

이용권 관련 서비스 구현

# 이용권 조회 기능 구현

1. 정기권, 일일권 입력하면 목록 뽑아오는 코드

```
CREATE OR REPLACE PROCEDURE check_day_tickets  
  
(  
    p_type IN ticket.ticket_type%TYPE  
)  
IS  
    CURSOR c_ticket IS  
        SELECT ticket_type, ticket_period, ticket_timepass, ticket_price  
        FROM ticket  
        WHERE ticket_type = p_type;  
    v_type ticket.ticket_type%TYPE;  
    v_period ticket.ticket_period%TYPE;  
    v_timepass ticket.ticket_timepass%TYPE;  
    v_price ticket.ticket_price%TYPE;
```

다음 >>

# 이용권 조회 기능 구현

## 1. 정기권, 일일권 입력하면 목록 뽑아오는 코드

```
BEGIN
    IF p_type = '일일권' OR p_type = '정기권' THEN
        DBMS_OUTPUT.PUT_LINE( CHR(10) || '   ' || ' 종류   기간   시간   금액   ');
        OPEN c_ticket;
        LOOP
            FETCH c_ticket INTO v_type, v_period, v_timepass, v_price;
            EXIT WHEN c_ticket%NOTFOUND;
            IF v_type = '일일권' THEN
                DBMS_OUTPUT.PUT_LINE( CHR(9) || v_type || ' ' || v_period || ' ' || v_timepass || ' ' || v_price);
            ELSIF v_type = '정기권' THEN
                DBMS_OUTPUT.PUT_LINE( CHR(9) || v_type || ' ' || v_period || ' ' || v_timepass || ' ' || v_price);
            END IF;
        END LOOP;
        CLOSE c_ticket;
    ELSE
        RAISE_APPLICATION_ERROR(-20000, '잘못된 정보입니다. 존재하지 않는 이용권입니다. ');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error Code: ' || SQLCODE || ' ' || SQLERRM);
END;

EXECUTE check_day_tickets('일일권');
EXECUTE check_day_tickets('정기권');
EXECUTE check_day_tickets('정');
```

다음 >> 실행으로

# 이용권 정보 조회 실행

1. 정기권, 일일권 입력하면 목록 뽑아오는 코드

종류	기간	시간	금액
일일권	1일	1시간	1000
일일권	1일	2시간	2000

종류	기간	시간	금액
정기권	7일	1시간	3000
정기권	7일	2시간	4000
정기권	30일	1시간	5000
정기권	30일	2시간	7000
정기권	180일	1시간	15000
정기권	180일	2시간	20000
정기권	365일	1시간	30000
정기권	365일	2시간	40000

Error Code: -20000 ORA-20000: 잘못된 정보입니다. 존재하지 않는 이용권입니다.

<<실행>>

# 이용권 결제 기능 구현

## 2. 이용권 구매 시 회원정보 정보에 해당 티켓코드 들어오게 하기

```
CREATE OR REPLACE PROCEDURE ins_ticket_order
(
  pvoucher_id t_order.voucher_id%type,
  puser_id t_order.user_id%type,
  pticket_code t_order.ticket_code%type,
  ppayment_service_code t_order.payment_service_code%type,
  porder_duration_start t_order.order_duration_start%type,
  porder_duration_end t_order.order_duration_end%type,
  porder_fee t_order.order_fee%type,
  porder_paidtime t_order.order_paidtime%type
)
IS
  vuser_id t_order.user_id%type := puser_id;
BEGIN
  INSERT INTO t_order
  VALUES (
    pvoucher_id, puser_id, pticket_code, ppayment_service_code
    ,porder_duration_start, porder_duration_end, porder_fee, porder_paidtime
  );
  UPDATE u_info
  SET ticket_code = pticket_code
  WHERE user_id = puser_id;
END;
```

다음 >> 실행으로

# 이용권 구매 실행

## 2. 이용권 구매 시 회원정보 정보에 해당 티켓코드 들어오게 하기

쿼리 실행 시 이용권 결제내역 테이블에 INSERT됨

```
EXECUTE ins_ticket_order('P-15','ABC123',11,'휴대폰결제','2023-4-4','2023-4-6',3000,'2023-4-3');
```

	VOUCH...	USER_ID	TICKET_CODE	PAYMENT_SERVICE_CODE	ORDER_DURATION_START	ORDER_DURATION_END	ORDER_FEE	ORDER...
1	P-1	ABC123	211	휴대폰결제	23/04/04	23/04/11	3000	23/04/03
2	P-2	ABC456	221	신용/체크카드	23/04/02	23/04/09	4000	23/03/29
3	P-3	ABC789	221	제로페이	23/04/01	23/04/08	4000	23/01/03
4	P-4	비회원1	11	휴대폰결제	23/04/05	23/04/05	1000	23/04/05
5	P-5	비회원2	12	신용/체크카드	23/04/05	23/04/05	2000	23/04/05
6	P-6	비회원3	11	신용/체크카드	23/05/05	23/05/05	1000	23/05/05
7	P-7	비회원4	12	카카오페이	23/06/08	23/06/08	2000	23/06/08
8	P-8	비회원5	12	제로페이	23/05/27	23/05/27	2000	23/05/27
9	P-9	ABC911	224	PAYCO	23/04/05	24/04/05	40000	23/04/05
10	P-10	ABC913	213	카카오페이	23/04/06	23/10/03	15000	23/04/06
11	P-11	ABC914	211	신용/체크카드	22/09/18	22/09/25	3000	22/04/08
12	P-12	ABC915	221	PAYCO	23/07/05	23/07/12	5000	23/04/07
13	P-13	ABC917	214	PAYCO	22/12/10	23/12/10	30000	22/12/09
14	P-14	ABC918	214	카카오페이	23/07/10	24/07/10	30000	23/07/08
15	P-15	ABC123	11	휴대폰결제	23/04/04	23/04/06	3000	23/04/03

