



```
-- 공연 테이블: 공연 정보 저장
CREATE TABLE concerts (
    id SERIAL PRIMARY KEY, -- 고유 공연 식별자
    title VARCHAR(255) NOT NULL, -- 공연 제목
    date_and_time TIMESTAMP NOT NULL UNIQUE, -- 공연 일정
    concerts_date DATE NOT NULL, -- 공연 날짜
    concerts_time TIME NOT NULL, -- 공연 시간
    poster_url VARCHAR(512), -- 포스터 이미지 URL
    location VARCHAR(255) NOT NULL, -- 공연 장소
    duration INT, -- 공연 관람시간 (분)
    seat_type VARCHAR(50), -- 좌석 유형
    rating VARCHAR(50), -- 공연 등급
    standing_price INT, -- 스탠딩석 가격
    seated_price INT, -- 지정석 가격
    ticket_link VARCHAR(512), -- 티켓 예매 링크

    view_count INT DEFAULT 0, -- 조회수
    like_count INT DEFAULT 0, -- 좋아요 수

    concert_status concert_status ENUM('Upcoming', 'Ongoing', 'Ended', 'Cancelled') NOT NULL -- 공연 상태 (공연 예정, 공연 진행중, 공연 종료, 공연 취소)

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- 생성 일시
    created_by INT REFERENCES users(id), -- 생성한 사용자 (시스템/크롤러 ID)
    modified_at TIMESTAMP, -- 수정 일시
    modified_by INT REFERENCES users(id), -- 수정한 사용자 (시스템/크롤러 ID)
    deleted_at TIMESTAMP, -- 삭제 시각 (Soft Delete 구현)
    deleted_by INT REFERENCES users(id), -- 삭제한 사용자 (시스템/크롤러 ID)
    is_deleted BOOLEAN DEFAULT FALSE, -- 삭제 여부 (Soft Delete 구현)

    INDEX idx_title (title), -- 공연 제목에 대한 인덱스 (조회 용도)
    INDEX idx_date (concerts_date), -- 공연 날짜에 대한 인덱스 (조회 용도)
    INDEX idx_like_count (like_count), -- 좋아요 수에 대한 인덱스 (정렬 용도)
    INDEX idx_view_count (view_count) -- 조회수에 대한 인덱스 (정렬 용도)
);
```

```
-- 공연 좋아요 테이블: 사용자의 관심 공연을 저장하는 테이블
CREATE TABLE concert_likes (
    id SERIAL PRIMARY KEY,
    concert_id INT REFERENCES concerts(id) ON DELETE CASCADE,
    user_id INT REFERENCES users(id) ON DELETE CASCADE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    UNIQUE (concert_id, user_id),
    INDEX idx_concert_id (concert_id),
    INDEX idx_user_id (user_id)
);

-- 고유 좋아요 식별자
-- 공연을 참조하는 외래키: 공연이 삭제되면 좋아요도 삭제
-- 사용자를 참조하는 외래키: 사용자가 삭제되면 좋아요도 삭제
-- 생성 일시

-- 한 사용자가 동일한 공연에 중복으로 좋아요를 누르지 못하도록 UNIQUE 제약 추가
-- 공연 ID에 대한 인덱스: 공연별로 좋아요를 빠르게 카운트 하기 위함
-- 사용자 ID에 대한 인덱스: 사용자별로 좋아요를 누른 공연을 빠르게 조회하기 위함
```

```
-- 티켓 테이블: 티켓 정보 저장
CREATE TABLE tickets (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(id),
  concert_id INT REFERENCES concerts(id),
  payment_id INT REFERENCES payment_history (id),

  seat_type VARCHAR(50),
  seat_number VARCHAR(50),
  purchase_price INT,
  purchase_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  ticket_status ENUM('Active', 'Cancelled', 'Used'),

  qr_code_key VARCHAR(255),
  qr_code_created_at TIMESTAMP,

  INDEX idx_concert_id (concert_id),
  INDEX idx_user_id (user_id)
  INDEX idx_payment_id (payment_id),
  INDEX idx_qr_code_key (qr_code_key)
);

-- 고유 티켓 식별자
-- 사용자를 참조하는 외래키
-- 공연을 참조하는 외래키
-- 결제 기록을 참조하는 외래키

-- 좌석 타입
-- 좌석 번호
-- 구매 가격
-- 구매 일시
-- 티켓 상태

-- QR 코드 값
-- QR 코드 생성 일시

-- 공연 ID에 대한 인덱스
-- 사용자 ID에 대한 인덱스
-- 공연 ID에 대한 인덱스
-- QR 코드 키에 대한 인덱스
```

```
-- 결제 기록 테이블: 결제 정보 저장
