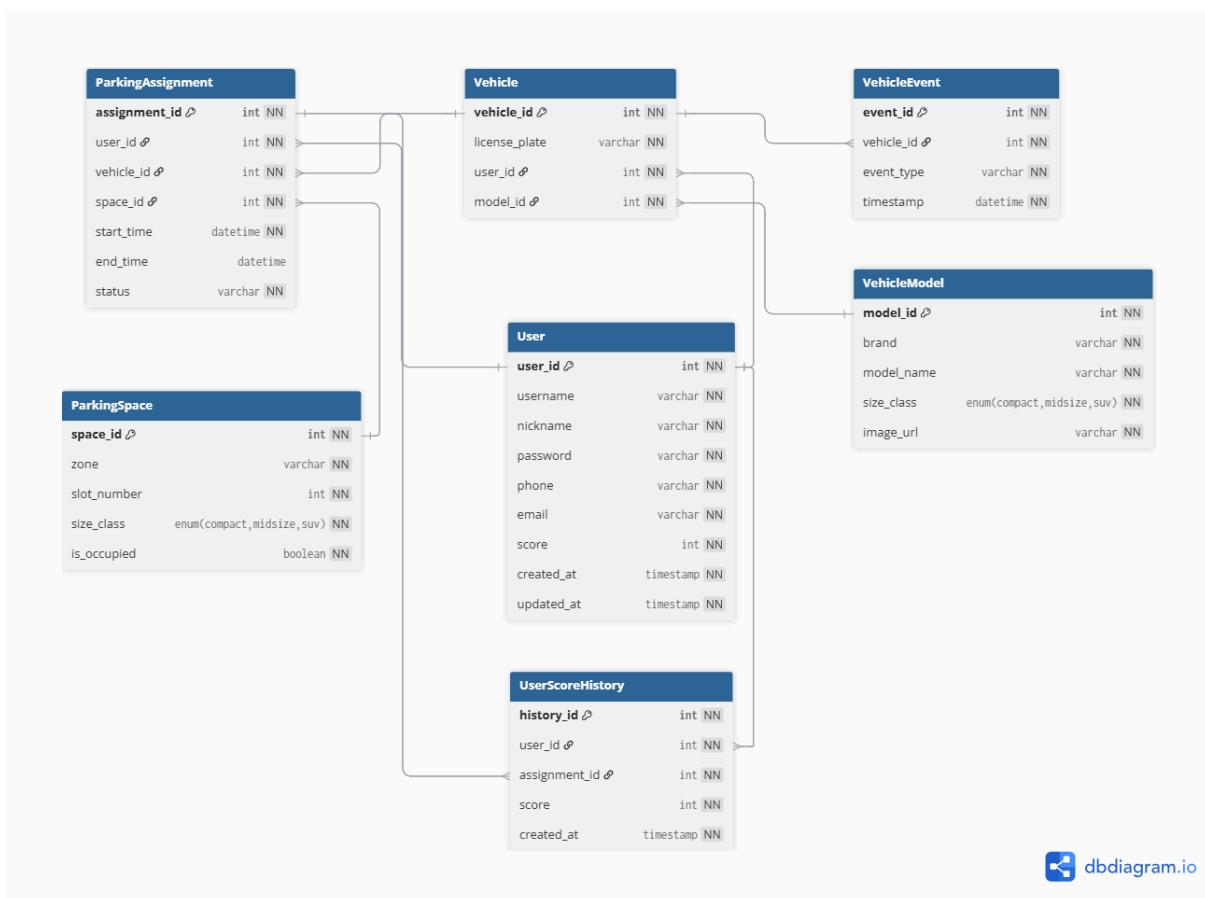




DB 설계 (ERD)

■ 상태 ■ 개발

<https://dbdiagram.io/d/687f4a8dcca18e685c17ed1d>



dbdiagram.io

[ERD 최종1.pdf](#)

// 사용자 정보를 저장하는 테이블

Table User {

```
user_id      int      [pk, increment, not null] // 사용자 고유 ID
username     varchar   [not null]                // 사용자 실명
nickname     varchar   [unique, not null]         // 로그인용 사용자명
password     varchar   [not null]                // 비밀번호 해시
phone        varchar   [not null]                // 연락처 전화번호
```

```

email      varchar [not null]          // 이메일 주소
score      int    [not null, default: 0]  // 주차 실력 점수
created_at timestamp [not null, default: `now()`]// 레코드 생성 시각
updated_at timestamp [not null, default: `now()`]// 레코드 최종 수정 시각
}

// 사용자 보유 차량을 저장하는 테이블
Table Vehicle {
    vehicle_id  int   [pk, increment, not null] // 차량 고유 ID
    license_plate varchar [unique, not null]      // 차량 번호판
    user_id     int   [unique, ref: > User.user_id, not null] // 소유자 User 테이
    블 참조
    model_id    int   [ref: > VehicleModel.model_id, not null] // 차량 모델 Ve
    hicleModel 참조
}

// 차량 모델 정보를 저장하는 테이블
Table VehicleModel {
    model_id  int   [pk, increment, not null]          // 모델 고유 ID
    brand      varchar [not null]                      // 제조사 (ex. Hyundai)
    model_name varchar [not null]                      // 모델명 (ex. Sonata)
    size_class enum('compact','midsize','suv') [not null] // 크기 분류
    image_url  varchar [not null]                      // 대표 이미지 URL
}

// 주차 공간 정보를 저장하는 테이블
Table ParkingSpace {
    space_id  int   [pk, increment, not null] // 공간 고유 ID
    zone      varchar [not null]           // 구역 구분 (예: A, B)
    slot_number int   [not null]           // 슬롯 번호 (예: 1~10)
    size_class enum('compact','midsize','suv') [not null] // 허용 차량 크기
    is_occupied boolean [not null, default: false] // 점유 여부
}

// 주차 배정 이력을 저장하는 테이블
Table ParkingAssignment {
    assignment_id int   [pk, increment, not null] // 배정 기록 ID
    user_id      int   [ref: > User.user_id, not null] // 배정된 사용자
}

```

```

vehicle_id int [ref: > Vehicle.vehicle_id, not null] // 배정된 차량
space_id int [ref: > ParkingSpace.space_id, not null] // 배정된 공간
start_time datetime [not null] // 입차 시각
end_time datetime // 출차 시각 (미완료 시 NULL)
status varchar [not null] // 상태 (ex. 'ASSIGNED','COMPLETED')
}
}

// 차량 이벤트(입차, 출차, 예외) 로그를 저장하는 테이블
Table VehicleEvent {
    event_id int [pk, increment, not null] // 이벤트 고유 ID
    vehicle_id int [ref: > Vehicle.vehicle_id, not null] // 관련 차량
    event_type varchar [not null] // 이벤트 종류 (ex. 'Entrance','Exit')
    timestamp datetime [not null] // 이벤트 발생 시각
}

// 사용자 점수 변경 이력을 저장하는 테이블
Table UserScoreHistory {
    history_id int [pk, increment, not null] // 히스토리 고유 ID
    user_id int [ref: > User.user_id, not null] // 해당 사용자
    assignment_id int [ref: > ParkingAssignment.assignment_id, not null]
    // (선택) 점수 변화 연관 주차 배정
    score int [not null] // 주차 점수
    created_at timestamp [not null, default: `now()`] // 점수 변경 시각
}

```