<<실행>>

# 대여 서비스 구현

# 대여가능 자전거(일반 / 새싹) 기능 구현

```
CREATE OR REPLACE PROCEDURE sel_bike_quantity
(
  ppaking_address bic_info.bicycle_stand$type
)
IS
  vparking_address bic_info.bicycle_stand$type := ppaking_address;
  vstand_id bic_std.stand_id$type;
  vcommon_bike number;
  vsprout_bike number ;
BEGIN
  SELECT COUNT(*)
  INTO vcommon_bike
  FROM bic_info
  WHERE bicycle_stand = ppaking_address AND bicycle_model= '일반' AND bicycle_status = '비치중';

  SELECT COUNT(*)
  INTO vsprout_bike
  FROM bic_info
  WHERE bicycle_stand = ppaking_address AND bicycle_model= '새싹' AND bicycle_status = '비치중';
```

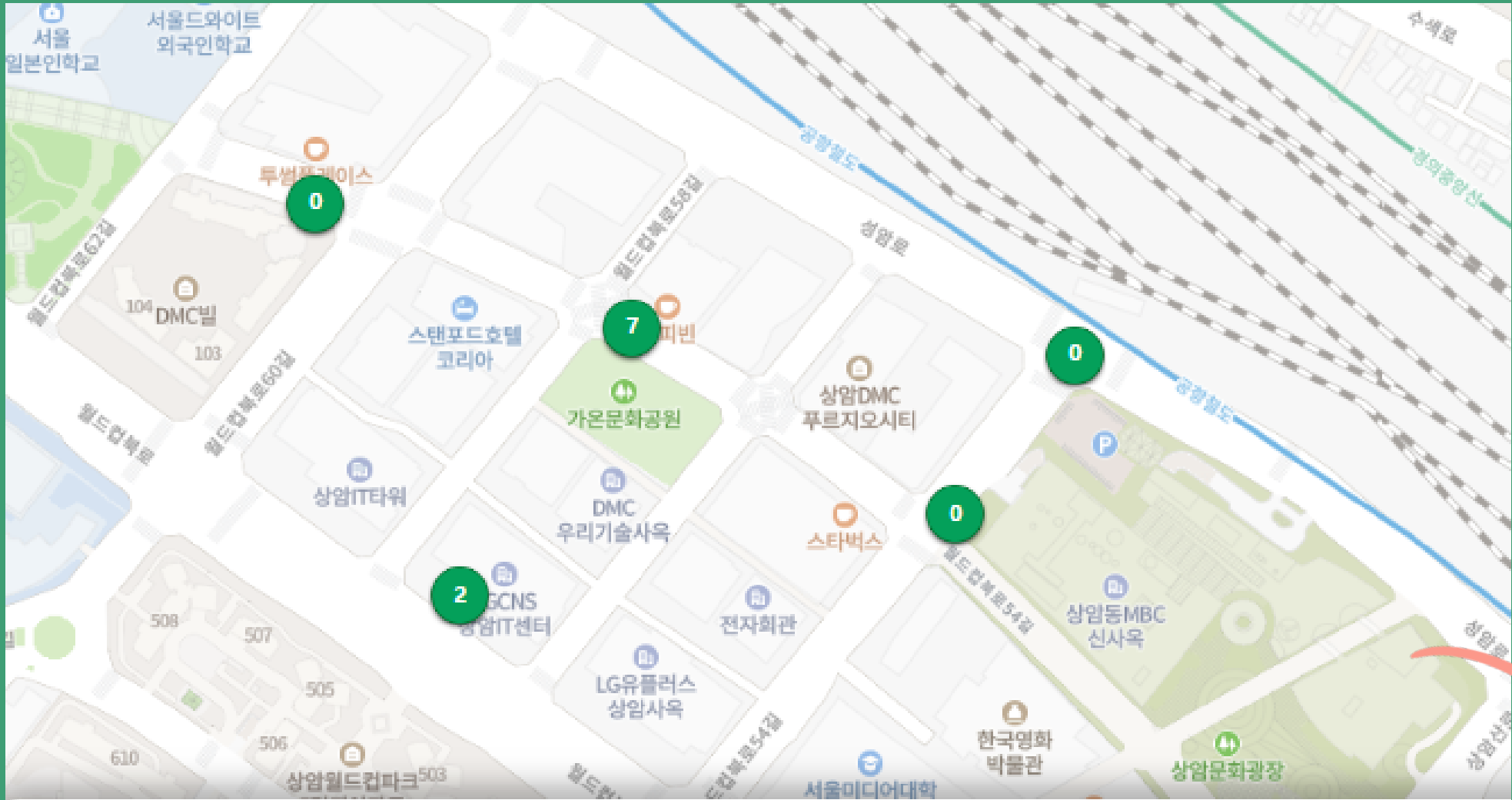
일반 따릉이 개수 조회

```
SELECT stand_id
INTO vstand_id
FROM bic_std
WHERE stand_name = ppaking_address;
DBMS_OUTPUT.PUT_LINE(vstand_id||vparking_address);
DBMS_OUTPUT.PUT_LINE('일반 따릉이 : '|| ' ' || vcommon_bike);
DBMS_OUTPUT.PUT_LINE('새싹 따릉이 : '|| ' ' || vsprout_bike);
END;
EXEC sel_bike_quantity('MBC 앞');
```

