

ICE4016 데이터베이스설계

<final 설계 프로젝트>

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2022년 12 월 18 일

학부 정보통신공학

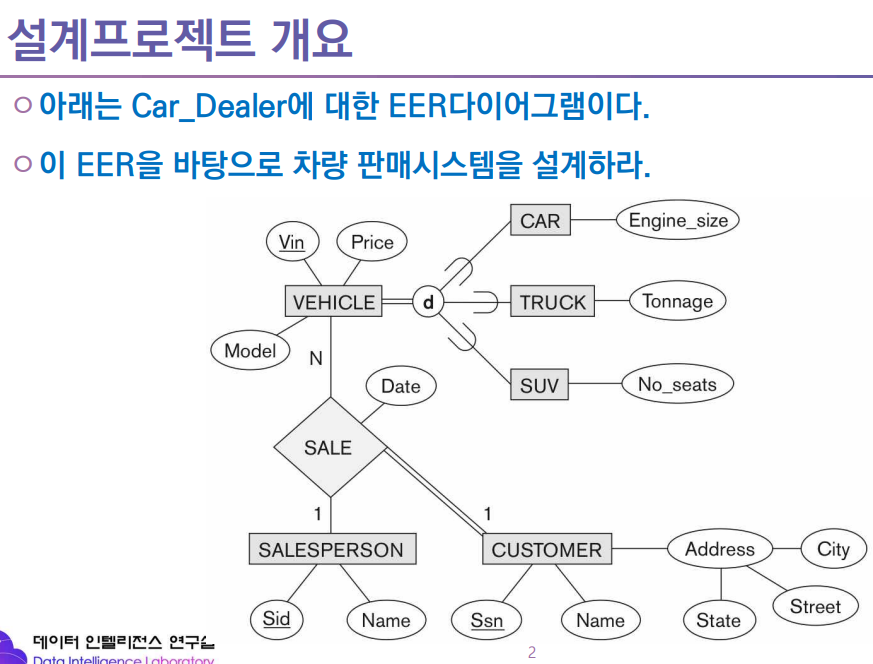
학년 3

성명 이해수

학번 12201942



1. 개요

텍스트이(가) 표시된 사진

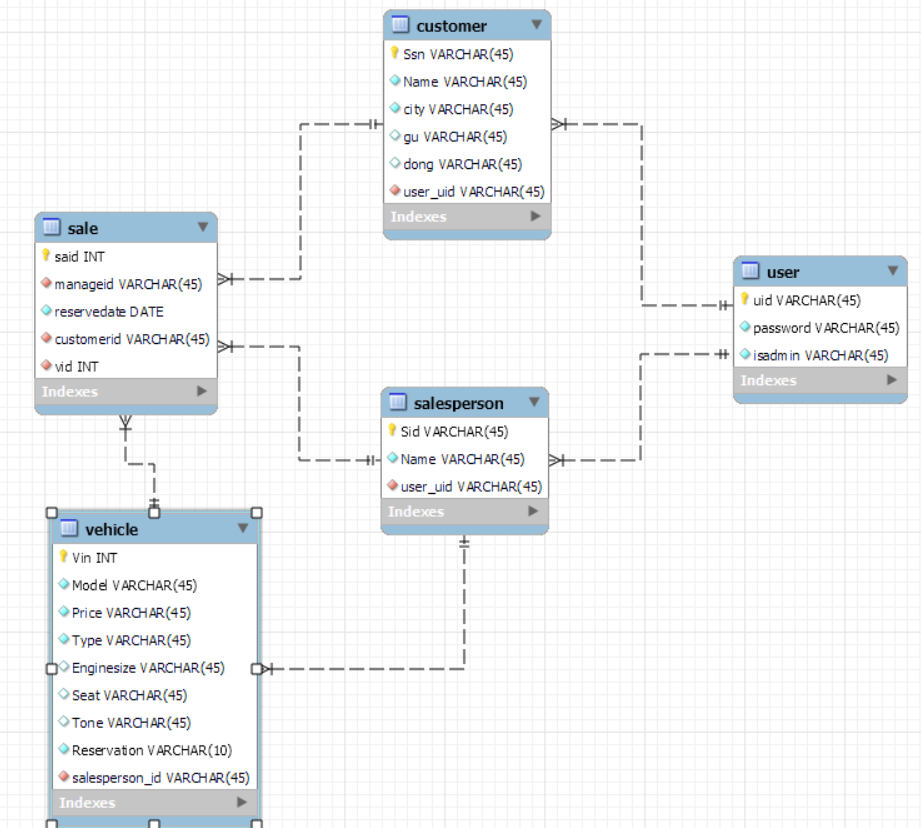
자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 상세 설계 내용

[ERD]



다음과 같이 ER Diagram을 구성하였습니다. 5개의 table이 존재합니다.

