table>
<%Response.Write "<tr>"
Response.Write "<td>CustomerID:</td>"
Response.Write "<td>"
Response.Write(CustomerID)
Response.Write "</td>"
Response.Write "<tr>"
Response.Write "<td>CustomerName:</td>"
Response.Write "<td>"
Response.Write(CustomerName)
Response.Write "</td>"
Response.Write "</tr>"
Response.Write "<tr><td>CsAddress:</td>"
Response.Write "<td>"
Response.Write(CsAddress)
Response.Write "</td>"
Response.Write "</tr>"
Response.Write "<tr><td>CsZIP:</td>"
Response.Write "<td>"
Response.Write(CsZIP)
Response.Write "</td>"
Response.Write "</tr>"
%>
```

```
</table><hr />
<h1>ONLINE BOOKSTORE 회원 리스트</h1>
<hr />
<center><table>
<tr>
<th>회원 아이디</th>
<th>회원 이름</th>
<th>회원 주소</th>
<th>우편번호</th>
<th>회원 전화번호</th>
<th>회원 E-mail</th>
<th>추천인</th>
</tr>
<%
Set rs = Server.CreateObject("ADODB.RecordSet")
SQLQuery = "SELECT CustomerID,
CustomerName, CsAddress, CsZIP,
CustomerPhone, CustomerEmail,
ReferredBy FROM CUSTOMER "
rs.Open SQLQuery, dbCon
```

<- 한글이 깨지는 문제를 방지하기 위해 해당 코드를 삽입하였다.

ReadBook.asp

책 조회 결과

ONLINE BOOKSTORE

책 목록

책 ID	책 이름	출간 년도	작가 이름	가격/th>	재고	페이지 수	EBook 여부	출판사
1	트렌드 코리아 2022	2021-01-01	김지수	16200	5	452	YES	미래의창
2	피아니스트	2020-01-01	김지수	13500	1	380	YES	좋은땅
3	유희 도시 기행	2019-01-01	유시민	14850	2	324	YES	생각의 길
4	바보	2013-01-01	임종현	10800	0	248	NO	지식과 감성
5	행성어 서점	2021-01-01	김초엽	13050	1	216	YES	마음산책
6	작별하지 않는다	2021-01-01	한강	12600	2	332	NO	문학동네
7	180일간의 트럭터 다이어리	2011-01-01	강기태	9100	0	200	YES	북스톤
8	작은 별이지만 빛나고 있어	2021-01-01	소운	13500	1	274	YES	북로망스
9	NFT 레플루션	2021-01-01	성소라	16200	2	324	YES	더퀘스트
10	인생의 답은 내 안에 있다	2021-01-01	김이섭	11060	0	300	NO	마디어슈
11	그냥 하지 말라	2021-01-01	송길영	15300	3	284	YES	북스톤
12	사이다 공식으로 푼다는 글쓰기 방법	2021-01-01	김주리	13500	4	272	YES	윙크북스
13	세계미래보고서 2021	2020-01-01	박영숙	16020	1	372	YES	비즈니스북스
14	오운영의 화해	2019-01-01	오운영	14400	3	320	YES	코리아닷컴
15	픽키	2021-01-01	김도윤	14400	2	280	YES	북로망스

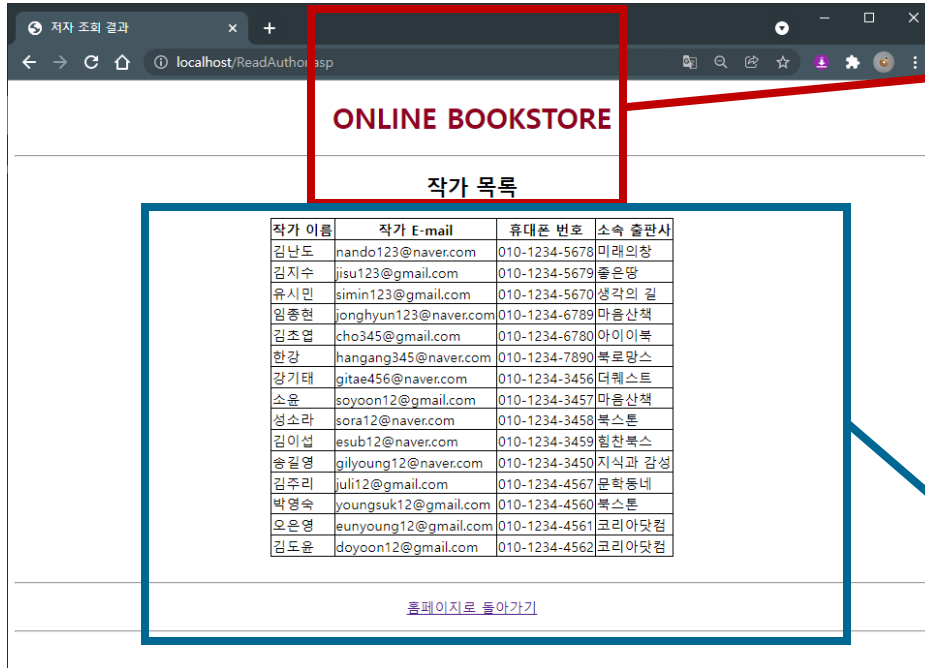
[홈페이지로 돌아가기](#)

