



HOSPITAL MANAGEMENT PROGRAM

3조 김성혜, 윤종현, 지영훈, 한정은
221116-221124

TABLE OF CONTENTS

01

프로젝트 개요

프로젝트 목적
프로젝트 일정

02

팀원 소개 및 역할

팀원 소개 및 역할

03

프로젝트 수행 절차 및 결과

프로젝트 수행 절차

-ERD, 테이블 명세서

프로젝트 수행 결과

-뷰, 함수/프로시저, 트리거, 파일

04

자체 평가 의견

자체 평가 의견

01

프로젝트 개요

프로젝트 목적
프로젝트 일정

프로젝트 목적



병원 직원 관리

진료과, 직급
의사, 간호사



예약 및 진료 관리

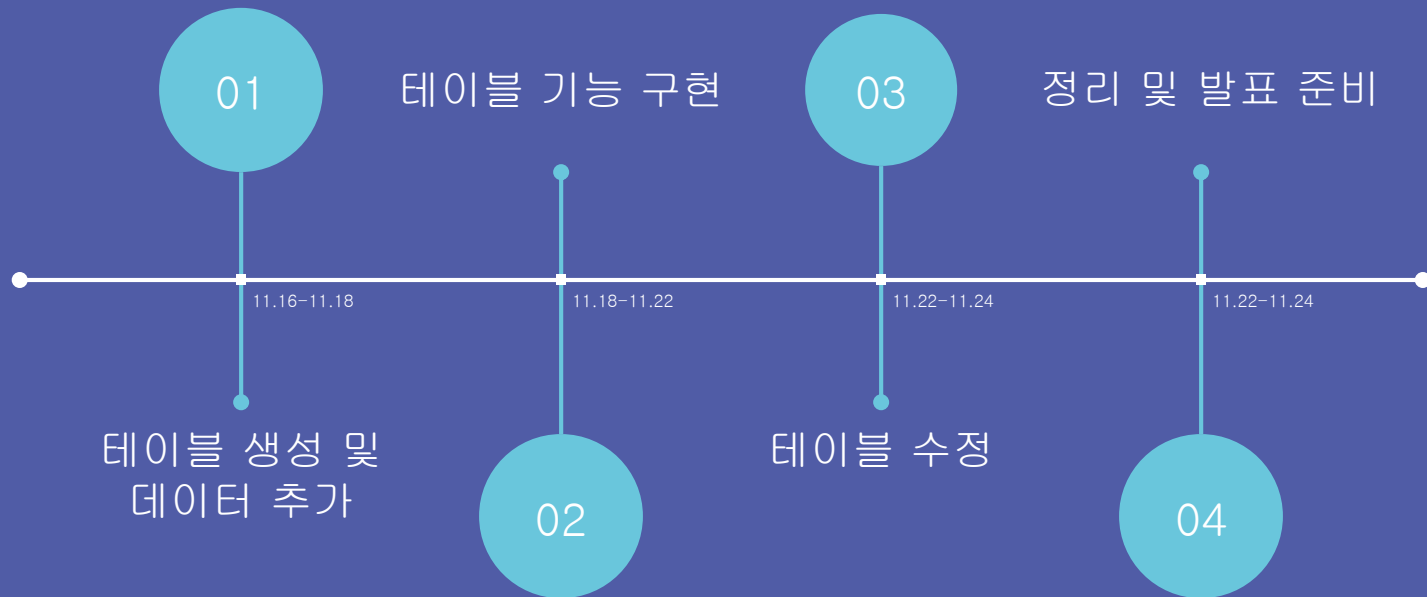
환자, 예약, 진료



입원 및 회계 관리

입원, 입원실

프로젝트 일정





02

팀원 소개 및 역할

팀원 소개 및 역할



팀원 역할 소개

지영훈 (조장)