CREATE TABLE payment_history (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(id) ON DELETE CASCADE,
  concert_id INT REFERENCES concerts(id) ON DELETE CASCADE,

  amount INT NOT NULL,
  payment_method ENUM('CreditCard', 'Paypal', 'BankTransfer'),
  payment_status ENUM('Success', 'Failed', 'Refunded') NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  modified_at TIMESTAMP,

  INDEX idx_user_id (user_id),
  INDEX idx_concert_id (concert_id),
  INDEX idx_payment_status (payment_status)
);

-- 고유 결제 식별자
-- 사용자를 참조하는 외래키
-- 공연을 참조하는 외래키

-- 결제 금액
-- 결제 방법
-- 결제 상태
-- 결제 시각
-- 수정 시각

-- 사용자 ID에 대한 인덱스
-- 공연 ID에 대한 인덱스
-- 결제 상태에 대한 인덱스
```

```
-- 알림 기록 테이블 : 알림 정보 저장
CREATE TABLE notifications (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(id) ON DELETE CASCADE,
  message TEXT NOT NULL,
  is_read BOOLEAN DEFAULT FALSE,
  read_at TIMESTAMP,

  related_concert_id INT REFERENCES concerts(id),
  related_ticket_id INT REFERENCES tickets(id),

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  created_by INT REFERENCES users(id),
  modified_at TIMESTAMP,
  modified_by INT REFERENCES users(id),
  delete_at TIMESTAMP,
  delete_by INT REFERENCES users(id),

  INDEX idx_user_id (user_id),
  INDEX idx_type (type),
  INDEX idx_status (status),
  INDEX idx_related_concert_id (related_concert_id),
  INDEX idx_related_ticket_id (related_ticket_id)
);

-- 고유 알림 식별자
-- 알림 수신자를 참조하는 외래키
-- 알림 내용
-- 알림 읽음 상태: 읽었으면 TRUE, 아니면 FALSE
-- 알림을 읽은 시간

-- 관련된 공연을 참조하는 외래키 (optional)
-- 관련된 티켓을 참조하는 외래키 (optional)

-- 알림 생성 시간
-- 생성한 사용자: 레코드를 생성한 사용자의 식별자
-- 수정 일시: 알림이 수정된 시간
-- 수정한 사용자: 레코드를 수정한 사용자의 식별자
-- 삭제 일시
-- 삭제한 사용자: 레코드를 생성한 사용자의 식별자

-- 사용자 ID에 대한 인덱스: 사용자별 알림을 빠르게 조회하기 위함
-- 알림 유형에 대한 인덱스: 알림 유형별로 빠르게 데이터를 가져오기 위함
-- 알림 상태에 대한 인덱스: 읽지 않은 알림을 빠르게 찾기 위함
-- 관련된 공연 ID에 대한 인덱스 (optional)
-- 관련된 티켓 ID에 대한 인덱스 (optional)
```

```
-- UserReviews 테이블: 사용자의 공연 후기를 저장하는 테이블 (추후 기능 제작여부 결정!)
CREATE TABLE concert_reviews (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(id) ON DELETE CASCADE,
  concert_id INT REFERENCES concerts(id) ON DELETE CASCADE,
  content TEXT NOT NULL,
  rating INT CHECK (rating >= 1 AND rating <= 5),

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  created_by INT REFERENCES users(id),
  modified_at TIMESTAMP,
  modified_by INT REFERENCES users(id),
  deleted_at TIMESTAMP,
  deleted_by INT REFERENCES users(id),
  is_deleted BOOLEAN DEFAULT FALSE,

  INDEX idx_user_id (user_id),
  INDEX idx_concert_id (concert_id),
  INDEX idx_rating (rating)
);

-- 고유 리뷰 식별자
-- 리뷰 작성자를 참조하는 외래키
-- 관련된 공연을 참조하는 외래키
-- 리뷰 내용
-- 공연에 대한 평점 (1~5)

-- 생성 일시
-- 생성한 사용자: 레코드를 생성한 사용자의 식별자
-- 수정 일시
-- 수정한 사용자: 레코드를 수정한 사용자의 식별자
-- 삭제 시각 (Soft Delete 구현)
-- 삭제한 사용자: 레코드를 삭제한 사용자의 식별자
-- 삭제 여부 (Soft Delete 구현)

-- 사용자 ID에 대한 인덱스: 사용자별 리뷰를 빠르게 조회하기 위함
-- 공연 ID에 대한 인덱스: 공연별 리뷰를 빠르게 조회하기 위함
-- 평점에 대한 인덱스: 평점별로 빠르게 데이터를 가져오기 위함
```