```
ReadBook.asp - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>책 조회 결과</title>
<style type="text/css">
h1 {text-align: center; color: #900020}
h2 {font-family: Ariel, sans-serif; text-align: left; color: black}
p.footer {text-align: center}
table, th, td {
border: 1px solid black;
border-collapse: collapse;
}
</style>
</head>
<body>
<h1>ONLINE BOOKSTORE</h1>
<hr />
<h2><center>책 목록</center></h2>
<%
Dim dbConCmd, SQLQuery, dbCon, rs, rsList
dbConCmd = "DSN=Online_BookStore; Database=Online_BookStore; UID=HSD-User; PWD=123456;"
SQLQuery = "SELECT BookID, BookName, PubYear, AUTHOR.AuthorName, Price, Inventory, Pages, Ebook, BOOK.PublisherName FROM BOOK INNER JOIN AUTHOR ON AUTHOR.AuthorID = BOOK.AuthorID"
Set dbCon = Server.CreateObject("ADODB.Connection")
Set rs = Server.CreateObject("ADODB.RecordSet")
dbCon.Open dbConCmd
rs.Open SQLQuery, dbCon
%>
<center><table>
<tr>
<th>책 ID</th> <th>책 이름</th> <th>출간 년도</th> <th>작가 이름</th> <th>가격</th> <th>재고</th> <th>페이지 수</th> <th>EBook 여부</th> <th>출판사</th>
</tr>
<%
IF (rs.EOF AND rs.EOF) THEN
Response.Write("책 정보가 없습니다.")
ELSE
rsList = rs.GetRows()
FOR x=0 TO UBound(rsList, 2)
Response.Write(" <tr>")
FOR y=0 TO UBound(rsList, 1)
Response.Write(" <td>")
Response.Write(rsList(y, x))
Response.Write(" </td>")
NEXT
Response.Write(" </tr>")
NEXT
Response.Write(" </table>")
END IF
%>
<br />
<hr />
<p class="footer">
<a href="index.html"> 홈페이지로 돌아가기 </a>
</p>
<hr />
</body>
</html>
```

BOOK table과 AUTHOR table을
INNER JOIN 하여

책 이름과 작가 이름을 동시에 출력하는 코드를 사용하였다.

ReadAuthor.asp



