

## 병원 업무 관리 ERD

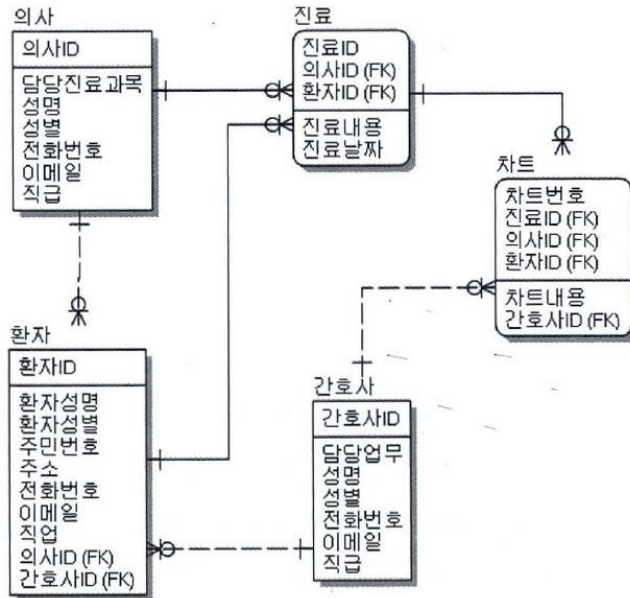
마당병원은 여러 명의 의사와 간호사가 환자를 관리한다. 병원은 환자를 체계적으로 관리하기 위하여 데이터베이스를 구축하였다. 의사, 환자, 간호사는 원하는 정보를 쉽게 검색, 관리할 수 있다.

**문제 1** 다음 요구사항을 읽고 개체, 속성, 관계를 식별한 후 <병원 업무 관리> ER 다이어그램을 작성하시오.

### ■ 요구사항

- ① 의사 한 명은 여러 명의 환자를 맡고, 환자 한 명은 한 명의 의사에게 진료를 받는다. 또한, 의사는 환자에 대한 모두 정보를 검색할 수 있다.
- ② 환자는 진료내용 등의 정보를 검색할 수 있으며 담당 의사에 대해서도 조회할 수 있다.
- ③ 간호사는 병원 진료 차트 정보를 검색할 수 있으며 의사와 더불어 환자를 관리한다. 간호사 업무는 여러 환자들의 진료접수, 환자관리, 차트관리 등을 수행한다.
- ④ 모든 의사는 의사ID를 부여하여 식별하며 담당진료과목, 성명, 성별, 전화번호, 이메일, 직급을 가진다.
- ⑤ 모든 환자는 환자ID를 부여하여 식별하며 담당의사, 담당간호사, 환자성명, 주민번호, 성별, 주소, 전화번호, 이메일, 직업을 가진다.
- ⑥ 모든 간호사는 간호사ID를 부여하여 식별하며 담당업무, 성명, 성별, 전화번호, 이메일, 직급을 가진다.
- ⑦ 진료ID는 날짜+진료순번으로 부여하고 진료한 의사ID, 환자ID, 진료내용, 진료날짜를 포함한다.
- ⑧ 진료를 받은 환자에 대한 내용은 차트에 기록하고, 차트는 차트번호를 부여하여 식별하며 담당 의사ID, 간호사ID, 환자ID와 진료ID 그리고, 의사 소견을 차트내용에 기록한다.

## 과제 1 : ER 다이어그램 구축 및 계정(schema) 생성



```

SQL> connect / as sysdba
SQL> CREATE TABLESPACE HOSTP_TBS
      2 DATAFILE ' 경로/파일명.dbf ' SIZE 20M;
SQL> CREATE USER U_HOSPITAL IDENTIFIED BY password;
SQL> GRANT CONNECT, RESOURCE, CREATE VIEW TO U_HOSPITAL;
SQL> CONNECT U_HOSPITAL/password

```

## 과제 2 : 테이블 구축

과제 1에서 구성한 ERD를 근거로 각각의 테이블 구조 및 관계, 제약조건 등을 설정하면 다음과 같다.