테이블 생성 및 데이터 추가
테이블 기능 구현
발표 자료 준비



3조

윤종현

테이블 생성 및 데이터 추가
테이블 기능 구현
발표 자료 준비 및 발표



김성혜

테이블 생성 및 데이터 추가
테이블 기능 구현
발표 자료 준비 및 PPT 작성



한정은

테이블 생성 및 데이터 추가
테이블 기능 구현
발표 자료 준비 및 발표



03

프로젝트 수행 절차 및 결과

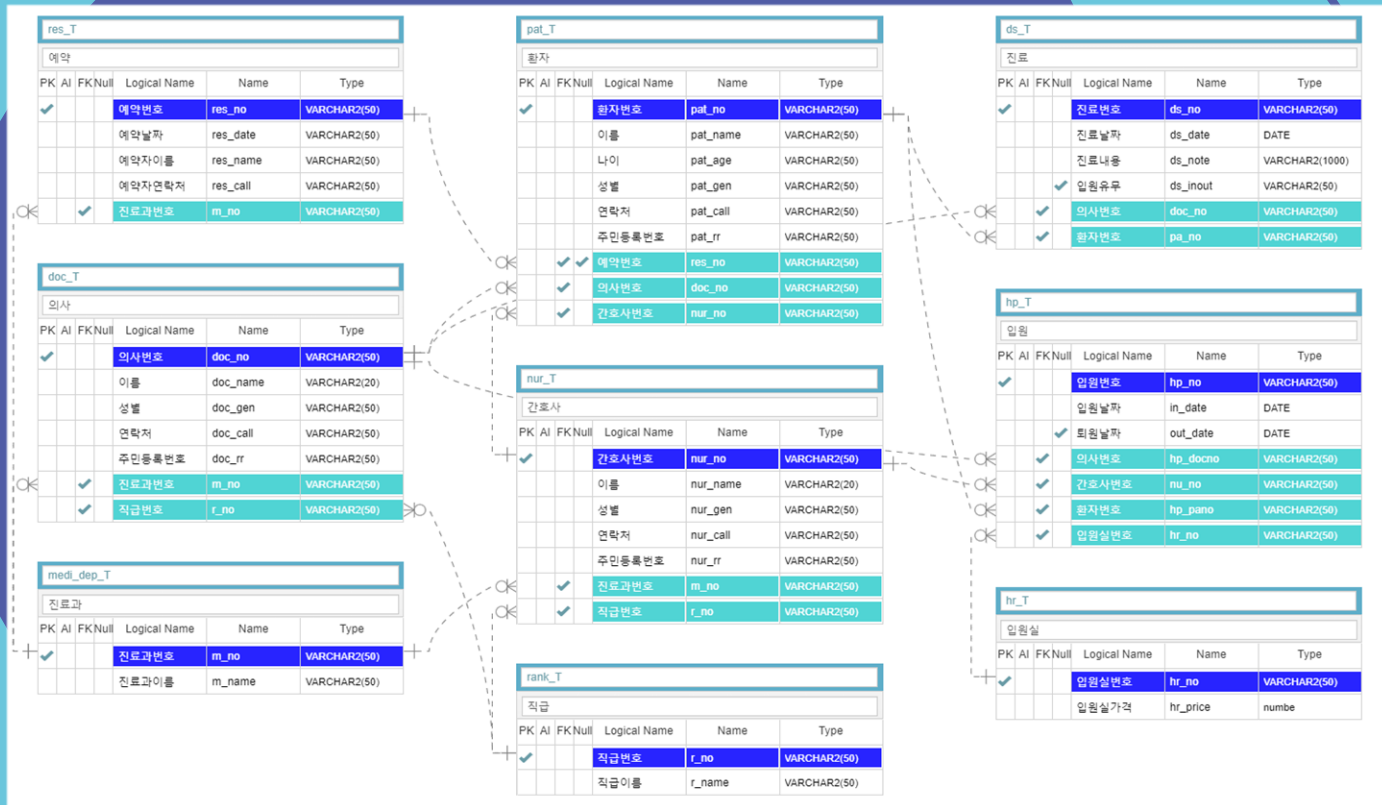
프로젝트 수행 절차

-ERD, 테이블 명세서

프로젝트 수행 결과

-뷰, 함수/프로시저, 트리거, 파일

프로젝트 수행 절차 - ERD



프로젝트 수행 절차 - 테이블 명세서



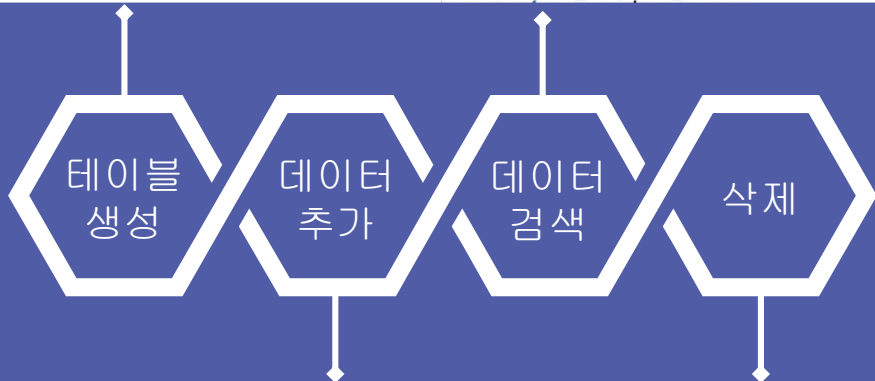
테이블명	doc_T									
논리명	의사									
테이블 설명										
PRIMARY KEY	doc_no									
FOREIGN KEY	m_no, r_no									
INDEX										
UNIQUE INDEX										
NO	PK	AI	FK	NULL	논리명	컬럼명	TYPE	DEFAULT	설명	참조 테이블
1	Y				의사번호	doc_no	VARCHAR2(50)			
2					이름	doc_name	VARCHAR2(20)			
3					성별	doc_gen	VARCHAR2(50)			
4					연락처	doc_call	VARCHAR2(50)			
5					주민등록번호	doc_rr	VARCHAR2(50)			
6			Y		진료과번호	m_no	VARCHAR2(50)			medi_dep_T
7			Y		직급번호	r_no	VARCHAR2(50)			rank_T
테이블명	ds_T									
논리명	진료									
테이블 설명										
PRIMARY KEY	ds_no									
FOREIGN KEY	doc_no, pa_no									
INDEX										
UNIQUE INDEX										
NO	PK	AI	FK	NULL	논리명	컬럼명	TYPE	DEFAULT	설명	참조 테이블
1	Y				진료번호	ds_no	VARCHAR2(50)			
2					진료날짜	ds_date	DATE			
3					진료내용	ds_note	VARCHAR2(1000)			
4			Y		입원유무	ds_inout	VARCHAR2(50)	외래		
5		Y			의사번호	doc_no	VARCHAR2(50)			doc_T
6		Y			환자번호	pa_no	VARCHAR2(50)			pat_T

