

주차장 관리 시스템 요구사항

- 1) 고객 목록에는 차량번호, 카드정보, 이름, 전화번호를 기록한다(개체).
- 2) 주차장 목록에는 주차 칸, 주차상태, 차량번호를 기록한다(개체).
- 3) 고객은 한 칸의 주차장에 주차를 할 수 있고, 이미 주차된 칸에는 주차할 수 없다(고객과 주차장의 관계는 1:1).
- 4) 주차장 관리 시스템은 고객이 주차를 시작한 시각과 출차한 시각, 주차 요금과 무료주차 시간, 결제 여부를 따로 저장한다(고객과 주차장의 관계에 속성이 존재함)
- 5) 쇼핑몰 관리 시스템은 고객이 쇼핑몰에서 결제한 시각과 금액, 고객의 전화번호를 저장한다.

트랜잭션

- 1) 차량 번호로 차량 위치 정보 검색이 가능해야 한다.
- 2) 차량 번호로 회원 여부, 회원의 카드 정보 검색이 가능해야 한다.
- 3) 특정 기간 안에서 월별, 요일별, 시간대별로 주차장 이용량(차량 수) 통계를 낼 수 있어야 한다.
- 4) 주차요금은 총 주차 시간에서 쇼핑몰 결제 금액에 따른 무료주차시간을 뺀 시간을 (단위 시간*단위요금)에 따라 계산한다. 단, 총 주차시간에서 무료주차시간을 뺀 시간은 0보다 크거나 같다.
- 5) 고객의 주차요금 결제가 완료되기 위하여, 해당 고객의 결제 여부가 0(false)에서 1(true)로 바뀐다.
- 6) 고객의 주차 또는 차량 빼기 작업에 따라 주차장 칸의 상태가 바뀐다.

Database Schema

PMS = { PARKINGLOT_LIST, CUSTOMER_LIST, PARK_PAY, SHOPPING_PAY, PRICE, DISCOUNT }

Table Schema

PARKINGLOT_LIST = {Parking_spot, Parking_status, Customer_car_num}

CUSTOMER_LIST = { Customer_car_num, Customer_card_info, Customer_name, Customer_phone}

PARK_PAY = {Customer_car_num, Parking_spot, Park_in, Park_out, Park_free_hour, Park_pay_amount, Park_is_paid}

SHOPPING_PAY = {Customer_phone, Shopping_pay_amount, Shopping_pay_time}

PRICE = {Unit_minute, Unit_price}

DISCOUNT = {Shopping_pay_minimum, Free_hour}

널 제약 조건

PARKINGLOT_LIST의 Parking_status는 NULL이 될 수 없다. Default 값은 0 (vacant)이다.

PARK_PAY의 Park_free_hour, Park_pay_amount는 NULL이 될 수 없다. Default 값은 0이다.

SHOPPING_PAY의 Shopping_pay_amount는 NULL이 될 수 없다. Default 값은 0이다.

외래키 제약 조건

TABLE PARKINGLOT_LIST

CONSTRAINT R_XX FOREIGN KEY (Customer_car_num) REFERENCES CUSTOMER_LIST
(Customer_car_num) ON DELETE CASCADE ON UPDATE CASCADE);

ALTER TABLE PARK_PAY

ADD (CONSTRAINT R_XX FOREIGN KEY (Customer_car_num) REFERENCES CUSTOMER_LIST
(Customer_car_num) ON DELETE CASCADE ON UPDATE CASCADE);

ALTER TABLE PARK_PAY

ADD (CONSTRAINT R_XX FOREIGN KEY (Parking_spot) REFERENCES PARKINGLOT_LIST
(Parking_spot) ON DELETE CASCADE ON UPDATE CASCADE);

ALTER TABLE SHOPPING_PAY

ADD (CONSTRAINT R_XX FOREIGN KEY (Customer_phone) REFERENCES CUSTOMER_LIST
(Customer_phone) ON DELETE CASCADE ON UPDATE CASCADE);