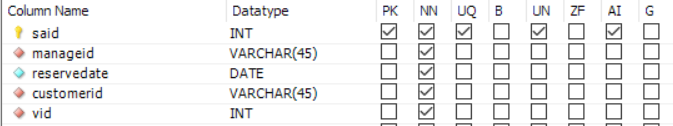
1. Vehicle

테이블이(가) 표시된 사진

자동 생성된 설명

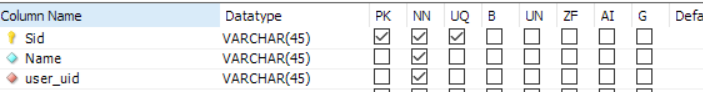
문제에서 주어진 disjoint를 만족하는 조건을 생각해보다 3가지의 차량 정보를 한 테이블에 담도록 하였습니다. 차량 각각에는 엔진 사이즈, 좌석 수, 적재 중량 특성 모두 존재하므로 모두 해당 테이블의 attribute로 지정하였고, Type 이라는 attribute 가 Car, Truck, SUV 종류를 나누는 특성을 가집니다. 또한 이미 예약된 vehicle은 다른 customer에게 보이지 않아야 하므로 Reservation이라는 attribute를 추가하여 이를 이용하여 예약된 차량인지에 대한 정보를 결정할 것입니다.

1. Sale



customer와 salesperson 간의 예약에 대한 data를 담는 sale 테이블입니다. 이는 예약일자와, 차량에 대한 id를 foreign key로 받아오고, 예약자 (customer) 아이디와 해당 예약 관리자 (salesperson) 아이디를 담습니다.

1. Salesperson



관리자 table에는 관리자의 user id가 존재합니다.

1. Customer

텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명

고객의 정보를 담는 테이블입니다. 고객 정보와 로그인에 필요한 user id가 존재합니다.

1. User

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

고객과 관리자 모두 웹에서 로그인을 해야하므로 이에 대한 정보가 담긴 테이블입니다. 관리자인지에 대한 여부를 확인하는 isadmin 애트리뷰트가 존재합니다. 로그인 시에 이를 이용하여 관리자와 고객을 판별할 것입니다.

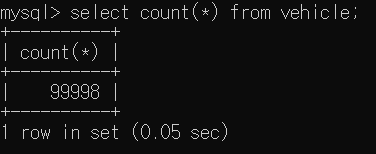
[10만개의 차량 데이터 생성]

우선 차량 타입이 Car, SUV, Truck 으로 나뉘어져 있는데, 차에 대해서 잘 모르기 때문에 9가지 정도 조사하였습니다. 또한 관리자 (판매자) 를 5명으로 설정하여 10만개를 5명이서 나눠 담당하는 것으로 설정하였습니다. 이를 위하여 파이썬을 이용하여 랜덤하게 생성하도록 코드를 작성하였습니다.

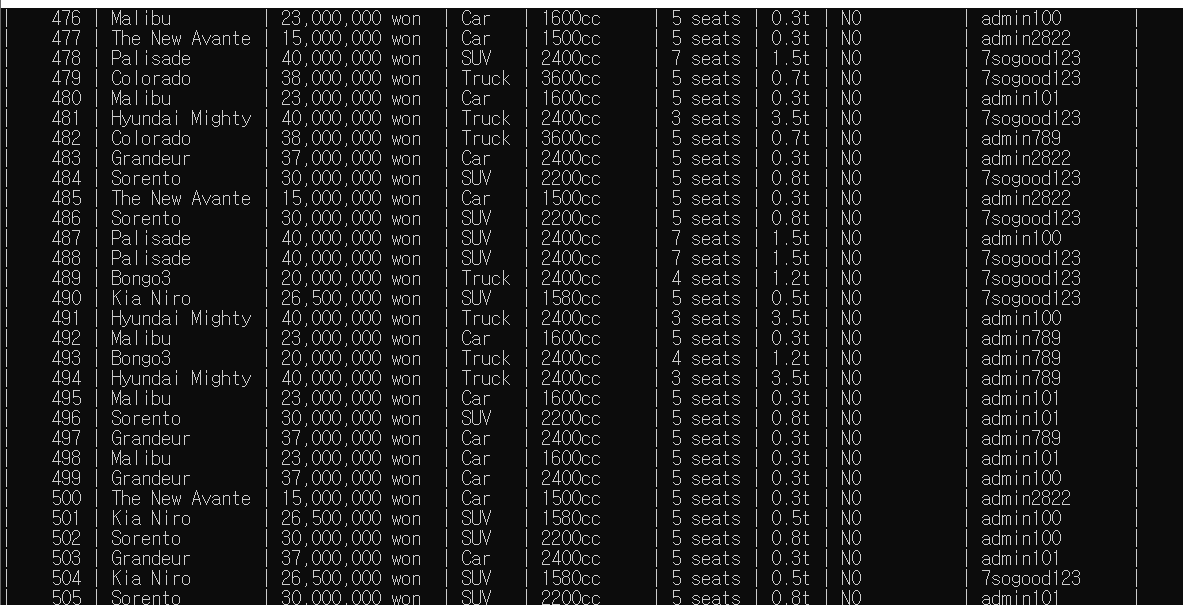
텍스트이(가) 표시된 사진

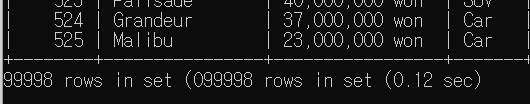
자동 생성된 설명

다음과 같이 9개의 차량을 담은 리스트에서 5명의 관리자 정보를 담은 리스트를 각각 랜덤하게 뽑아 만든 insert query를 담은 sql문을 생성하고 이를 source 명령어를 이용하여 mysql 내에서 insert 하였습니다.



테스트를 위해 몇 개의 데이터를 delete 한 것을 제외하면 랜덤 데이터로 모두 채워진 것을 알 수 있었습니다.





대략 select 해보면 랜덤하게 들어간 것을 알 수 있습니다.