프로젝트 수행 결과-기본

```
CREATE TABLE rank_T  
(  
  r_no  VARCHAR2(50) NOT NULL,  
  r_name VARCHAR2(50) NOT NULL,  
  PRIMARY KEY (r_no)  
);
```

```
select pat_name "이름", hr_no "입원실"  
from pat_t join hp_t on pat_no=hp_pano  
group by hr_no, pat_name  
order by hr_no, pat_name;
```

	이름	입원실
1	성조기	101
2	조기	101
3	이민	201
4	이민	201
5	정기	201
6	정기	301
7	정기	301
8	정기	401
9	정기	501
10	정기	501



```
insert into medi_dep_T values('M001','피부과');  
insert into medi_dep_T values('M002','소아과');  
insert into medi_dep_T values('M003','정형외과');  
insert into medi_dep_T values('M004','치과');  
insert into medi_dep_T values('M005','안과');
```

```
drop table ds_T;  
drop table hp_T;  
delete from ds_t where ds_no='DS001';  
delete from ds_t where ds_no='DS002';  
drop function random_call;
```

프로젝트 수행 결과-기본-데이터 추가

```
CREATE OR REPLACE FUNCTION random_pr RETURN VARCHAR2 IS
```

```
f_rr VARCHAR2(50);
f_year NUMBER := trunc(dbms_random.value(70, 95));
f_month NUMBER := trunc(dbms_random.value(1, 12));
f_day NUMBER := trunc(dbms_random.value(01, 30));
f_fr NUMBER := trunc(dbms_random.value(1, 2));
f_se NUMBER := trunc(dbms_random.value(100000, 999999));
BEGIN
    f_rr := to_char(f_year)
        || to_char(f_month, 'FM00')
        || to_char(f_day, 'FM00')
        || '-'
        || to_char(f_fr)
        || to_char(f_se);

    RETURN f_rr;
END;
/
```

```
CREATE OR REPLACE FUNCTION random_call RETURN VARCHAR2 IS
```

```
f_call VARCHAR2(50);
f_first VARCHAR2(3) := '010';
f_second NUMBER := trunc(dbms_random.value(0001, 9999));
f_third NUMBER := trunc(dbms_random.value(0001, 9999));
BEGIN
    f_call := f_first
        || to_char(f_second, 'FM0000')
        || to_char(f_third, 'FM0000');

    RETURN f_call;
END;
/
```

```
insert into doc_T values('D001','고영호','남',random_call,'730906-1831367','M001','R001');
```

```
alter table pat_T modify(pat_rr VARCHAR2(50) unique);
alter table nur_T modify(nur_rr VARCHAR2(50) unique);
alter table doc_T modify(doc_rr VARCHAR2(50) unique);
alter table pat_T modify(pat_gen VARCHAR2(50) check (pat_gen in('남','여')));
alter table nur_T modify(nur_gen VARCHAR2(50) check (nur_gen in('남','여')));
alter table doc_T modify(doc_gen VARCHAR2(50) check (doc_gen in('남','여')));
alter table ds_T modify(ds_inout VARCHAR2(50) check (ds_inout in('입원','외래')));
```

주민번호
랜덤함수

전화번호
랜덤함수

제약조건

프로젝트 수행 결과-활용

1. 진료과별 의사, 간호사, 환자 이름 보기
2. 진료과별 진료 기록 보기
3. 진료과별 외래 환자 수, 입원 환자 수 보기
4. 특정 날짜 사이의 입원 환자 보기

1. 진료 테이블에서 입원이면 입원 테이블 자동 입력
2. 예약 시 의사에 담당 환자가 제일 적은 의사 넣기



1. 특정 날짜 사이의 입원 환자 보기
2. 입원 테이블에서 환자 이름 검색 시 퇴원일 보기
3. 입원료 계산
4. 입원실 입력 시 입원 중인 환자들 이름 보기(+커서)

1. 환자 이름 입력 시 모든 진료 기록 txt 파일로 출력
2. 모든 환자들의 진료 기록을 txt 파일로 출력

프로젝트 수행 결과-뷰1-1

뷰1-1

진료과별 의사, 환자 이름 보기

```
CREATE OR REPLACE VIEW m_d_p_see AS
```

```
SELECT
```

```
    m.m_name "진료과 이름",
```

```
    d.doc_name "의사 이름",
```

```
    p.pat_name "환자 이름"
```

```
FROM
```

```
    medi_dep_t m
```

```
  LEFT OUTER JOIN doc_t  d ON m.m_no = d.m_no
```

```
  LEFT OUTER JOIN pat_t  p ON d.doc_no = p.doc_no
```

```
GROUP BY
```

```
  ROLLUP(m.m_name,
```

```
         d.doc_name,
```

```
         p.pat_name)
```

```
ORDER BY
```

```
    m.m_name,
```

```
    d.doc_name,
```

```
    p.pat_name;
```

0

select*from m_d_p_see;