### ① 테이블명 : Doctors(의사)

테이블 구조, 관계, 제약조건

개체	Doctors(의사)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	의사ID	doc_id	number	10	Not Null / Unique
외래키					
속성	담당진료과목	major_treat	varcahr2	25	Not Null
	성명	doc_name	varchar2	20	Not Null
	성별	doc_gen	char	1	Not Null
	전화번호	doc_phone	varchar2	15	
	이메일	doc_email	varchar2	50	Unique
	직급	doc_position	varchar2	20	Not Null



## 테이블 생성 구문

```
CREATE TABLE Doctors
(
    doc_id          NUMBER(10) NOT NULL,
    major_treat     VARCHAR2(25) NOT NULL,
    doc_name        VARCHAR2(20) NOT NULL,
    doc_gen         CHAR(1) NOT NULL,
    doc_phone       VARCHAR2(15) NULL,
    doc_email       VARCHAR2(50) UNIQUE,
    doc_position    VARCHAR2(20) NOT NULL
);

ALTER TABLE Doctors
    ADD CONSTRAINT doc_id_pk PRIMARY KEY (doc_id);
```

## ② 테이블명 : Nurses(간호사)

### 테이블 구조, 관계, 제약조건

개체	Nurses(간호사)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	간호사ID	nur_id	number	10	Not Null / Unique
외래키					
속성	담당진료과목	major_job	varcahr2	25	Not Null
	성명	nur_name	varchar2	20	Not Null
	성별	nur_gen	char	1	Not Null
	전화번호	nur_phone	varchar2	15	
	이메일	nur_email	varchar2	50	Unique
	직급	nur_position	varchar2	20	Not Null

## 테이블 생성 구문

```
CREATE TABLE Nurses
(
    nur_id          NUMBER(10) NOT NULL,
    major_job       VARCHAR2(25) NOT NULL,
    nur_name        VARCHAR2(20) NOT NULL,
    nur_gen         CHAR(1) NOT NULL,
    nur_phone       VARCHAR2(15) NULL,
    nur_email       VARCHAR2(50) UNIQUE,
    nur_position    VARCHAR2(20) NOT NULL
);

ALTER TABLE Nurses
ADD CONSTRAINT nur_id_pk PRIMARY KEY (nur_id);
```

## ③ 테이블명 : Patients(환자)

### 테이블 구조, 관계, 제약조건

개체	Patients(환자)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	환자ID	pat_id	number	10	Not Null / Unique
외래키	간호사ID	nur_id	number	10	Not Null
	의사ID	doc_id	number	10	Not Null
속성	환자성명	pat_name	varcahr2	20	Not Null
	환자성별	pat_gen	char	1	Not Null
	주민번호	pat_jumin	varcahr2	14	Not Null
	주소	pat_addr	varchar2	100	Not Null
	전화번호	pat_phone	varchar2	15	
	이메일	pat_email	varchar2	50	Unique
	직업	pat_job	varcahr2	20	Not Null



## 테이블 생성 구문

```
CREATE TABLE Patients
(
    pat_id          NUMBER(10) NOT NULL,
    nur_id          NUMBER(10) NOT NULL,
    doc_id          NUMBER(10) NOT NULL,
    pat_name        VARCHAR2(20) NOT NULL,
    pat_gen         CHAR(1) NOT NULL,
    pat_jumin       VARCHAR2(14) NOT NULL,
    pat_addr        VARCHAR2(100) NOT NULL,
    pat_phone       VARCHAR2(15) NULL,
    pat_email       VARCHAR2(50) UNIQUE,
    pat_job         VARCHAR2(20) NOT NULL
);

ALTER TABLE Patients
    ADD CONSTRAINT pat_id_pk PRIMARY KEY (pat_id);

ALTER TABLE Patients
    ADD (CONSTRAINT R_2 FOREIGN KEY (doc_id) REFERENCES Doctors (doc_id));

ALTER TABLE Patients
    ADD (CONSTRAINT R_3 FOREIGN KEY (nur_id) REFERENCES Nurses (nur_id));
```