[인덱스 생성]

텍스트, 검은색, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 관리자에 따라 차량 정보를 검색하는 경우가 많으므로 salesperson\_id에 인덱스를 걸어준 모습입니다. 데이터가 가장 많은 vehicle 테이블에 걸어주었습니다.

[sql] C,R,U,D 를 맡음

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

//read를 담당하는 selectSql 입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

//차량과 예약을 삭제하는 deletesql 입니다

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

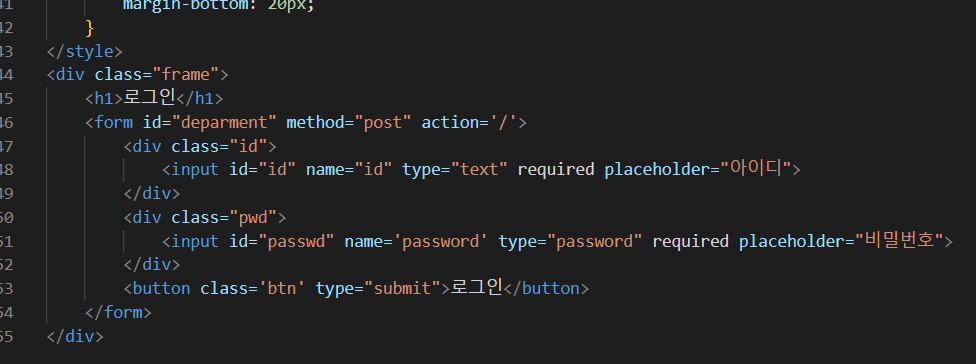
차량과 예약을 추가하는 insertsql 입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

차량과 예약 정보를 수정하거나, 예약이 완료되거나 예약이 안 된 상태로 vehicle 내의 reservation atrribute 값을 수정하는 코드들이 담긴 updatesql 입니다.

[login 코드]



//login.hbs

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명//login.js

우선 로그인을 수행하는 코드는 실습을 참고하였습니다. 이 때 구현해주어야 할 기능은 관리자와 고객을 구분해야하는 것이었습니다. 이를 위해서 우선 로그인에 필요한 유저 테이블을 생성하고, 테이블 내에 아이디 비밀번호 외에 관리자 여부를 묻는 attribute를 추가하였습니다. login.js 에서 isadmin= ‘YES

인지 ‘NO’인지에 따라 매니저와 고객을 구분짓고 고객이면 /select로 이동하여 자신의 예약 정보와 예약이 가능한 차량들에 대해서만 확인할 수 있도록 할 것이고 관리자면 /manage로 이동하도록 하고 해당 화면에서 내가 관리하는 차량과 예약 정보를 확인할 것입니다.

[/manage] 관리자가 자신이 관리하는 차량과 예약 정보를 확인하는 페이지입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명//manage.js

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명//manage.hbs

자신이 담당하고 있는 예약과 차량 정보를 확인하기 위해 getadminsale과 getadminvehicle 함수를 이용하여 값을 확인합니다. 이 둘은 where 문에 자신이 로그인 한 관리자 아이디가 들어가는 함수들입니다.

[/select] 고객이 자신의 예약 정보들과 아직 예약되지 않은 차량들을 확인할 수 있는 화면입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명//select.js

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트, 화면, 은색, 스크린샷이(가) 표시된 사진

자동 생성된 설명

//select.hbs

자신이 예약한 정보를 확인하도록  var logid = getlogid();

  const [rows] = await promisePool.query(`select \* from sale where customerid = "${logid}"`);

로그인 아이디를 받아와서 자신의 아이디를 customerid로 입력합니다.

  getpossiblevehicle : async () => {

    const [rows] = await promisePool.query(`select \* from vehicle where Reservation = 'NO'`);

    return rows

}, // 예약 가능한 차량 정보만 read. Reservation 값이 NO 인지로 구분

예약 가능한 차량일 경우 reservation 컬럼 값이 no 이므로 이 값을 갖는 차량만 read 합니다.

[/insert/shopping] 고객 예약 추가

//js

텍스트이(가) 표시된 사진

자동 생성된 설명’//hbs

고객이 예약 정보를 추가하는 부분입니다. 이 때 router.post 내에서 update와 create를 동시에 진행합니다. 이는 내가 예약하고자 하는 차량의 reservation 값을 YES로 update 해주고 나의 예약정보를 insert하는 것입니다.

[insert/adminvhc] 관리자 차량 추가

텍스트이(가) 표시된 사진

자동 생성된 설명 //insert.js

 텍스트이(가) 표시된 사진

자동 생성된 설명//adminvhc.hbs

관리자가 차량을 추가하는 부분입니다. 이 때 차량을 받아와서 차량의 모든 값들을 입력해주면 추가됩니다.

[/update/vhc] 관리자가 차량을 수정

텍스트이(가) 표시된 사진

자동 생성된 설명//js

텍스트이(가) 표시된 사진

자동 생성된 설명//hbs

관리자가 차량을 수정하는 페이지입니다. 차량의 수정하고 싶은 값과 차량의 다른 주요 값들을 함께 입력하면 관리자는 차량을 수정할 수 있습니다.

[/update/sale] 관리자 예약 정보 수정

텍스트이(가) 표시된 사진

자동 생성된 설명//js

텍스트이(가) 표시된 사진

자동 생성된 설명//hbs

관리자가 예약을 수정하는 코드입니다.