새싹 따릉이 개수 조회



# 대여가능 자전거(일반 / 새싹) 실행



3007.MBC 앞 (신) 자전거 추가배치 요청

일반 따릉이 0

새싹 따릉이 0

ST-002MBC 앞

일반 따릉이 : 1

새싹 따릉이 : 3

쿼리 출력 값

실제 홈페이지 내용

# 회원별 이용내역 업데이트 기능 구현

```
CREATE OR REPLACE PROCEDURE ins_update
(
  pbicycle_rent_id bic_rent.bicycle_rent_id%type,
  pbicycle_id bic_rent.bicycle_id%type,
  puser_id bic_rent.user_id%type,
  pvoucher_id bic_rent.voucher_id%type,
  pstand_id_start bic_rent.stand_id_start%type,
  pstand_id_end bic_rent.stand_id_end%type,
  pbicycle_distance_once bic_rent.bicycle_distance_once%type,
  prent_start_time varchar2,
  prent_return_time varchar2,
  prent_late_fee bic_rent.rent_late_fee%type
)
IS

  vuser_id bic_rent.user_id%type := puser_id;
  vbicycle_distance_once bic_rent.bicycle_distance_once%type;
  vrent_start_time bic_rent.rent_start_time%type:= TO_DATE(prent_start_time, 'YYYY-MM-DD HH24:MI:SS');
  vrent_return_time bic_rent.rent_return_time%type := TO_DATE(prent_return_time, 'YYYY-MM-DD HH24:MI:SS');

  vgap number ;
  vtransf number;

BEGIN
  INSERT INTO bic_rent (
    bicycle_rent_id, bicycle_id, user_id, voucher_id, stand_id_start, stand_id_end,
    bicycle_distance_once, rent_start_time, rent_return_time, rent_late_fee
  )
```

# 회원별 이용내역 업데이트 기능 구현

```
VALUES (  
    pbicycle_rent_id, pbicycle_id, puser_id, pvoucher_id, pstand_id_start, pstand_id_end,  
    pbicycle_distance_once, vrent_start_time, vrent_return_time, prent_late_fee  
);
```

```
SELECT SUM ( bicycle_distance_once) INTO vbicycle_distance_once  
FROM bic_rent  
WHERE user_id = puser_id;
```

```
SELECT COUNT(*) * 100 INTO vtransf  
FROM transf  
WHERE user_id = puser_id;
```

```
SELECT SUM ( CEIL((rent_return_time - rent_start_time) * 24 * 60) ) INTO vgap  
FROM bic_rent  
WHERE user_id = puser_id;
```

업데이트문



```
UPDATE user_using  
SET user_total_mileage = vtransf  
WHERE user_id = puser_id;
```

```
UPDATE user_using  
SET user_used_min = vgap  
WHERE user_id = puser_id;
```

```
UPDATE user_using  
SET user_used_km = vbicycle_distance_once  
WHERE user_id = puser_id;  
END;
```

# 회원별 이용내역 업데이트 실행

```
EXEC ins_update('R-32', 'SPB-001', 'ABC123', 'P-1', 'ST-001', 'ST-001', 3.5, '2023-04-13 15:20:25', '2023-04-13 16:20:00', 0);
```

-> 이용내역 업데이트 실행시

31 R-31	SPB-001	ABC123	P-1	ST-001	ST-001	3.5	23/04/13	23/04/13	0
32 R-32	SPB-001	ABC123	P-1	ST-001	ST-001	3.5	23/04/13	23/04/13	0