크립트 출력 x | 질의 결과 x

SQL | 50개의 행이 인출됨(0.004초)

	진료과 이름	의사 이름	환자 이름
1	소아과	박성화	이은만
2	소아과	박재하	(null)
3	소아과	박재하	김명주
4	소아과	박재하	조동호
5	소아과	박재하	(null)
6	소아과	박소장	김지우
7	소아과	조상현	(null)
8	소아과	조상현	(null)
9	소아과	조상현	(null)
10	소아과	이호성	부영림
11	소아과	이호성	조다름
12	소아과	이호성	(null)
13	소아과	조재환	이재환
14	소아과	조재환	(null)
15	소아과	허재	박지후
16	소아과	허재	(null)
17	소아과	(null)	(null)
18	안과	박정아	(null)
19	안과	박정아	(null)
20	안과	임가영	최은미
21	안과	임가영	(null)
22	안과	차향선	최지영
23	안과	차향선	(null)
24	안과	차향선	(null)
25	안과	조규환	송현수
26	안과	조규환	(null)
27	안과	(null)	(null)
28	정형외과	권아름	차민서

프로젝트 수행 결과-뷰1-2

뷰1-2

진료과별 의사, 간호사 이름 보기

```
CREATE OR REPLACE VIEW v_mndoc AS
```

```
SELECT
```

```
    dep.m_name 진료과,  
    d.doc_name 의사이름,  
    n.nur_name 간호사이름
```

```
FROM
```

```
    medi_dep_t dep,  
    doc_t      d,  
    nur_t      n
```

```
WHERE
```

```
    dep.m_no = d.m_no
```

```
    AND substr(d.doc_no, 2) IN ( substr(n.nur_no, 2) - 300 );
```

의사 테이블은 진료과 테이블의 진료과 번호를 참조

의사 테이블과 간호사 테이블의 의사 번호와 간호사 번호는 뒤의 두 문자가 01, 02, 03 순으로 증가
Substr 함수로 뒤의 두문자만 남도록 자른 다음
두문자가 일치하면 출력

0

```
select * from v_mndoc;
```

크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 30x0,008초)

	진료과	의사이름	간호사이름
1	피부과	고정환	장기영
2	피부과	고정환	장기영
3	피부과	고정환	장기영
4	피부과	고정환	장기영
5	피부과	고정환	장기영
6	피부과	고정환	장기영
7	소아과	고정환	장기영
8	소아과	고정환	장기영
9	소아과	고정환	장기영
10	소아과	고정환	장기영
11	소아과	고정환	장기영
12	소아과	고정환	장기영
13	소아과	고정환	장기영
14	소아과	고정환	장기영
15	소아과	고정환	장기영
16	소아과	고정환	장기영
17	소아과	고정환	장기영
18	소아과	고정환	장기영
19	소아과	고정환	장기영
20	소아과	고정환	장기영
21	소아과	고정환	장기영
22	소아과	고정환	장기영
23	소아과	고정환	장기영
24	소아과	고정환	장기영
25	소아과	고정환	장기영

프로젝트 수행 결과-뷰2

뷰2

진료과별 진료 기록 보기

모든 진료과별 진료 기록 보기

원하는 진료과별 진료 기록 보기

```
CREATE VIEW medi_ds AS
```

```
SELECT
```

```
m.m_name    진료과이름,
p.m_no      진료과번호,
d.doc_no    의사번호,
d.ds_date   진료날짜,
d.ds_note   진료내용,
d.pa_no     환자번호,
d.ds_inout  입원or외래
```

```
FROM
```

```
medi_dep_t m,
ds_t       d,
doc_t      p
```

```
WHERE
```

```
m.m_no = p.m_no
```

```
AND p.doc_no = d.doc_no
```

```
ORDER BY
```

```
m.m_name;
```

select * from MEDI_DS order by 진료과번호;

진료과이름	진료과번호	의사번호	진료날짜	진료내용	환자번호	입원OR외래
1 피부과	M001	D012	21/07/17	피부질환	P028	외래
2 피부과	M001	D003	21/07/05	피부염	P028	입원
3 피부과	M001	D027	21/08/02	피부질환	P022	외래
4 피부과	M001	D020	22/07/27	아토피	P018	외래
5 피부과	M001	D003	21/10/25	피부질환	P006	외래
6 피부과	M001	D013	21/07/17	피부질환	P003	입원
7 피부과	M001	D013	21/11/02	굴절	P009	외래
8 피부과	M001	D020	21/07/20	아토피	P009	입원
9 피부과	M001	D013	21/12/27	피부염	P003	외래
10 소아과	M002	D016	22/08/27	소화기	P017	입원
11 소아과	M002	D023	21/12/27	소화기	P004	외래
12 소아과	M002	D008	22/04/27	종합검진	P039	외래
13 소아과	M002	D002	21/09/06	알레르기	P027	외래
14 소아과	M002	D008	21/10/05	호흡기	P021	입원
15 소아과	M002	D008	21/10/09	소화기	P039	외래
16 소아과	M002	D016	21/10/11	인플루엔자	P017	외래
17 소아과	M002	D019	21/10/15	예방주사	P013	외래
18 정형외과	M003	D009	22/04/27	외상치료	P038	입원
19 정형외과	M003	D025	22/04/27	물리치료	P037	외래

```
select * from MEDI_DS where 진료과이름 = '피부과';
select * from MEDI_DS where 진료과이름 = '소아과';
select * from MEDI_DS where 진료과이름 = '정형외과';
select * from MEDI_DS where 진료과이름 = '치과';
select * from MEDI_DS where 진료과이름 = '안과';
```