[delete/usersale] 고객이 예약을 취소

텍스트이(가) 표시된 사진

자동 생성된 설명//js

텍스트, 모니터, 실내, 화면이(가) 표시된 사진

자동 생성된 설명//hbs

고객이 예약을 취소합니다. 자신의 예약을 취소하는 것이므로 자신의 아이디를 넣는 query문을 동작하도록 작성하였습니다. 예약을 취소하면 해당 차량은 예약이 없는 상태가 되므로 이후에 해당 차량의 reservation 애트리뷰트를 NO로 update 해주도록 하였습니다.

[/delete/complete] 관리자가 판매 완료 설정

텍스트이(가) 표시된 사진

자동 생성된 설명//js

텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명//hbs

판매가 완료될 시 해당 예약과 해당 차량을 삭제시키도록 코드 작성하였습니다.

[/delete/fail] 관리자 판매 실패

텍스트이(가) 표시된 사진

자동 생성된 설명//js

텍스트이(가) 표시된 사진

자동 생성된 설명//hbs

판매가 실패할 경우에는 해당 예약만 삭제하고 차량은 다시 예약이 없는 상태로 돌아가야 하므로 reservation 을 NO 로 수정시킵니다.

텍스트이(가) 표시된 사진

자동 생성된 설명//js

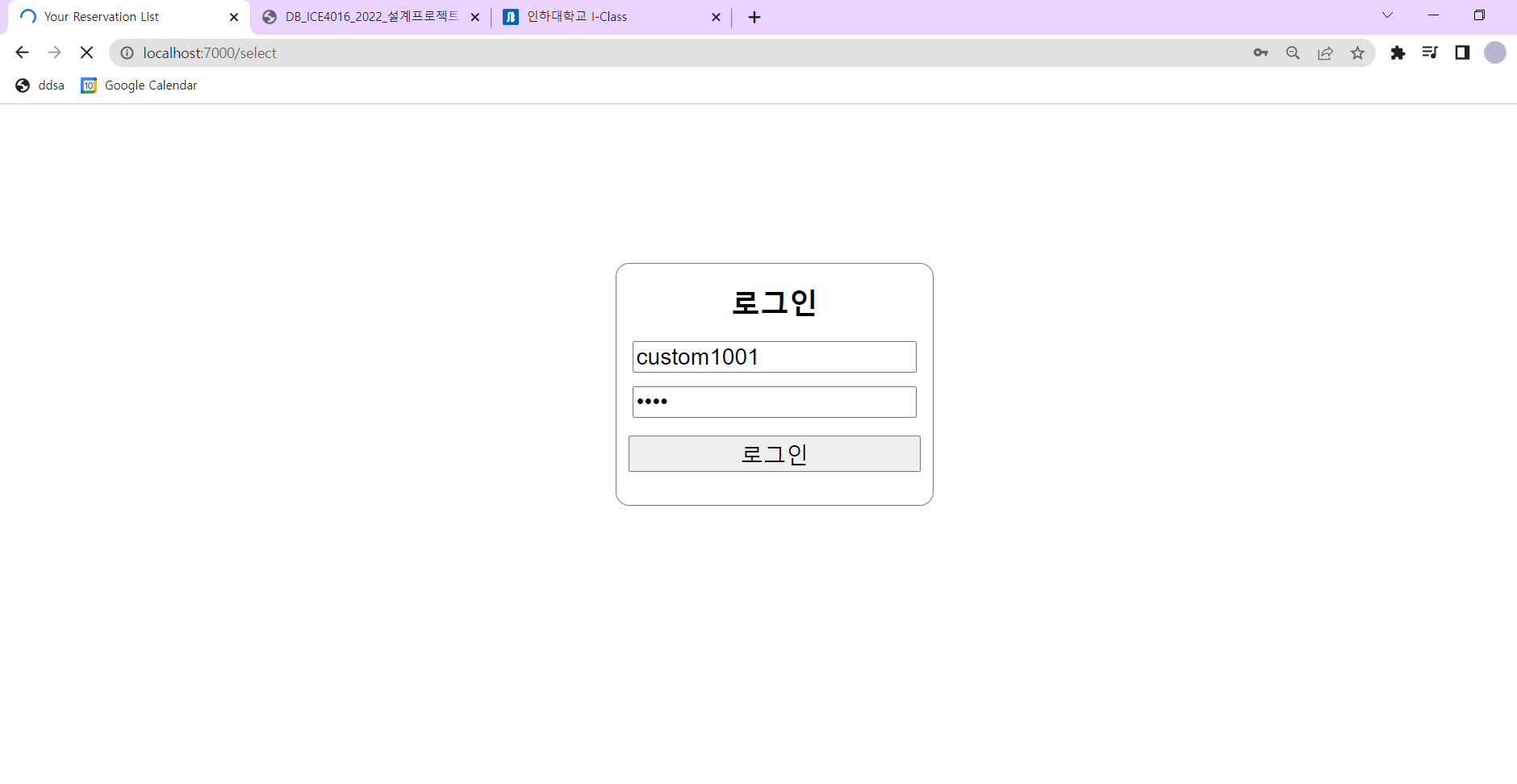
텍스트, 화면, 스크린샷, 은색이(가) 표시된 사진

자동 생성된 설명//hbs

관리자가 차량을 삭제하는 코드입니다. vehicle의 id를 받아와서 삭제를 진행합니다.

1. 실행 화면

[고객 – 로그인]



로그인하여 들어갑니다.