## ④ 테이블명 : Treatments(진료)

테이블 구조, 관계, 제약조건

개체	Treatments(진료)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	진료ID	treat_id	number	15	Not Null / Unique
	환자ID	pat_id	number	10	Not Null / Unique
	의사ID	doc_id	number	10	Not Null / Unique
외래키	환자ID	pat_id	number	10	
	의사ID	doc_id	number	10	
속성	진료내용	treat_contents	varcahr2	1000	Not Null
	진료날짜	treat_date	date		Not Null

## 테이블 생성 구문

```
CREATE TABLE Treatments
(
    treat_id      NUMBER(15) NOT NULL,
    pat_id        NUMBER(10) NOT NULL,
    doc_id        NUMBER(10) NOT NULL,
    treat_contents VARCHAR2(1000) NOT NULL,
    treat_date    DATE NOT NULL
);

ALTER TABLE Treatments
    ADD CONSTRAINT treat_pat_doc_id_pk PRIMARY KEY (treat_id, pat_id, doc_id);

ALTER TABLE Treatments
    ADD (CONSTRAINT R_5 FOREIGN KEY (pat_id) REFERENCES Patients (pat_id));

ALTER TABLE Treatments
    ADD (CONSTRAINT R_6 FOREIGN KEY (doc_id) REFERENCES Doctors (doc_id));
```

## ⑤ 테이블명 : Charts(차트)

### 테이블 구조, 관계, 제약조건

개체	Charts(차트)				
구분	Logical	Physical			
		Column	Data type	Size	Null / Unique
기본키	차트번호	chart_id	varcahr2	20	Not Null / Unique
	진료ID	treat_id	number	15	Not Null / Unique
	의사ID	doc_id	number	10	Not Null / Unique
	환자ID	pat_id	number	10	Not Null / Unique
외래키	진료ID	treat_id	number	10	
	의사ID	doc_id	number	10	
	환자ID	pat_id	number	10	
	간호사ID	nur_id	number	10	Not Null
속성	차트내용	chart_contents	varcahr2	1000	Not Null



## 테이블 생성 구문

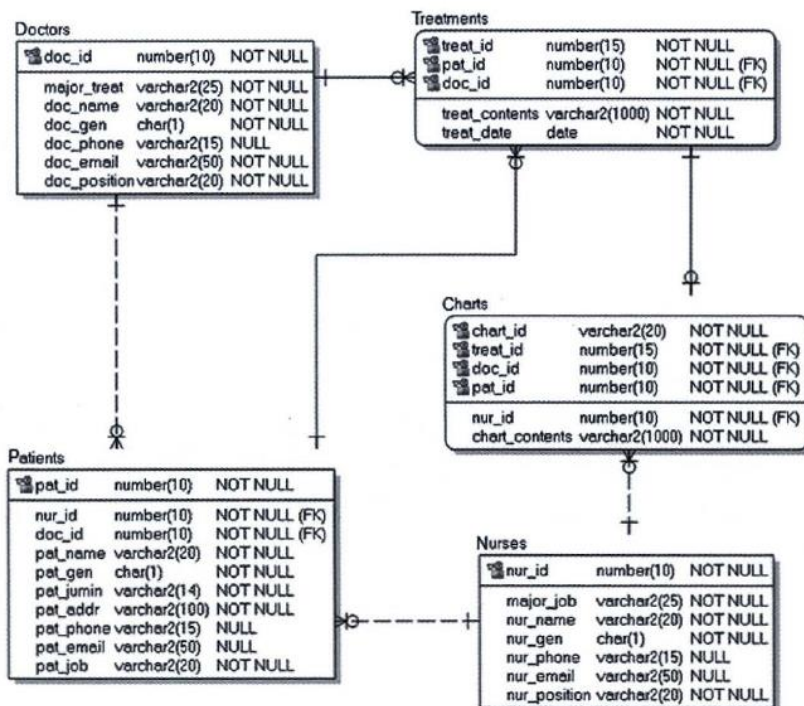
```
CREATE TABLE Charts
(
    chart_id          VARCHAR2(20) NOT NULL,
    treat_id          NUMBER(15) NOT NULL,
    doc_id            NUMBER(10) NOT NULL,
    pat_id            NUMBER(10) NOT NULL,
    nur_id            NUMBER(10) NOT NULL,
    chart_contents    VARCHAR2(1000) NOT NULL
);

ALTER TABLE Charts
    ADD CONSTRAINT chart_treat_doc_pat_id_pk PRIMARY KEY (chart_id, treat_id, doc_id, pat_id);

ALTER TABLE Charts
    ADD (CONSTRAINT R_4 FOREIGN KEY (nur_id) REFERENCES Nurses (nur_id));

ALTER TABLE Charts
    ADD (CONSTRAINT R_7 FOREIGN KEY (treat_id, pat_id, doc_id) REFERENCES Treatments (treat_id, pat_id, doc_id));
```