-> 업데이트 실행 후 출력 내용

물적 자원 관리기능 구현

# 자전거 상태 조회 기능 구현 & 실행

자전거의 상태를 입력하고 실행하면

```
EXEC sel_bike_status('수리중');
```

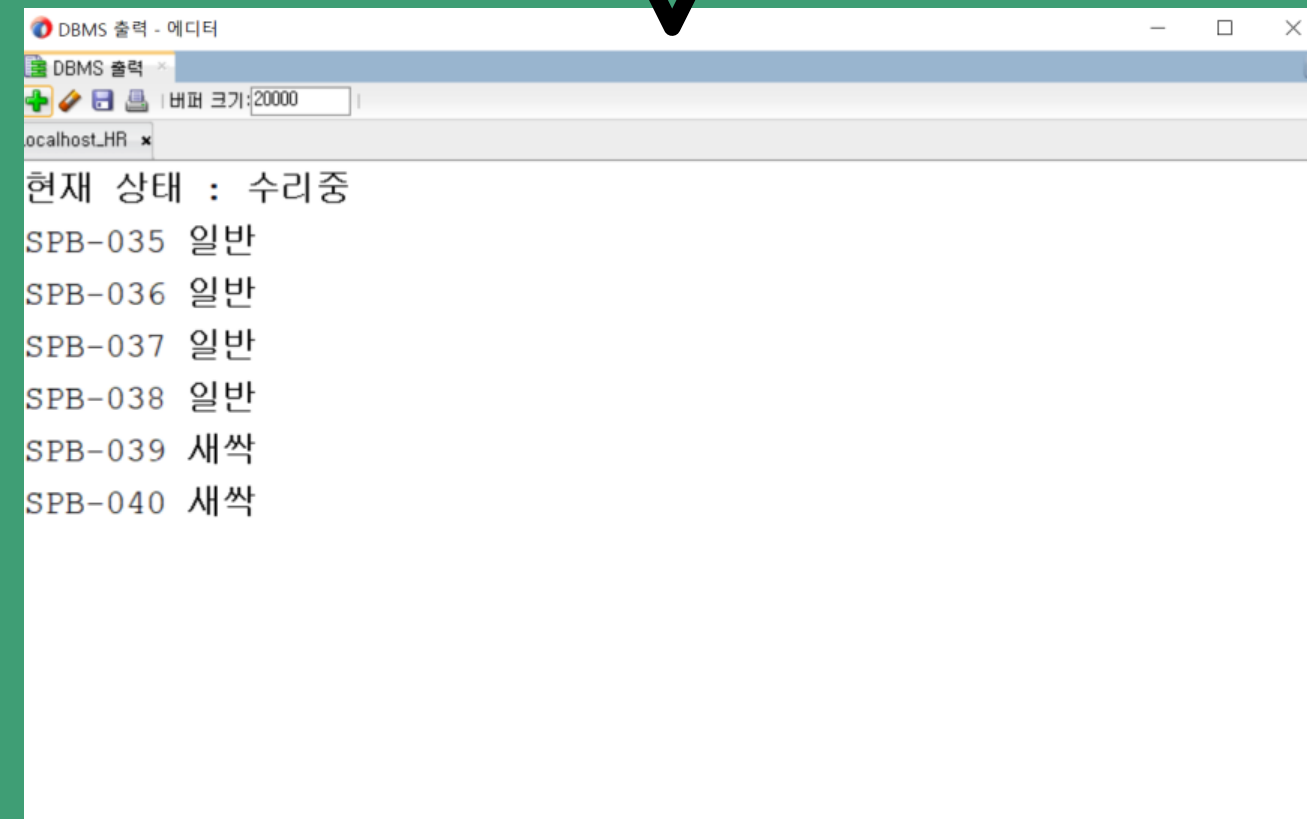
입력한 키워드와 일치하는 상태를  
가진 자전거 정보를 출력

```
CREATE OR REPLACE PROCEDURE sel_bike_status
(
  pbike_status bic_info.bicycle_status%type
)
IS
  vbike_id bic_info.bicycle_id%type;
  vbicycle_model bic_info.bicycle_model%type;
  vbicycle_status bic_info.bicycle_status%type;

  CURSOR vcursor IS(
    SELECT bicycle_id, bicycle_model , bicycle_status
    FROM bic_info
    WHERE bicycle_status = pbike_status
  );
BEGIN

  DBMS_OUTPUT.PUT_LINE('현재 상태 : ' || pbike_status);
  OPEN vcursor;
  LOOP
    FETCH vcursor INTO vbike_id, vbicycle_model , vbicycle_status;
    EXIT WHEN vcursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(vbike_id || ' ' || vbicycle_model);
  END LOOP;
  CLOSE vcursor;
END;
```

코드 상세



```
DBMS 출력 - 에디터
DBMS 출력
버퍼 크기: 20000
localhost_HR
현재 상태 : 수리중
SPB-035 일반
SPB-036 일반
SPB-037 일반
SPB-038 일반
SPB-039 새싹
SPB-040 새싹
```

# 자전거 추가 기능 구현 & 실행

```
CREATE OR REPLACE PROCEDURE ins_bic_info
(
  pbicycle_id bic_info.bicycle_id%type,
  pbicycle_model bic_info.bicycle_model%type,
  pbicycle_reg_date bic_info.bicycle_reg_date%type,
  pbicycle_stand bic_info.bicycle_stand%type,
  pbicycle_status bic_info.bicycle_status%type
)
IS
  vstand_parking_capacity number;
  vstand_current_storage number;
BEGIN
  INSERT INTO bic_info
  VALUES (
    pbicycle_id, pbicycle_model, pbicycle_reg_date, pbicycle_stand
    ,pbicycle_status);

  IF pbicycle_status = '대여중' THEN
    UPDATE bic_std
    SET stand_current_storage = stand_current_storage + 1
    WHERE stand_name = pbicycle_stand;
  END IF;

  SELECT stand_current_storage INTO vstand_current_storage
  FROM bic_std
  WHERE stand_name = pbicycle_stand;

  SELECT stand_parking_capacity INTO vstand_parking_capacity
  FROM bic_std
  WHERE stand_name = pbicycle_stand;

  IF vstand_current_storage > vstand_parking_capacity THEN
    DBMS_OUTPUT.PUT_LINE('거치소의 자전거 최대치보다 많습니다');
    ROLLBACK;
  END IF;
END;
```

코드 상세

## 기존 자전거 정보에서

	BICYCLE_ID	BICYCLE_MODEL	BICYCLE_REG_DATE	BICYCLE_STAND	BICYCLE_STATUS
46	SPB-046	일반	19/02/11	개봉푸르지오아파트 상가	방치접수됨
47	SPB-047	일반	22/10/06	강남 코엑스 앞	방치접수됨
48	SPB-048	새싹	19/04/03	MBC 앞	문실
49	SPB-049	일반	20/11/07	광화문역 1번출구 앞	문실
50	SPB-050	일반	23/04/01	가락1동주민센터	문실



## 추가 프로시저를 실행하면?

```
execute ins_bic_info('SPB-051','일반','2023-12-14','가락1동주민센터','대여중');
```



## 자전거 정보가 추가된 모습!

50	SPB-050	일반	23/04/01	가락1동주민센터	문실
51	SPB-051	일반	23/12/14	가락1동주민센터	대여중



# 대여소 추가 기능 구현 코드

```
CREATE OR REPLACE PROCEDURE add_bic_std (  
    p_stand_id IN VARCHAR2,  
    p_stand_reg_date IN DATE,  
    p_stand_name IN VARCHAR2,  
    p_stand_address IN VARCHAR2,  
    p_stand_latitude IN NUMBER,  
    p_stand_longitude IN NUMBER,  
    p_stand_parking_capacity IN NUMBER,  
    p_stand_current_storage IN NUMBER DEFAULT 0  
)  
IS  
BEGIN  
    INSERT INTO bic_std (  
        stand_id, stand_reg_date, stand_name,  
        stand_address, stand_latitude, stand_longitude,  
        stand_parking_capacity, stand_current_storage  
    ) VALUES (  
        p_stand_id, p_stand_reg_date, p_stand_name,  
        p_stand_address, p_stand_latitude, p_stand_longitude,  
        p_stand_parking_capacity, p_stand_current_storage  
    );  
    DBMS_OUTPUT.PUT_LINE('대여소가 추가되었습니다.');
```