테이블이(가) 표시된 사진

자동 생성된 설명/select

다음과 같이 나의 예약 정보와 가능한 차량 정보들을 확인할 수 있습니다.

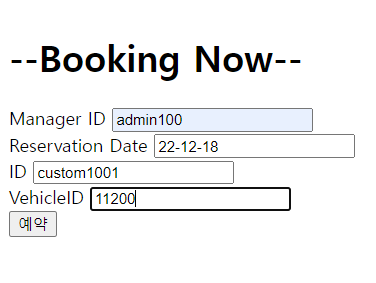
10만개의 데이터라 스크롤이 너무 길어 캡쳐를 생략하였습니다.

[고객 – 예약]

텍스트이(가) 표시된 사진

자동 생성된 설명

Go to Book을 눌러서 예약 페이지로 이동하면 값을 작성하여 예약할 수 있습니다.



다음과 같이 입력하고 예약을 누른 뒤 다시 Go to ~ 링크를 눌러 이동하면

텍스트이(가) 표시된 사진

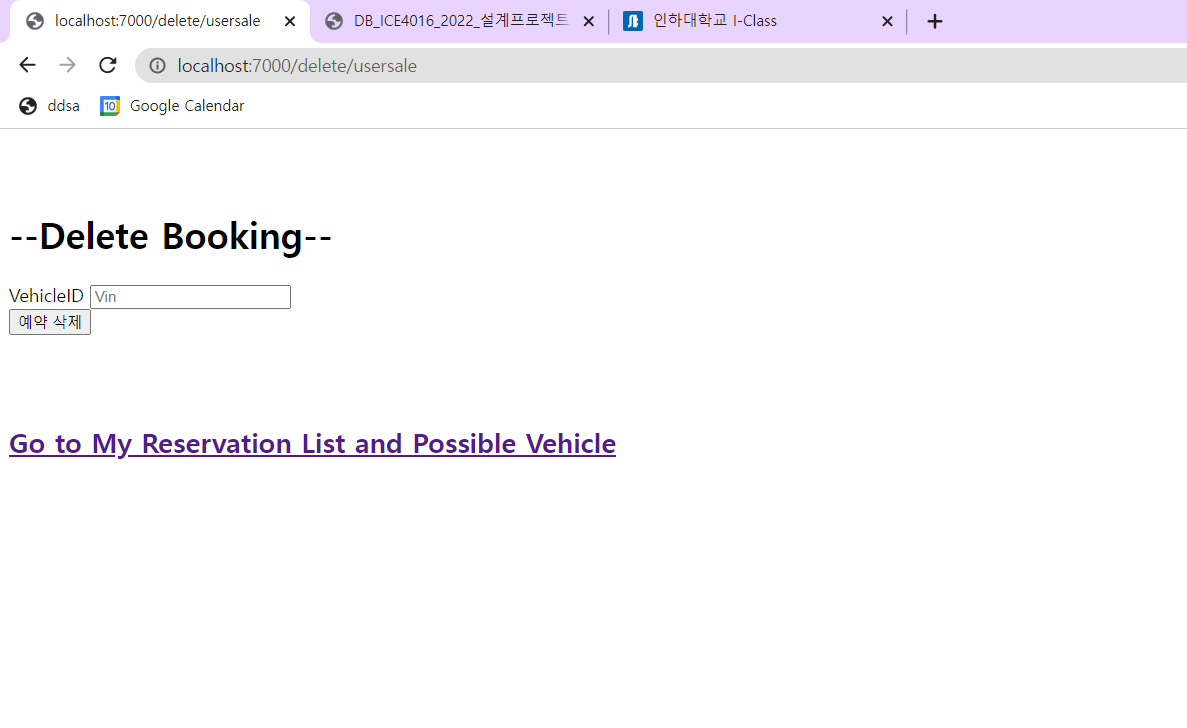
자동 생성된 설명

예약이 추가됩니다.

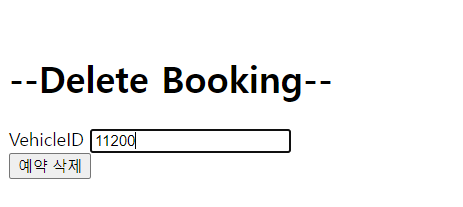
[예약 – 삭제]