앞의 표를 확인하고 실제 데이터베이스에 접속하여 각각의 테이블을 해당 계정(schema)에 생성한다.





### 과제 3 : 데이터 입력(INSERT)

생성된 다섯 개의 테이블에 다음과 같은 정보를 입력한다. 각 테이블 컬럼의 데이터 타입 및 사이즈를 확인한 후 정확히 입력하도록 한다.

예) 의사 테이블에 데이터를 입력하는 경우

```
INSERT INTO Doctors VALUES (070576, '피부과', '홍길동', 'F', '016-333-7263', 'lja@hanbh.com', '전문의');
SELECT *
FROM Doctors;
```

DOC_ID	MAJOR_TREAT	DOC_NAME	DOC_GEN	DOC_PHONE	DOC_EMAIL	DOC_POSITION
70576	피부과	홍길동	F	016-333-7263	lja@hanbh.com	전문의

#### ① 테이블명 : Doctors(의사)

순서	doc_id	major_treat	doc_name	doc_gent	doc_phone	doc_email	doc_position
1	980312	소아과	이태정	M	010-333-1340	ltj@hanbh.com	과장
2	000601	내과	안성기	M	011-222-0987	ask@hanbh.com	과장
3	001208	외과	김민종	M	010-333-8743	kmj@han.com	과장
4	020403	피부과	이태서	M	019-777-3764	lts@hanbh.com	과장
5	050900	소아과	김연아	F	010-555-3746	kya@habhn.com	전문의
6	050101	내과	차태현	M	011-222-7643	cth@hanbh.com	전문의
7	062019	소아과	전지현	F	010-999-1265	jjh@hanbh.com	전문의
8	070576	피부과	홍길동	M	016-333-7263	hgd@hanbh.com	전문의
9	080543	방사선과	유재석	M	010-222-1263	yjs@hanbh.com	과장
10	091001	외과	김병만	M	010-555-3542	kbm@hanbh.com	전문의

#### ② 테이블명 : Nurse(간호사)

순서	nur_id	major_job	nur_name	nur_gen	nur_phone	nur_email	nur_position
1	050302	소아과	김은영	F	010-555-8751	key@hanbh.com	수간호사
2	050021	내과	윤성애	F	016-333-8745	ysa@hanbh.com	수간호사
3	040089	피부과	신지원	M	010-666-7646	sjw@hanbh.com	주임
4	070605	방사선과	유정화	F	010-333-4588	yjh@hanbh.com	주임

5	070804	내과	라하나	F	010-222-1340	nhn@hanbh.com	주임
6	071018	소아과	김화경	F	019-888-4116	khk@hanbh.com	주임
7	100356	소아과	이선용	M	010-777-1234	lsy@hanbh.com	간호사
8	104145	외과	김현	M	010-999-8520	kh@hanbh.com	간호사
9	120309	피부과	박성완	M	010-777-4996	psw@hanbh.com	간호사
10	130211	외과	이서연	F	010-222-3214	lsy2@hanbh.com	간호사