```
ReadAuthor.asp - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>저자 조회 결과</title>
<style type="text/css">
h1 {text-align: center; color: #900020}
h2 {font-family: Arial, sans-serif; text-align: left; color: black}
p.footer {text-align: center}
table, th, td {
border: 1px solid black;
border-collapse: collapse;
}
</style>
</head>
<body>
<h1>ONLINE BOOKSTORE</h1>
<hr />
<h2><center>작가 목록</h2>
<%
Session.CodePage= "65001"
Response.CharSet= "utf-8"
%>
<%
Dim dbConCmd, SQLQuery, dbCon, rs, rsList
dbConCmd= "DSN=Online_BookStore; Database=Online_BookStore; UID=HSD-User; PWD=123456;
SQLQuery = "SELECT AuthorName, AuthorEmail, AuthorPhone, PublisherName FROM AUTHOR"
Set dbCon = Server.CreateObject("ADODB.Connection")
Set rs = Server.CreateObject("ADODB.RecordSet")
dbCon.Open dbConCmd
rs.Open SQLQuery, dbCon
%>
<center><table>
<tr>
<th>작가 이름</th> <th>작가 E-mail</th> <th>휴대폰 번호</th> <th>소속 출판사</th>
</tr>
<%
IF (rs.BOF AND rs.EOF) THEN
Response.Write("작가정보가 존재하지 않습니다.")
ELSE
rsList = rs.GetRows()
FOR x=0 TO UBound(rsList, 2)
Response.Write(" <tr>")
FOR y=0 TO Ubound(rsList, 1)
Response.Write(" <td>")
Response.Write(rsList(y, x))
Response.Write(" </td>")
NEXT
Response.Write(" </tr>")
NEXT
Response.Write(" </table>")
END IF
%>
<br />
<hr />
<p class="footer">
<a href="index.html"> 홈페이지로 돌아가기</a>
</p>
<hr />
</body>
</html>
```


ReadCoupon.asp

쿠폰 조회 결과

localhost/ReadCoupon.asp

ONLINE BOOKSTORE

쿠폰 리스트

고객 ID	고객 이름	쿠폰 이름	할인율	유효 기간
hongildong123	홍길동	크리스마스 기념 이벤트	10%	2021-01-07
snk2020	김선남	10주년 기념 이벤트	15%	2020-12-22
sunh2201	곽선희	15000원 이상 구매 시 할인	5%	2021-04-21
yiuli22	이율리	20000원 이상 구매 시 할인	10%	2021-04-21
jhee103	박지희	30000원 이상 구매 시 할인	15%	2021-04-21
honga223	홍가진	실버 회원 우대	15%	2020-11-12
hyungsuk12	이형석	골드 회원 우대	20%	2021-08-17
ina89	홍이나	신학기 대학생 지원 이벤트	20%	2018-06-09
yunghee103	김영희	수험생을 위한 선물	30%	2020-03-05
misun182	박미선	신규 가입자 환영	25%	2019-02-12
minsu88	김민수	1년 이상 장기 고객 우대	15%	2018-07-16
minho89	장민호	즐거운 한가위는 독서와 함께!	5%	2019-11-15
jiho99	박지호	새해 복 많이 받으세요!	15%	2021-03-01
gihun2	이기훈	50000원 이상 구매 시 할인	20%	2020-03-11
somang88	이소망	할로윈 데이 깜짝 이벤트	10%	2021-11-30

[홈페이지로 돌아가기](#)

**BOOK table과 AUTHOR table을
INNER JOIN 하여**

책 이름과 작가 이름을 동시에 출력하는 코드를 사용하였다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>쿠폰 조회 결과</title>

</head>
<body>
<h1>ONLINE BOOKSTORE</h1>
<hr />
<h2><center>쿠폰 리스트</center></h2>

</body>
</html>

Session.CodePage="65001"
Response.CharSet="utf-8"
%>
<%
Session.CodePage = 949
Response.CharSet = "euc-kr"
Response.AddHeader "Pragma","no-cache"
Response.AddHeader "cache-control", "no-staff"
Response.Expires = -1
%>
<%
Session.CodePage="65001"
Response.CharSet="utf-8"
%>
<%
dbConn = Server.CreateObject("ADODB.Connection")
dbConn.Open "DSN=Online_BookStore; Database=Online_BookStore; UID=HSD-User; PWD=123456;"
SQLQuery = "SELECT CT.CustomerID, CustomerName, CouponName, DiscountRate, Validity FROM (CUSTOMER CT INNER JOIN COUPONLIST CL ON CT.CustomerID = CL.CustomerID) INNER JOIN COUPON CP ON CL.CouponID = CP.CouponID"
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open SQLQuery, dbConn
%>
<center><table>
<tr>
<th>고객 ID</th>
<th>고객 이름</th>
<th>쿠폰 이름</th>
<th>할인율</th>
<th>유효 기간</th>
</tr>
<%
IF (rs.EOF AND rs.EOF) THEN
Response.Write("쿠폰정보가 존재하지 않습니다.")
ELSE
rsList = rs.GetRows()
FOR x=0 TO UBound(rsList, 2)
Response.Write("<tr>")
FOR y=0 TO Ubound(rsList, 1)
Response.Write("<td>")
Response.Write(rsList(y, x))
Response.Write("</td>")
NEXT
Response.Write("</tr>")
NEXT
Response.Write("</table>")
END IF
%>
<br />
<hr />
<p class="footer">
<a href="localhost"> 홈페이지로 돌아가기</a>
</p>
</body>
</html>
```