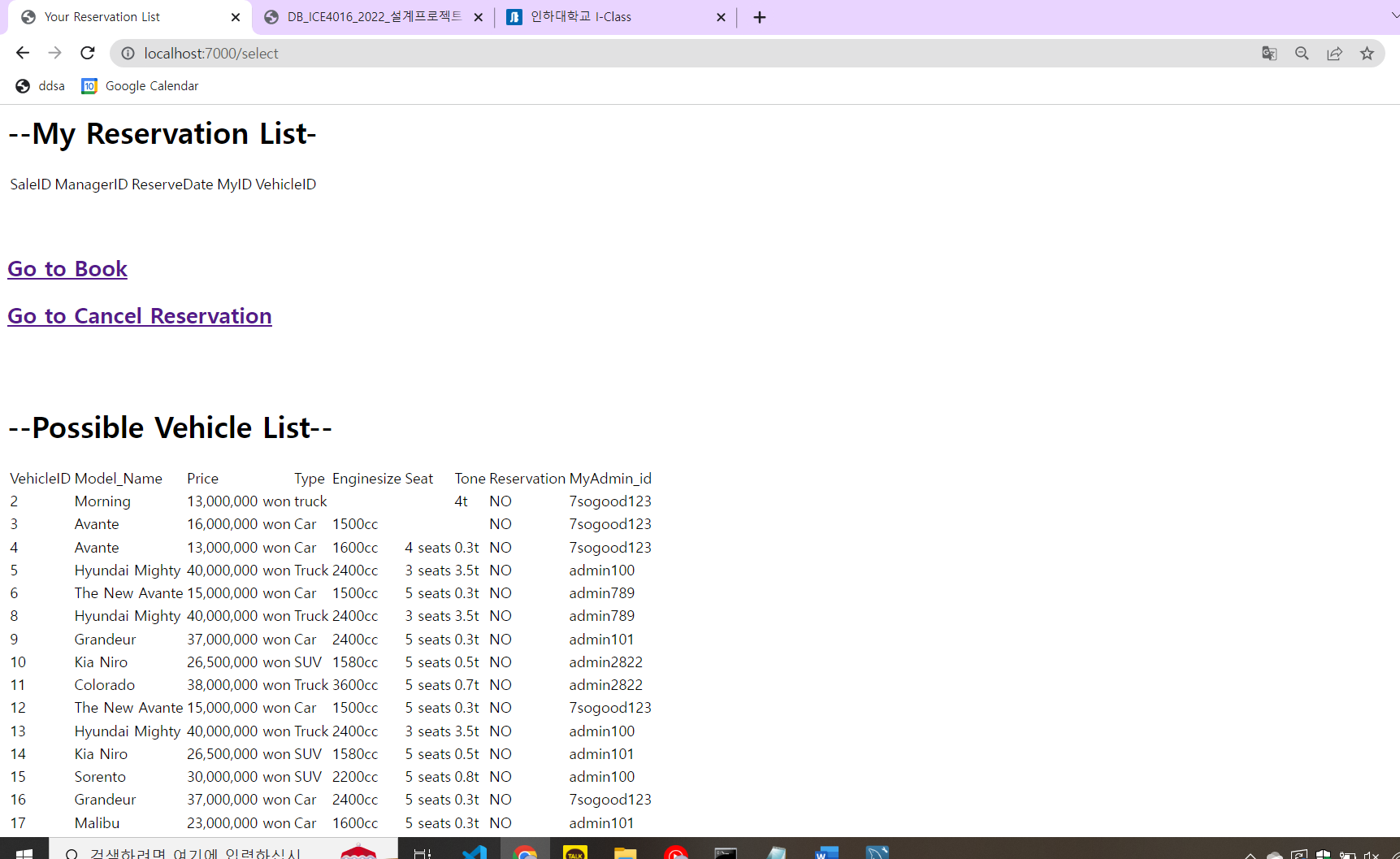
이번에는 select 페이지에서 go to cancel reservation 을 눌러 이동하면



예약을 삭제하는 페이지가 나옵니다.