진료과이름	진료과번호	의사번호	진료날짜	진료내용	환자번호	입원OR외래
안과	M005	D018	21/01/07	안구건조증	P040	외래
안과	M005	D007	21/01/10	안구건조증	P040	외래
안과	M005	D026	21/01/17	녹내장수술	P040	입원
안과	M005	D007	21/02/07	시력검사	P033	외래
안과	M005	D006	21/02/15	시력검사	P033	외래
안과	M005	D018	21/11/27	시력검사	P012	외래
안과	M005	D006	22/01/27	녹내장	P035	입원
안과	M005	D007	22/02/27	시력검사	P033	외래
안과	M005	D018	22/03/27	시력검사	P040	외래

프로젝트 수행 결과-뷰3

뷰3

진료과별 외래 환자 수, 입원 환자 수 보기

```
CREATE OR REPLACE VIEW v_medi AS
```

```
SELECT
```

```
    m.m_name                                "진료과 이름",
```

```
    COUNT(decode(p.pat_no, h.hp_pano, NULL, p.pat_no)) "외래 환자 수",
```

```
    COUNT(h.hp_no)                                "입원 환자 수"
```

```
FROM
```

```
    medi_dep_t m
```

```
    RIGHT OUTER JOIN doc_t  d ON m.m_no = d.m_no
```

```
    RIGHT OUTER JOIN pat_t  p ON d.doc_no = p.doc_no
```

```
    LEFT OUTER JOIN hp_t    h ON p.pat_no = h.hp_pano
```

```
GROUP BY
```

```
    m.m_name;
```

```
select * from v_medi;
```

진료과 이름	외래 환자 수	입원 환자 수
1 안과	2	2
2 피부과	7	3
3 정형외과	7	2
4 소아과	6	2
5 치과	7	1

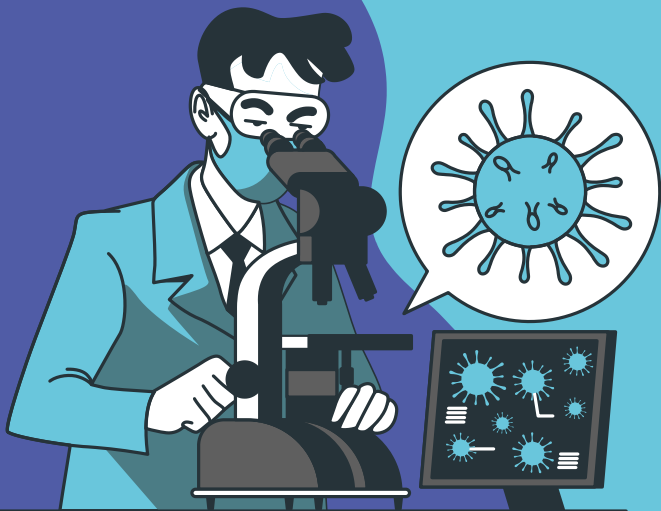
```
select * from v_medi where "진료과 이름" = '안과';
```

진료과 이름	외래 환자 수	입원 환자 수
안과	2	2

프로젝트 수행 결과-뷰4

뷰4

특정 날짜 사이 입원 환자 보기



```
CREATE OR REPLACE VIEW v_date AS  
SELECT
```

```
    pa.pat_name 환자이름,  
    hp.hr_no 입원실,  
    hp.in_date 입원일
```

```
FROM
```

```
    pat_t pa,  
    hp_t hp
```

```
WHERE
```

```
    hp.hp_pano = pa.pat_no  
    AND hp.in_date > '21/01/01'  
    AND hp.in_date < '21/12/12'
```

```
ORDER BY
```

```
    hp.in_date;
```

```
select *from v_date;
```

크립트 출력 x

질의 결과 x

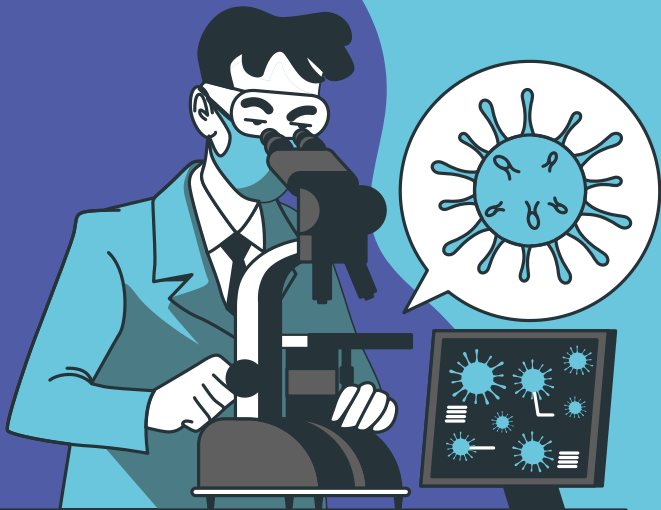
SQL | 인출된 모든 행: 7(0,008초)