EXCEPTION

```
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('대여소 추가 중 오류가 발생했습니다.');
```

END;



# 대여소 추가 기능 실행

## 추가 전 : 대여소 현황

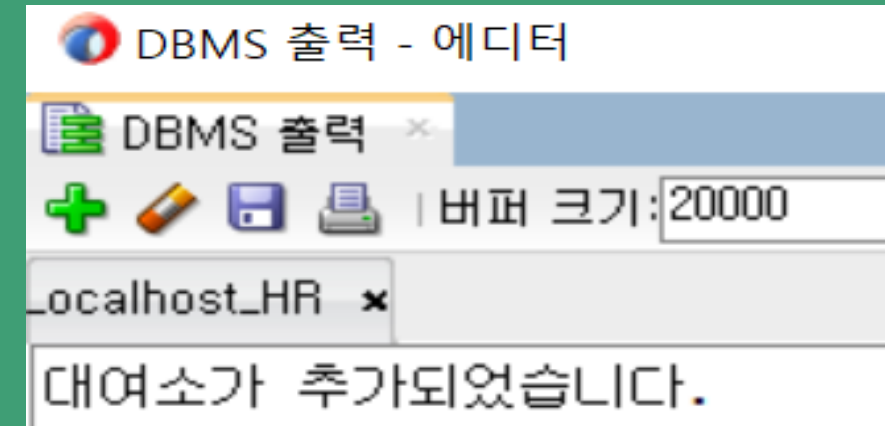
	STAND_ID	STAND_REG_DATE	STAND_NAME	STAND_ADDRESS	STAND_LATITUDE	STAND_LONGITUDE	STAND_PARKING_CAPACITY	STAND_CURRENT_STORAGE
1	ST-001	17/05/11	개봉푸르지오아파트 상가	서울특별시 구로구 남부순환로 775 개봉푸르지오	37.501705	126.847282	12	0
2	ST-002	20/04/06	MBC 앞	서울특별시 마포구 성암로 267 (주) 문화방송	37.581947	126.890297	10	0
3	ST-003	20/12/30	가락1동주민센터	서울특별시 송파구 양재대로 925 가락1동 주민센터	37.496511	127.110329	9	1
4	ST-004	15/10/07	광화문역 1번출구 앞	서울특별시 종로구 세종대로 지하189 세종로공원	37.571769	126.974662	8	0
5	ST-005	18/07/31	강남 코엑스 앞	서울 강남구 영동대로 513	37.512985	127.058947	10	0

## 프로시저를 실행하면

```
EXECUTE add_bic_std('ST-006', '2022-10-12', '방아삼거리', '서울특별시 강동구 둔촌동 118-2', 37.523281, 127.146133, 8, 0);
```

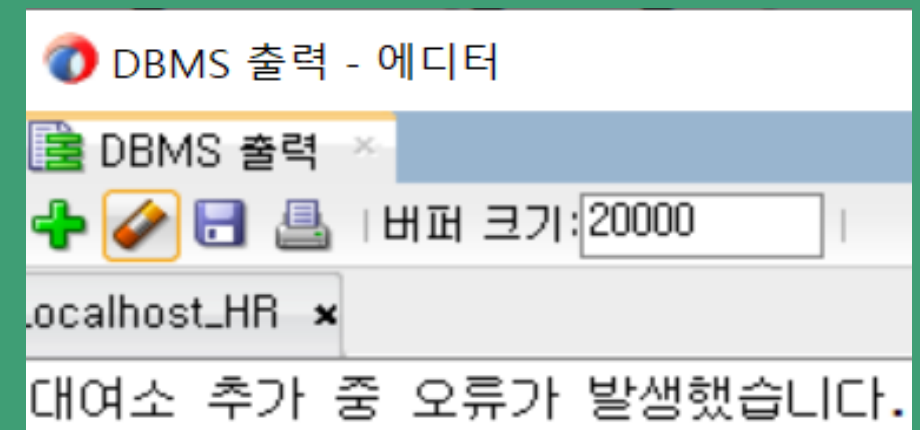
## 추가 후 : 대여소 현황에서 추가 대여소 확인 가능

	STAND_ID	STAND_REG_DATE	STAND_NAME	STAND_ADDRESS	STAND_LATITUDE	STAND_LONGITUDE	STAND_PARKING_CAPACITY	STAND_CURRENT_STORAGE
1	ST-001	17/05/11	개봉푸르지오아파트 상가	서울특별시 구로구 남부순환로 775 개봉푸르지오	37.501705	126.847282	12	0
2	ST-002	20/04/06	MBC 앞	서울특별시 마포구 성암로 267 (주) 문화방송	37.581947	126.890297	10	0
3	ST-003	20/12/30	가락1동주민센터	서울특별시 송파구 양재대로 925 가락1동 주민센터	37.496511	127.110329	9	1
4	ST-004	15/10/07	광화문역 1번출구 앞	서울특별시 종로구 세종대로 지하189 세종로공원	37.571769	126.974662	8	0
5	ST-005	18/07/31	강남 코엑스 앞	서울 강남구 영동대로 513	37.512985	127.058947	10	0
6	ST-006	22/10/12	방아삼거리	서울특별시 강동구 둔촌동 118-2	37.523281	127.146133	8	0