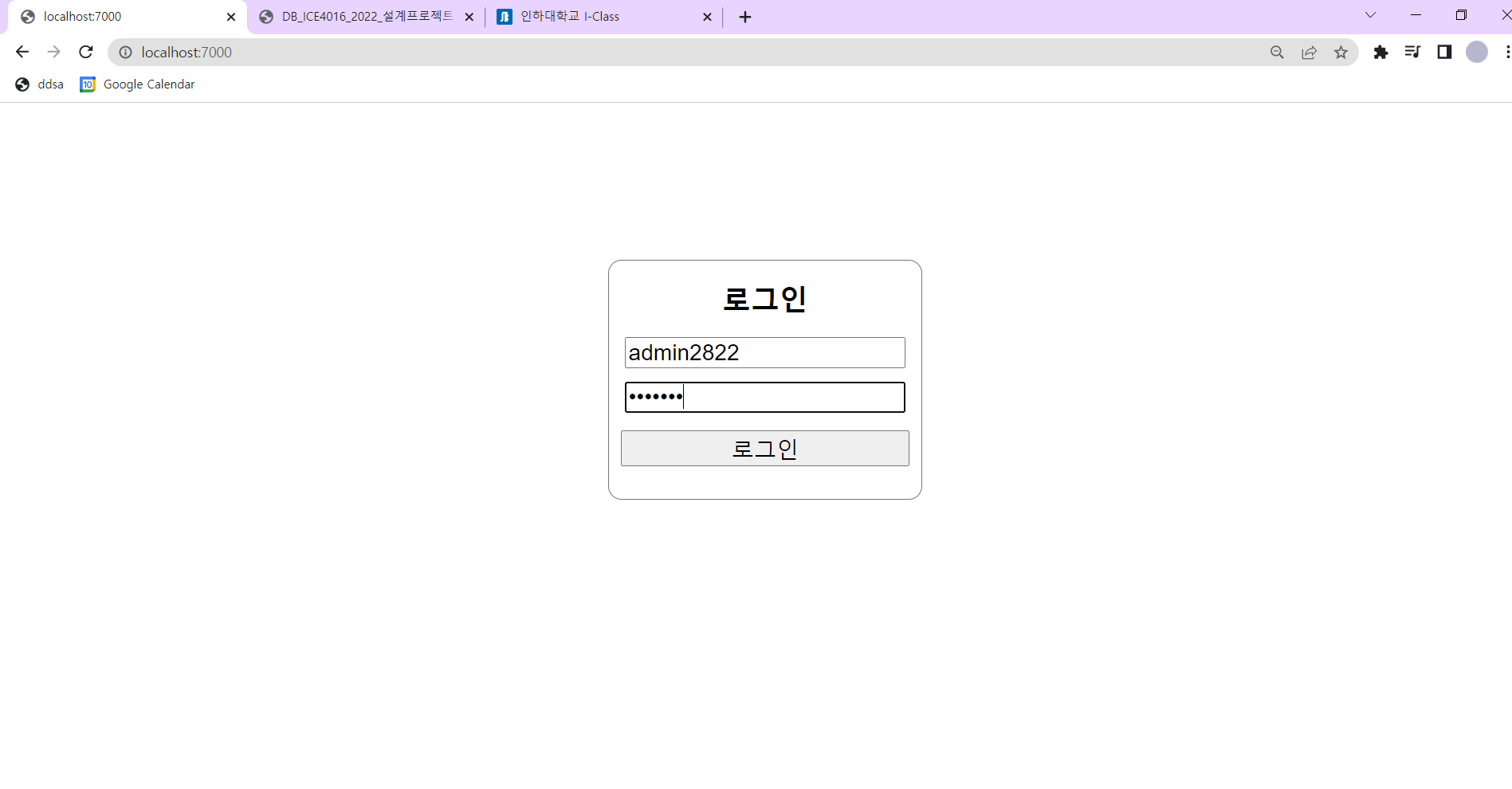


삭제하려는 차량의 아이디를 입력하고 다시 select로 이동하면



다시 없어진 모습을 확인할 수 있습니다.

[관리자 – 로그인]



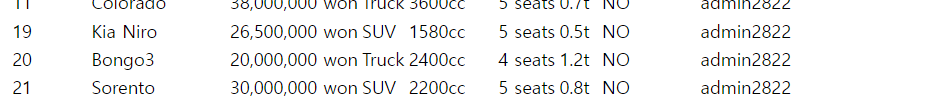
관리자 아이디로 로그인하면

텍스트이(가) 표시된 사진

자동 생성된 설명

내가 관리하고 있는 차량과 예약 정보가 뜹니다.

[차량 정보 수정]



20번 bongo3의 가격을 2100만원으로 수정해보겠습니다.

차량 수정하기로 이동하여

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 수정하고 정보 수정을 누르고 다시 이동하면

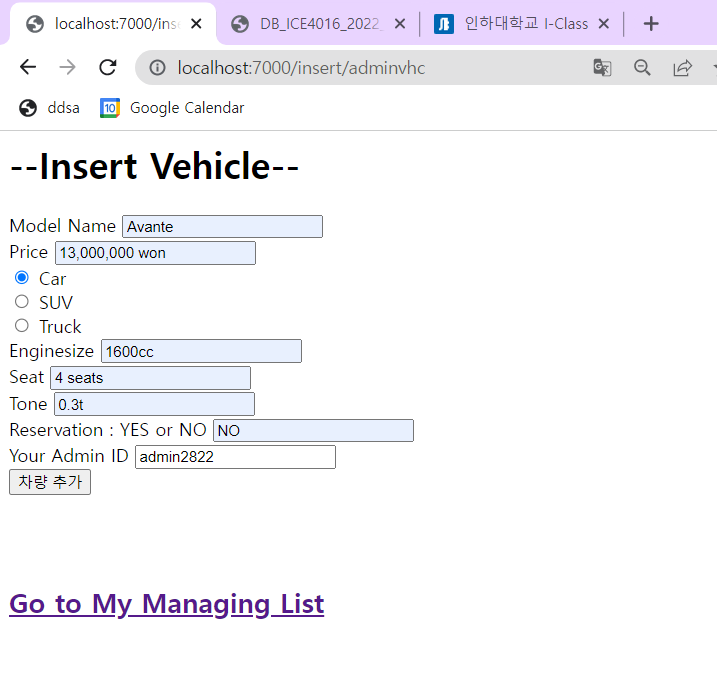
텍스트이(가) 표시된 사진

자동 생성된 설명

수정되었습니다.

[차량 추가]

차량 추가하기로 이동하면



다음과 같이 차량 정보를 입력할 수 있습니다. 이 때 Type을 결정하는 부분에서는 input type을 radio로 하여 세 개 중 선택하도록 하였습니다.

추가버튼을 누르고 이동해보면

테이블이(가) 표시된 사진

자동 생성된 설명

가장 마지막에 내가 관리하는 차량으로 등록된 것을 볼 수 있습니다.

[차량 삭제하기]

텍스트이(가) 표시된 사진

자동 생성된 설명

차량 삭제로 이동하여 방금 추가한 아반떼를 없애보면

텍스트이(가) 표시된 사진

자동 생성된 설명

완료 메시지를 띄우고

텍스트이(가) 표시된 사진

자동 생성된 설명

없어진 것을 알 수 있습니다 (방금 추가한 차만 1300만원이고 나머지는 모두 다른 금액이라 1300만으로 검색했습니다)

[판매완료처리]

텍스트이(가) 표시된 사진

자동 생성된 설명

판매완료하기로 이동하여 예약된 67번차를 판매완료하겠습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

완료된 모습입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

참고로 로그인 실패 시 실패라고 뜨는 것을 볼 수 있습니다.