	환자이름	입원실	입원일
1	성진호	101	21/01/17
2	이준모	201	21/03/02
3	설시준	401	21/06/30
4	홍성민	501	21/07/05
5	김규환	301	21/07/17
6	김차왕	201	21/07/20
7	소다름	101	21/10/05

프로젝트 수행 결과-프로시저1

프로시저1

특정 날짜 사이 입원 환자 보기



```
CREATE OR REPLACE PROCEDURE HP_DATE_PROC (  
    P_A IN DATE,  
    P_B IN DATE)  
IS  
    P_PANO HP_T.HP_NO%TYPE;  
    P_NAME PAT_T.PAT_NAME%TYPE;  
    P_IN_DATE HP_T.IN_DATE%TYPE;  
    P_OUT_DATE HP_T.OUT_DATE%TYPE;  
BEGIN  
    FOR HP_DATE_BUF IN (SELECT HP_PANO,P_NAME,IN_DATE,OUT_DATE  
        INTO P_PANO,P_NAME,P_IN_DATE,P_OUT_DATE  
        FROM HP_T H  
        WHERE P_A < IN_DATE AND IN_DATE < P_B)  
    loop  
        DBMS_OUTPUT.PUT_LINE('환자번호 : '||HP_DATE_BUF.HP_PANO);  
        DBMS_OUTPUT.PUT_LINE('환자이름 : '||HP_DATE_BUF.P_NAME);  
        DBMS_OUTPUT.PUT_LINE('입원날짜 : '||HP_DATE_BUF.IN_DATE);  
        DBMS_OUTPUT.PUT_LINE('퇴원날짜 : '||HP_DATE_BUF.OUT_DATE);  
    end loop;  
END;  
/  
EXECUTE HP_DATE_PROC('21/01/01','21/12/12');
```

```
=====
```

환자번호 : P015
입원날짜 : 21/03/02
퇴원날짜 : 21/03/17

```
=====
```

환자번호 : P034
입원날짜 : 21/06/30
퇴원날짜 : 21/07/07

```
=====
```

환자번호 : P028
입원날짜 : 21/07/05
퇴원날짜 : 21/07/10

```
=====
```

프로젝트 수행 결과-프로시저2

프로시저2

입원 테이블에서 환자 이름 검색 시 퇴원일 보기

```
execute PROC hp pa name('성진호');
```

스크립트 출력 x

작업이 완료되었습니다. (0.073초)

금일 22/11/24

이미 퇴원한 환자입니다.

```
CREATE OR REPLACE PROCEDURE hr_room_no (
    room IN hp_t.hr_no%TYPE
) IS
    res VARCHAR2(20);
BEGIN
    FOR c_pano IN (
        SELECT
            pa.pat_name
        FROM
            hp_t hp,
            hr_t hr,
            pat_t pa
        WHERE
            hp.hr_no = hr.hr_no
            AND pat_no = hp.hp_pano
            AND hp.hr_no = room
    ) LOOP
        dbms_output.put_line('입원실'
            || room
            || '에 '
            || c_pano.pat_name
            || '님이 입원하고 있습니다.');
    END LOOP;
END;
```

프로젝트 수행 결과-프로시저3

프로시저3

입원료 계산

```
CREATE OR REPLACE PROCEDURE ho_bills (
  pt_name IN pat_t.pat_name%TYPE
) IS

  in_date hp_t.in_date%TYPE;
  out_date hp_t.out_date%TYPE;
  hr_price hr_t.hr_price%TYPE;
  h_day NUMBER;
  res NUMBER;
BEGIN
  SELECT
    h.in_date
  INTO in_date
  FROM
    hp_t h,
    pat_t p
  WHERE
    h.hp_pano = p.pat_no
    AND p.pat_name = pt_name;

  SELECT
    h.out_date
  INTO out_date
  FROM
    hp_t h,
    pat_t p
  WHERE
    h.hp_pano = p.pat_no
    AND p.pat_name = pt_name;
```

```
SELECT
  r.hr_price
INTO hr_price
FROM
  hr_t r,
  hp_t h,
  pat_t p
WHERE
  h.hp_pano = p.pat_no
  AND p.pat_name = pt_name
  AND r.hr_no = h.hr_no;

h_day := trunc(out_date) - trunc(in_date) + 1;
res := h_day * hr_price;
IF out_date IS NULL THEN
  dbms_output.put_line('퇴원일이 미정입니다.');
```

execute ho_bills('이은만');

이은만님의 입원료는 2200000원입니다.

```
ELSIF out_date IS NOT NULL THEN
  dbms_output.put_line(pt_name
    || '님의 입원료는 '
    || res
    || '원입니다.');
```

```
ELSE
  dbms_output.put_line('알 수 없는 에러가 발생했습니다.');
```

```
END IF;

END;
```

프로젝트 수행 결과-프로시저4

프로시저4

입원실 입력 시 입원 중인 환자들 이름 보기(+커서)

```
execute hr_room_no('101');
```

스크립트 출력 x

작업이 완료되었습니다.(0.057초)