## 잘못된 정보를 입력할시

```
EXECUTE add_bic_std('ST-006', '2022-10-12', 999, 37.523281, 127.146133, 8, 0);
```



관리자 전용 서비스 구현

# 관리자 전용 추가 서비스 기능 구현

## 1. 일별 매출, 일별 이용권 카운트

```
CREATE OR REPLACE PROCEDURE ticket_total
(
    p_pad    t_order.order_paidtime%TYPE
)
IS
    vcount11 NUMBER;
    vcount12 NUMBER;

    vcount211 NUMBER;
    vcount221 NUMBER;

    vcount212 NUMBER;
    vcount222 NUMBER;

    vcount213 NUMBER;
    vcount223 NUMBER;

    vcount214 NUMBER;
    vcount224 NUMBER;

    -----

    vcountpricell1 NUMBER ;
    vcountpricell2 NUMBER ;

    vcountprice211 NUMBER ;
    vcountprice221 NUMBER ;

    vcountprice212 NUMBER ;
```

```
    vcount222 NUMBER;

    vcount213 NUMBER;
    vcount223 NUMBER;

    vcount214 NUMBER;
    vcount224 NUMBER;

    -----

    vcountpricell1 NUMBER ;
    vcountpricell2 NUMBER ;

    vcountprice211 NUMBER ;
    vcountprice221 NUMBER ;

    vcountprice212 NUMBER ;
    vcountprice222 NUMBER ;

    vcountprice213 NUMBER ;
    vcountprice223 NUMBER ;

    vcountprice214 NUMBER ;
    vcountprice224 NUMBER ;

BEGIN
```

다음>>

# 관리자 전용 추가 서비스 실행

## 1. 일별 매출, 일별 이용권 카운트

```
SELECT count(*) INTO vcount11  -- 일일권 11판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,2) = '11';

SELECT count(*) INTO vcount12  -- 일일권 12판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,2) = '12';

-----

SELECT count(*) INTO vcount211  -- 일일권 211판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '211';

SELECT count(*) INTO vcount221  -- 일일권 221판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '221';

SELECT count(*) INTO vcount212  -- 일일권 212판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '212';

SELECT count(*) INTO vcount222  -- 일일권 222판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '222';
```

```
SELECT count(*) INTO vcount213  -- 일일권 213판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '213';
```

```
SELECT count(*) INTO vcount223  -- 일일권 223판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '223';
```

```
SELECT count(*) INTO vcount214  -- 일일권 214판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '214';
```

```
SELECT count(*) INTO vcount224  -- 일일권 224판매 개수
FROM t_order
WHERE order_paidtime = p_pad AND substr(ticket_code , 1,3) = '224';
```

```
vcountpricell := vcount11 *1000;
vcountpricel2 := vcount12 *2000;
```

```
vcountprice211 := vcount211 *3000;
vcountprice221 := vcount221 *4000;
```

# 관리자 전용 추가 서비스 기능 구현

## 1. 일별 매출, 일별 이용권 카운트

```
vcountprice212 := vcount212 *5000;
vcountprice222 := vcount222 *7000;

vcountprice213 := vcount213 *15000;
vcountprice223 := vcount223 *20000;

vcountprice214 := vcount214 * 30000;
vcountprice224 := vcount224 * 40000;

DBMS_OUTPUT.PUT_LINE(' [이용권종류] || ' [이용권 코드] || ' [이용권 판매 수량] || '[일간 매출]');

DBMS_OUTPUT.PUT_LINE(' 일일권 || ' || '11' || ' || vcount11 || ' || vcountprice11);
DBMS_OUTPUT.PUT_LINE(' 일일권 || ' || '12' || ' || vcount12 || ' || vcountprice12);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '211' || ' || vcount211 || ' || vcountprice211);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '221' || ' || vcount221 || ' || vcountprice221);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '212' || ' || vcount212 || ' || vcountprice212);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '222' || ' || vcount222 || ' || vcountprice222);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '213' || ' || vcount213 || ' || vcountprice213);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '223' || ' || vcount223 || ' || vcountprice223);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '214' || ' || vcount214 || ' || vcountprice214);
DBMS_OUTPUT.PUT_LINE(' 정기권 || ' || '224' || ' || vcount224 || ' || vcountprice224);

--EXCEPTION
END;
EXECUTE ticket_total ('2022-12-09');
```

# 관리자 전용 추가 서비스 실행

## 1. 일별 매출, 일별 이용권 카운트

[이용권종류]	[이용권 코드]	[이용권 판매 수량]	[일간 매출]
일일권	11	0	0
일일권	12	0	0
정기권	211	0	0
정기권	221	0	0
정기권	212	0	0
정기권	222	0	0
정기권	213	0	0
정기권	223	0	0
정기권	214	1	30000
정기권	224	0	0

# 관리자 전용 추가 서비스 기능 구현

## 2. 회원연체료 조회

```
CREATE OR REPLACE PROCEDURE late_rental_user_inquire
IS
    v_bicycle_rent_id bic_rent.bicycle_rent_id%TYPE;
    v_bicycle_id bic_rent.bicycle_id%TYPE;
    v_user_id bic_rent.user_id%TYPE;
    v_rent_start_time bic_rent.rent_start_time%TYPE;
    v_rent_return_time bic_rent.rent_return_time%TYPE;
    v_rent_late_fee bic_rent.rent_late_fee%TYPE;
    v_ticket_code u_info.ticket_code%TYPE;
    CURSOR vcur_rent_inquire IS (
        SELECT r.bicycle_rent_id, r.bicycle_id, r.user_id, r.rent_start_time, r.rent_return_time, r.rent_late_fee, i.ticket_code
        FROM bic_rent r
        JOIN u_info i ON r.user_id = i.user_id
        WHERE r.rent_late_fee > 0
    );
BEGIN
    DBMS_OUTPUT.PUT_LINE( '회원연체료 조회' || chr(10) || '출력일자 : ' || SYSDATE || chr(10) ||
        '회원 아이디' || chr(9) || '자전거 식별번호' || chr(9) || '이용권 코드' || chr(9) || '대여 시작 일시'
        || chr(9) || chr(9) || '대여 반납 일시' || chr(9) || chr(9) || '연체료' || chr(10) ||
        '-----'
    );
END;
```