ReadReview.asp

리뷰 결과 조회

ONLINE BOOKSTORE

리뷰 목록

책 ID	책 이름	주문 ID	고객 ID	평점	리뷰 상세 내용	리뷰 작성일
4	바보	1	hongildong123	3.6	그저 그런 책	2021-10-13
5	행성어 서점	2	snk2020	4.2	한 번쯤 읽어봐야 할 자기계발서이다	2021-11-23
6	작별하지 않는다	3	sunh2201	4.2	배경지식을 쌓기에 유용한 도서라고 생각함!	2021-08-20
7	180일간의 트랙터 다이어리	4	yiuli22	2.8	독자의 성향에 따라 완전히 다르게 해석될 수 있는 오묘한 책이다.	2013-09-27
8	작은 별이지만 빛나고 있어	5	jhee103	3.2	장르와 관련이 있는 지 모르겠다	2021-04-12
9	NFT 레플루션	6	honga223	1.9	차례의 구성이 이해하기 너무 어렵게 되어있다.	2021-08-17
10	인생의 답은 내 안에 있다	7	hyungsuk12	3		2021-09-02
11	그냥 하지 말라	8	ina89	4.8	왜 베스트 셀러인지 첫 접점에서 바로 알 수 있는 책	2021-07-08
12	사이다 공식으로 톡 쓰는 글쓰기 비법	9	yunghee103	2.6	돈이 아깝다. 읽지 마세요.	2021-11-12
13	세계미래보고서 2021	10	misun182	3.5	누구나 한 번쯤 고민하는 것들을 잘 정리해서 쓴 책임. 추천함.	2021-07-10
14	오은영의 화해	11	minsu88	4		2019-05-22
15	럭키	12	minho89	4.3		2021-12-22
1	트랜드 코리아 2022	13	jih99	4.9	서재에 보관하고 계속 읽어야 할 책인 것 같다.	2021-08-04
2	피아니스트	14	gihun2	4	아이들에게도 읽게 해주면 좋을 만한 소재를 다룬 듯 하다.	2020-02-20
3	유럽 도시 기행	15	somang88	3		2020-10-22

[홈페이지로 돌아가기](#)

```
*ReadReview.asp - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>리뷰 결과 조회</title>
</head>
<body>
<h1>ONLINE BOOKSTORE</h1>
<hr />
<h2><center>리뷰 목록</center></h2>
</body>
</html>

Session.CodePages="65001"
Response.CharSet="utf-8"
%>
<%
Dim dbConCmd, SQLQuery, dbCon, rs, rsList
SQLQuery = "SELECT RV.BookID,BOOK.BookName, OD.OrderID,CT.CustomerID,Grade,ReviewDetail.EnrollDate FROM
(CUSTOMER,CT INNER JOIN ORDERS OD ON CT.CustomerID = OD.CustomerID) INNER JOIN REVIEW RV ON OD.OrderID = RV.OrderID)INNER JOIN BOOK ON RV.BookID = BOOK.BookID"
Set dbCon = Server.CreateObject("ADODB.Connection")
Set rs = Server.CreateObject("ADODB.RecordSet")
dbCon.Open dbConCmd
rs.Open SQLQuery, dbCon
%>
<center><table>
<tr>
<th>책 ID</th> <th>책 이름</th> <th>주문 ID</th> <th>고객 ID</th> <th>평점</th> <th>리뷰 상세 내용</th> <th>리뷰 작성일</th>
</tr>
<%
IF (rs.EOF AND rs.EOF) THEN
Response.Write("리뷰가 없습니다.")
ELSE
rsList = rs.GetRows()
FOR x=0 TO UBound(rsList, 2)
Response.Write("<tr>")
FOR y=0 TO UBound(rsList, 1)
Response.Write("<td>")
Response.Write(rsList(y, x))
Response.Write("</td>")
NEXT
Response.Write("</tr>")
NEXT
Response.Write("</table>")
END IF
%>
<br />
<hr />
<p class="footer">
<a href="index.html">홈페이지로 돌아가기</a>
</p>
</body>
</html>
```

CUSTOMER,ORDERS,REVIEW,BOOK table을
INNER JOIN 하여

책/주문/고객/리뷰에 대한 정보를
한번에 보여주는 코드를 사용했다.

감사합니다!