입원실101에 성진호님이 입원하고 있습니다.
입원실101에 소다름님이 입원하고 있습니다.

```
CREATE OR REPLACE PROCEDURE hr_room_no (
    room IN hp_t.hr_no%TYPE
) IS
    res VARCHAR2(20);
BEGIN
    FOR c_pano IN (
        SELECT
            pa.pat_name
        FROM
            hp_t hp,
            hr_t hr,
            pat_t pa
        WHERE
            hp.hr_no = hr.hr_no
            AND pat_no = hp.hp_pano
            AND hp.hr_no = room
    ) LOOP
        dbms_output.put_line('입원실'
            || room
            || '에 '
            || c_pano.pat_name
            || '님이 입원하고 있습니다.');    END LOOP;
END;
```

프로젝트 수행 결과-트리거1

트리거1

진료 테이블에서 입원이면 입원 테이블 자동 입력

진료 테이블

DS_NO	DS_DATE	DS_NOTE	DS_INOUT	DOC_NO	PA_NO
DS058	22/11/27	골절	입원	D021	P015

입원 테이블

HP_NO	IN_DATE	OUT_DATE	HP_DOCNO	NU_NO	HP_PANO	HR_NO
HP11	22/11/27	(null)	D021	N321	P015	101

```
CREATE OR REPLACE TRIGGER ds_insert_hp
AFTER INSERT on ds_t
FOR EACH ROW
DECLARE
BEGIN
IF :NEW.ds_inout = '입원' THEN
  INSERT INTO hp_t VALUES ('HP' || (SELECT COUNT(*) FROM hp_t), :NEW.ds_date, NULL, :NEW.doc_no,
    (SELECT nur_no FROM nur_t WHERE substr(:NEW.doc_no,2) = substr(nur_no,2) - 300), :NEW.pa_no, '101');
  DBMS_OUTPUT.PUT_LINE('입원 차트에 추가하였습니다. ');
END IF;
END;
/
```

프로젝트 수행 결과-트리거2

트리거2

예약 시 의사에 담당 환자가 제일 적은 의사 넣기

```
CREATE OR REPLACE TRIGGER res_view_tr
instead OF insert on res_view referencing old as old new as new
for each row
DECLARE
begin
INSERT INTO res_t values (:NEW.res_no, :NEW.res_date, :NEW.res_name, :NEW.res_call, :NEW.m_no, (select doc_no from(
select d.doc_no, ROW_NUMBER() OVER (ORDER BY (
select count(*) from pat_t p,doc_t d where p.doc_no = d.doc_no) desc) as rank_pa
from pat_t p,doc_t d where substr(d.m_no,2)= substr(:NEW.m_no,2) and rownum < 2 group by d.doc_no order by rank_pa)
where rank_pa <2));
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE(:NEW.res_name||'님');
DBMS_OUTPUT.PUT_LINE('예약하였습니다. ');
DBMS_OUTPUT.PUT_LINE('-----');
end;
/
```


프로젝트 수행 결과-트리거2

트리거2

예약 시 의사에 담당 환자가 제일 적은 의사 넣기

행 삽입

```
insert into res_view values ('RE021',sysdate,'이명재','01045679865','M001');
```

스크립트 출력 x | 질의 결과 x

작업이 완료되었습니다.(0.039초)

1 행 이(가) 삽입되었습니다.

Select문 보기

```
select * from res_t;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 21(0.003초)

	RES_NO	RES_DATE	RES_NAME	RES_CALL	M_NO
1	RE021	22/11/23	이명재	01045679865	M001
2	RE001	21/01/22	홍성근	0109461457	M001
3	RE002	22/02/10	소동호	0109461457	M001

View 보기

```
create OR REPLACE view res_view  
as select * from res_t;  
select * from res_view;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 21(0.004초)

	RES_NO	RES_DATE	RES_NAME	RES_CALL	M_NO
1	RE021	22/11/23	이명재	01045679865	M001
2	RE001	21/01/22	홍성근	0109461457	M001
3	RE002	22/02/10	소동호	0109461457	M001

프로젝트 수행 결과-파일1 (프로시저)

파일1

환자 이름 입력 시 모든 진료 기록 txt 파일로 출력

관리자 계정

```
CREATE DIRECTORY project_dir AS 'D:\';
```

```
GRANT READ, WRITE ON DIRECTORY project_dir TO project;
```

사용자 계정

```
CREATE OR REPLACE VIEW chart_view as
```

```
select rownum 차트번호, ds.ds_no 진료번호, ds.doc_no 의사번호, dc.doc_name 의사이름,  
ds.pa_no 환자번호, pa.pat_name 환자이름, ds.ds_note 소견
```

```
from ds t ds, pat t pa, doc t dc where ds.doc_no = dc.doc_no and ds.pa_no = pa.pat_no;
```

```
CREATE OR REPLACE PROCEDURE ds_WFile_name(fname varchar2, pname varchar2) IS
```

```
v_output UTL_FILE.FILE_TYPE;
```

```
v_result VARCHAR2(4000);
```

```
CURSOR sql_cur IS
```

```
SELECT 진료번호, 의사이름, 환자이름, 소견
```

```
FROM chart_view
```

```
where 환자이름 = pname;
```

```
BEGIN
```

```
v_output := UTL_FILE.FOPEN('PROJECT_DIR', fname, 'w', 32767);
```

```
FOR v_cur IN sql_cur LOOP
```

```
v_result := v_cur.진료번호||' '||v_cur.의사이름||' '||v_cur.환자이름||' '||v_cur.소견;
```

```
DBMS_OUTPUT.PUT_LINE(v_cur.진료번호||' '||v_cur.의사이름||' '||v_cur.환자이름||' '||v_cur.소견);
```

```
UTL_FILE.PUT_LINE(v_output, v_result);
```

```
END LOOP;
```

```
UTL_FILE.FCLOSE(v_output);
```

```
EXCEPTION
```

```
WHEN UTL_FILE.INVALID_PATH THEN
```

```
DBMS_OUTPUT.PUT_LINE('잘못된 경로');
```