1. 결론

erd를 그리는 것이 내가 원하는 대로 고쳐야 하는데 처음에 만든 것과 계속 달라져서 힘들었습니다. 각종 제약조건을 설정해주는 부분에서 어려움이 많았고 테이블을 수백번은 고친 것 같습니다. 그만큼 처음에 모든 설정을 신중하게 하고 개념설계를 깊이 고민하며 진행해야한다는 것을 느꼈습니다. 또 트랜잭션이나 이런부분에 대해 많이 고민해보았는데 고객간의 충돌이 있을 때 어려움을 느꼈습니다. 예약 부분에서 YES NO로 예약된 차는 안보이게 한 것과 예약 취소를 하고 판매가 완료 될 때 해당 값을 수정해주는 것으로 그나마의 충돌을 막을 수 있었다고 생각합니다. 또한 자바스크립트와 hbs 를 많이 사용해보지 않았지만 실습 때 꾸준히 했던 내용이라 쉬울 것이라고 생각하였으나, 막상 시도해보니 해보지 않은 방법들에 대해서는 어려움이 많았습니다. 그렇지만 구글링도 열심히 하였고, 또 그냥 따라하기 식의 공부법이 아니라 직접 어떻게 전달이 되고 어떤 식으로 동작하는 지에 대한 로직을 이해하고 풀어보니 해결이 쉬웠던 것 같습니다. 데이터 베이스 설계를 통해 정말 데이터베이스의 기본적인 이론들과 해당 실습 언어들을 익힐 수 있어 저에게 정말 도움이 되었습니다.

데이터베이스 스크립트

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema finalproject

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema finalproject

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `finalproject` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci ;

USE `finalproject` ;

-- -----------------------------------------------------

-- Table `finalproject`.`user`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `finalproject`.`user` (

`uid` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`isadmin` VARCHAR(45) NOT NULL,

PRIMARY KEY (`uid`),

UNIQUE INDEX `uid\_UNIQUE` (`uid` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `finalproject`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `finalproject`.`customer` (

`Ssn` VARCHAR(45) NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`city` VARCHAR(45) NOT NULL,

`gu` VARCHAR(45) NULL DEFAULT NULL,

`dong` VARCHAR(45) NULL DEFAULT NULL,

`user\_uid` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Ssn`),

UNIQUE INDEX `Ssn\_UNIQUE` (`Ssn` ASC) VISIBLE,

INDEX `fk\_customer\_user1\_idx` (`user\_uid` ASC) VISIBLE,

CONSTRAINT `fk\_customer\_user1`

FOREIGN KEY (`user\_uid`)

REFERENCES `finalproject`.`user` (`uid`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `finalproject`.`salesperson`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `finalproject`.`salesperson` (

`Sid` VARCHAR(45) NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`user\_uid` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Sid`),

UNIQUE INDEX `Sid\_UNIQUE` (`Sid` ASC) VISIBLE,

INDEX `fk\_salesperson\_user1\_idx` (`user\_uid` ASC) VISIBLE,

CONSTRAINT `fk\_salesperson\_user1`

FOREIGN KEY (`user\_uid`)

REFERENCES `finalproject`.`user` (`uid`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `finalproject`.`vehicle`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `finalproject`.`vehicle` (

`Vin` INT NOT NULL AUTO\_INCREMENT,

`Model` VARCHAR(45) NOT NULL,

`Price` VARCHAR(45) NOT NULL,

`Type` VARCHAR(45) NOT NULL,

`Enginesize` VARCHAR(45) NULL DEFAULT NULL,

`Seat` VARCHAR(45) NULL DEFAULT NULL,

`Tone` VARCHAR(45) NULL DEFAULT NULL,

`Reservation` VARCHAR(10) NOT NULL,

`salesperson\_id` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Vin`),

UNIQUE INDEX `Vin\_UNIQUE` (`Vin` ASC) VISIBLE,

INDEX `fk\_vehicle\_salesperson1\_idx` (`salesperson\_id` ASC) VISIBLE,

CONSTRAINT `fk\_vehicle\_salesperson1`

FOREIGN KEY (`salesperson\_id`)

REFERENCES `finalproject`.`salesperson` (`Sid`))

ENGINE = InnoDB

AUTO\_INCREMENT = 100005

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `finalproject`.`sale`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `finalproject`.`sale` (

`said` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`manageid` VARCHAR(45) NOT NULL,

`reservedate` DATE NOT NULL,

`customerid` VARCHAR(45) NOT NULL,

`vid` INT NOT NULL,

PRIMARY KEY (`said`),

UNIQUE INDEX `said\_UNIQUE` (`said` ASC) VISIBLE,

INDEX `fk\_SALE\_SALESPERSON1\_idx` (`manageid` ASC) VISIBLE,

INDEX `fk\_SALE\_CUSTOMER1\_idx` (`customerid` ASC) VISIBLE,

INDEX `fk\_SALE\_VEHICLE1\_idx` (`vid` ASC) VISIBLE,

CONSTRAINT `fk\_SALE\_CUSTOMER1`

FOREIGN KEY (`customerid`)

REFERENCES `finalproject`.`customer` (`Ssn`),

CONSTRAINT `fk\_SALE\_SALESPERSON1`

FOREIGN KEY (`manageid`)

REFERENCES `finalproject`.`salesperson` (`Sid`),

CONSTRAINT `fk\_SALE\_VEHICLE1`

FOREIGN KEY (`vid`)

REFERENCES `finalproject`.`vehicle` (`Vin`))

ENGINE = InnoDB

AUTO\_INCREMENT = 15

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;