# 관리자 전용 추가 서비스 실행

## 2. 회원연체료 조회

```
OPEN vcur_rent_inquire;
LOOP
FETCH vcur_rent_inquire INTO v_bicycle_rent_id, v_bicycle_id, v_user_id, v_rent_start_time, v_rent_return_time, v_rent_late_fee, v_ticket_code;
  EXIT WHEN vcur_rent_inquire%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE (
    v_user_id || chr(9) || chr(9) || v_bicycle_id || chr(9) || chr(9) || chr(9) || v_ticket_code
    || chr(9) || chr(9) || chr(9) || TO_CHAR( v_rent_start_time, 'YYYY-MM-DD HH24:MI:SS' )
    || chr(9) || TO_CHAR( v_rent_return_time, 'YYYY-MM-DD HH24:MI:SS' ) || chr(9) || v_rent_late_fee
  );
END LOOP;
CLOSE vcur_rent_inquire;
END;

EXEC late_rental_user_inquire;
```

<<실행>>

회원연체료 조회						
출력일자 : 23/04/09						
회원 아이디	자전거 식별번호	이용권 코드	대여 시작 일시	대여 반납 일시	연체료	
-----						-----



# 관리자 전용 추가 서비스 기능 구현

## 3. 관리자가 1대 1질문 답변

```
CREATE OR REPLACE PROCEDURE check_my_qna
(
    p_login_id IN qna.user_id%TYPE
    , p_keyword IN qna.inquiry_title%TYPE
)
IS
    c_inquiry SYS_REFCURSOR;
    v_inquiry_title qna.inquiry_title%TYPE := p_keyword;
    v_content qna.inquiry_contents%TYPE := p_keyword;
    v_inquiry_id answ.inquiry_id%TYPE;
    v_user_id qna.user_id%TYPE := p_login_id;
    v_reg_date qna.inquiry_reg_date%TYPE;
    v_inquiry_status VARCHAR2(20);

BEGIN
    DBMS_OUTPUT.PUT_LINE( CHR(10) || ' 검색 내용을 입력해주세요. ' || CHR(10) );
    DBMS_OUTPUT.PUT_LINE( CHR(9) || CHR(9) || CHR(9) || CHR(9) || ' [ 조회된 문의글 목록 ] ' || CHR(9) || CHR(9) || CHR(9) );
    DBMS_OUTPUT.PUT_LINE( CHR(9) || CHR(9) || CHR(9) || ' [ 제목 ] ' || CHR(9) || CHR(9) || CHR(9) || ' [ 답변 상태 ] ' || CHR(9) || ' [ 날짜 ] ');
```

다음 >>실행으로

# 관리자 전용 추가 서비스 실행

## 3. 관리자가 1대 1질문 답변

<<실행>>

검색 내용을 입력해주세요.

[ 조회된 문의글 목록 ]

[ 제목 ] [ 답변 상태 ] [ 날짜 ]

해당 데이터가 없습니다

검색 내용을 입력해주세요.

[ 조회된 문의글 목록 ]

[ 제목 ] [ 답변 상태 ] [ 날짜 ]

해당 데이터가 없습니다

검색 내용을 입력해주세요.

[ 조회된 문의글 목록 ]

[ 제목 ] [ 답변 상태 ] [ 날짜 ]

해당 데이터가 없습니다

검색 내용을 입력해주세요.

[ 조회된 문의글 목록 ]

[ 제목 ] [ 답변 상태 ] [ 날짜 ]

해당 데이터가 없습니다

# 관리자 전용 회원정보 조회 기능 구현

```
CREATE OR REPLACE PROCEDURE user_info_select
(
    p_user_id IN u_info.user_id%TYPE
)
IS
    v_user_id u_info.user_id%TYPE;
    v_ticket_code u_info.ticket_code%TYPE;
    v_user_password u_info.user_password%TYPE;
    v_user_name u_info.user_name%TYPE;
    v_user_email u_info.user_email%TYPE;
    v_user_phone_num u_info.user_phone_num%TYPE;
    v_user_gender u_info.user_gender%TYPE;
    v_user_birth u_info.user_birth%TYPE;
    v_user_weight u_info.user_weight%TYPE;
    v_user_level u_info.user_level%TYPE;
    v_level_type VARCHAR2(10);
    v_ticket_type VARCHAR2(10);
    CURSOR v_cur_info IS (
        SELECT user_id, ticket_code, user_password, user_name, user_email, user_phone_num, user_gender, user_birth, user_weight, user_level
        FROM u_info
        WHERE user_id = p_user_id AND p_user_id NOT LIKE '%비회원%'
    );
```