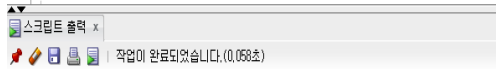
```
WHEN UTL_FILE.INVALID_OPERATION THEN
```

```
DBMS_OUTPUT.PUT_LINE('잘못된 쿼리');
```

```
END;
```

```
exec ds_WFile_NAME('chart.txt', '성진호');
```

결과



내 PC > 로컬 디스크 (D:)

이름

chart.txt

DS001	채향석	성진호	안구건조증
DS002	최규환	성진호	안구건조증
DS003	박정아	성진호	녹내장수술
DS041	채향석	성진호	시력검사



프로젝트 수행 결과-파일2 (프로시저)

파일2

모든 환자들의 진료 기록을 txt파일로 출력

관리자 계정

```
CREATE DIRECTORY project_dir AS 'D:\';
```

```
GRANT READ, WRITE ON DIRECTORY project_dir TO project;
```

사용자 계정

```
CREATE OR REPLACE VIEW chart_view as
```

```
select rownum 차트번호, ds.ds_no 진료번호, ds.doc_no 의사번호, dc.doc_name 의사이름,  
ds.pa_no 환자번호, pa.pat_name 환자이름, ds.ds_note 소견
```

```
from ds t ds, pat t pa, doc t dc where ds.doc no = dc.doc no and ds.pa no = pa.pat no;  
CREATE OR REPLACE PROCEDURE ds_WFile(fname varchar2) IS
```

```
v_output UTL_FILE.FILE_TYPE;
```

```
v_result VARCHAR2(4000);
```

```
CURSOR sql_cur IS
```

```
SELECT 진료번호, 의사이름, 환자이름, 소견
```

```
FROM chart_view;
```

```
BEGIN
```

```
v_output := UTL_FILE.FOPEN('PROJECT_DIR', fname, 'w', 32767);
```

```
FOR v_cur IN sql_cur LOOP
```

```
v_result := v_cur.진료번호||' | '||v_cur.의사이름||' | '||v_cur.환자이름||' | '||v_cur.소견;
```

```
DBMS_OUTPUT.PUT_LINE(v_cur.진료번호||' | '||v_cur.의사이름||' | '||v_cur.환자이름||' | '||v_cur.소견);
```

```
UTL_FILE.PUT_LINE(v_output, v_result);
```

```
END LOOP;
```

```
UTL_FILE.FCLOSE(v_output);
```

```
EXCEPTION
```

```
WHEN UTL_FILE.INVALID_PATH THEN
```

```
DBMS_OUTPUT.PUT_LINE('잘못된 경로');
```

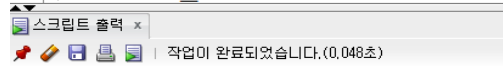
```
WHEN UTL_FILE.INVALID_OPERATION THEN
```

```
DBMS_OUTPUT.PUT_LINE('잘못된 쿼리');
```

```
END;
```

```
exec ds_WFile('Chart.txt');
```

결과

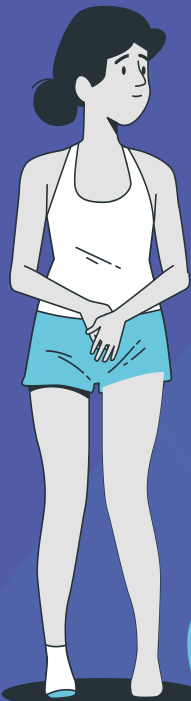


```
DS057 도진국 이윤모 골절  
DS001 채향석 성진호 안구건조증  
DS002 최규환 성진호 안구건조증  
DS003 박정아 성진호 녹내장수술  
DS004 박용호 설시윤 사랑니발치
```

내 PC > 로컬 디스크 (D:)

이름

chart.txt



04

자체 평가 의견

자체 평가 의견

자체 평가 의견

지영훈

이번 프로젝트를 진행하면서 지금까지 배웠던 내용을 다시 한번 확인해 보고 어떤 부분이 부족한지 확인해 볼 수 있었다.

윤종현

수업에서 배운 SQL 쿼리를 사용해 보는 좋은 기회를 가졌고 SQL, 쿼리를 찾고 문제를 해결하는 과정에서 다른 것들을 배운 좋은 경험이었습니다.

김성혜

SQL을 실습해보며 재밌었습니다. 잘하는 팀원들을 보며 많이 배운 시간이었습니다.

한정은

프로젝트를 진행하면서 부족했던 트리거, 프로시저, 함수 문법 등을 공부할 수 있었고 데이터 베이스를 처음부터 설계하면서 여러 가지 오류를 경험할 수 있었다.