-> 다음

# 관리자 전용 회원정보 조회 기능 구현

```
BEGIN

    DBMS_OUTPUT.PUT_LINE( '회원정보 조회' || chr(10) || '출력일자 : ' || SYSDATE || chr(10) ||
        '[회원아이디]' || ' ' || '[이용권식별번호]' || ' ' || '[이용권유형]' || ' ' || '[회원비밀번호]' || ' ' || '[회원이름]' || ' ' ||
        '[회원이메일]' || ' ' || '[회원전화번호]' || ' ' || '[회원성별]' || ' ' || '[회원생년월일]' || ' ' ||
        '[회원체중]' || ' ' || '[회원레벨]' || chr(10) ||
        '-----' );

    OPEN v_cur_info;
    LOOP
        FETCH v_cur_info INTO v_user_id, v_ticket_code, v_user_password, v_user_name, v_user_email, v_user_phone_num, v_user_gender, v_user_birth, v_user_weight, v_user_level;
        EXIT WHEN v_cur_info%NOTFOUND;
        IF SUBSTR( v_ticket_code, 1, 1 ) = 1 THEN v_ticket_type := '일일권';
        ELSIF SUBSTR( v_ticket_code, 1, 1 ) = 2 THEN v_ticket_type := '정기권';
        END IF;
        IF v_user_level = 1 THEN
            v_level_type := '회원';
        ELSIF v_user_level = 2 THEN
            v_level_type := '관리자';
        END IF;
        DBMS_OUTPUT.PUT_LINE(
            ' ' || v_user_id || chr(9) || chr(9) || chr(9) || v_ticket_code || chr(9) || chr(9) || chr(9) || v_ticket_type || chr(9) || chr(9) || v_user_password || chr(9) ||
            v_user_name || chr(9) || v_user_email || chr(9) || v_user_phone_num || chr(9) || v_user_gender || chr(9) ||
            v_user_birth || chr(9) || chr(9) || v_user_weight || chr(9) || chr(9) || chr(9) || v_level_type
        );
    END LOOP;
    IF v_cur_info%ROWCOUNT = 0 THEN DBMS_OUTPUT.PUT_LINE('해당하는 데이터가 없습니다.');
```

```
    END IF;
    CLOSE v_cur_info;
END;
```

# 관리자 전용 회원정보 조회 실행

`EXECUTE user_info_select ('ABC123');` 회원 아이디는 중복되지 않기에 회원 아이디와 일치하는 정보 확인 가능

회원정보 조회

출력일자 : 23/04/09

[회원아이디] [이메일주소] [이름] [회원등급] [회원비밀번호] [회원이름] [회원이메일] [회원전화번호] [회원성명] [회원생년월일] [회원체중] [회원레벨]

ABC123	211	정기권	ESFP	홍찬기	hcgghcg17@naver.com	010-9998-1635	남자 10/13/1997	100	관리자
--------	-----	-----	------	-----	---------------------	---------------	---------------	-----	-----

```
EXECUTE user info select ('nullMember');
```

회원정보 조회

출력일자 : 23/04/09

[회원아이디] [이메일주소] [이름] [회원포인트] [회원등급] [회원아이디] [회원이메일] [회원전화번호] [회원성명] [회원생년월일] [회원체중] [회원레벨]

해당하는 데이터가 없습니다.

존재하지 않는 회원 아이디 검색 시 해당되는 데이터 없음 문구 출력

Q & A

# 프로젝트 후기

# 프로젝트 후기

## 하 동호

1주일이라는 긴 시간동안 팀원들이 힘을 합쳐 원하는 결과물을 만들었다고 생각합니다.

## 김 예지

처음에는 오라클을 배우는 것이 매우 어려웠습니다. 하지만, 조원분들과 함께한 스터디를 통해 내용을 복습하면서 많은 도움을 받았습니다. 이번 스터디와 프로젝트를 진행하면서 많은 것을 배우고 성장할 수 있었던 것에 대해 감사한 마음을 가지고 있습니다. 3조분들 모두 고생했어요 !!

## 박 진용

프로젝트를 받았을때에 어떻게 요구분석부터 쿼리까지 어떻게 해야하나 걱정이 많았는데 어려움이 있을때 팀원들이랑 같이 하면서 프로젝트를 끝낼 수 있었습니다~~@@

## 박 현주

요구분석부터 모델링, 쿼리 작성까지 처음에는 막막하게 느껴지던 것들이 프로젝트가 진행됨에 따라 형태가 잡히고 구체화 되어 가는 점이 좋았습니다. 어려운 부분은 팀원들과 도와가며 진행할 수 있었던 프로젝트라 더욱 배울 점이 많았던 것 같습니다 !!





# 프로젝트 후기

## 이 태규

이번 프로젝트는 저번 프로젝트와 달리 명확히 정해진 것 없이 주제부터 정하고 각 테이블 간의 관계가 있어 좀 더 복잡했었던 것 같습니다. 하지만 각자 발생하는 문제를 해결하고 구상한 기능이 구현되는 재미가 있었고 여러 문제를 팀원들과 같이 고민하고 해결하는 과정이 즐거웠습니다.

## 하 지예

부족한 실력이지만 조원분들이 많이 도와주고 격려해주어서 덕분에 잘 마무리할 수 있었습니다. 프로젝트 준비 기간 동안 다들 정말 고생 많으셨고 진심으로 감사했습니다.



## 홍 성철

첫 데이터베이스 프로젝트라서 어려운 점이 많았지만, 팀원들 모두 적극적으로 참여하고 좋은 케미를 보여주셔서 많이 배울 수 있었습니다. 프로젝트를 진행하면서 느낀 점은 제대로 된 요구분석을 설계하는 것이 개발에 얼마나 중요한지를 깨닫게 되어 좋았습니다.

## 홍 찬기

이번 프로젝트를 하며 초반에 적극적으로 참여하지 못했는데 중반부터 적극적으로 참여한 것 같아서 다행이었던 것 같습니다. 좋은 조원들을 만나서 재밌게 프로젝트를 진행하고 마무리 하여 배운 것도 많은 시간이었습니다.



# THANK YOU.

경청해 주셔서 감사합니다.