### ③ 테이블명 : Patients(환자)

순서	pat_id	nur_id	doc_id	pat_name	pat_gen	pat_jumin	pat_addr	pat_phone	pat_email	pat_job
1	2345	050302	980312	안상건	M	232345	서울	010-555-7845	ask@ab.com	회사원
2	3545	040089	020403	김성룡	M	543545	서울	010-333-7812	ksr@bb.com	자영업
3	3424	070605	080543	이종진	M	433424	부산	019-888-4859	lj@ab.com	회사원
4	7675	100356	050900	최광석	M	677675	당진	010-222-4847	cks@cc.com	회사원
5	4533	070804	000601	정한경	M	744533	강릉	010-777-9630	jhk@ab.com	교수
6	5546	120309	070576	유원현	M	765546	대구	016-777-0214	ywh@cc.com	자영업
7	4543	070804	050101	최재정	M	454543	부산	010-555-4187	cjj@bb.com	회사원
8	9768	130211	091001	이진희	F	119768	서울	010-888-3675	ljh@ab.com	교수
9	4234	130211	091001	오나미	F	234234	속초	010-999-6541	onm@cc.com	학생
10	7643	071018	062019	송성묵	M	987643	서울	010-222-5874	ssm@bb.com	학생

### ④ 테이블명 : Treatments(진료)

순서	treat_id	pat_id	doc_id	treat_contents	treat_date
1	130516023	2345	980312	감기, 몸살	2013-05-16
2	130628100	3545	020403	피부 트러블 치료	2013-06-28
3	131205056	3424	080543	목 디스크로 MRI 촬영	2013-12-05
4	131218024	7675	050900	종이염	2013-12-18
5	131224012	4533	000601	장염	2013-12-24
6	140103001	5546	070576	여드름 치료	2014-01-03



7	140109026	4543	050101	위염	2014-01-09
8	140226102	9768	091001	화상치료	2014-02-26
9	140303003	4234	091001	교통사고 외상치료	2014-03-03
10	140308087	7643	062019	장염	2014-03-08

#### ⑤ 테이블명 : Charts(차트)

순서	chart_id	treat_id	doc_id	pat_id	nur_id	chart_contents
1	p_130516023	130516023	980312	2345	050302	감기 주사 및 약 처방
2	d_130628100	130628100	020403	3545	040089	피부 감염 방지 주사
3	r_131205056	131205056	080543	3424	070605	주사 처방
4	p_131218024	131218024	050900	7675	100356	귓속청소 및 약 처방
5	l_131224012	131224012	000601	4533	070804	장염 입원치료
6	d_140103001	140103001	070576	5546	120309	여드름 치료약 처방
7	i_140109026	140109026	050101	4543	070804	위내시경
8	s_140226102	140226102	091001	9768	130211	화상 크림약 처방
9	s_140303003	140303003	091001	4234	130211	입원치료
10	p_140308087	140308087	062019	7643	071018	장염 입원치료

#### 과제 4 : 데이터 변경(UPDATE) 및 삭제(DELETE)

**문제 1** 홍길동 의사가 맡고 있던 담당진료과목이 피부과에서 소아과로 변경되어 내일부터 진료를 시작할 예정이다. 이 정보에 대한 테이블 정보를 변경하시오.

```

UPDATE  Doctors
SET      major_treat = '소아과'
WHERE    doc_name = '홍길동';

```

**문제 2** 김은영 간호사는 대학원 진학으로 오늘까지만 근무하고 퇴사하게 되었다. 이 정보에 대한 테이블 정보를 변경 하시오.

```
DELETE FROM Nurses
WHERE nur_name = '김은영';
```

## 과제 5 : 데이터 검색(QUERY)

**문제 1** 담당진료과목이 '소아과'인 의사에 대한 정보를 출력하시오.

```
SELECT *
FROM Doctors
WHERE major_treat = '소아과';
```

**문제 2** 홍길동 의사에게 진료를 받은 환자에 대한 모든 정보를 출력하시오.

```
SELECT p.*, d.doc_name
FROM Doctors d JOIN Patients p ON (d.doc_id = p.doc_id)
WHERE d.doc_name = '홍길동';
```

**문제 3** 진료날짜가 2013년 12월인 환자에 대한 모든 정보를 오름차순 정렬하여 출력하시오.

```
SELECT p.*, t.treat_date
FROM Treatments t JOIN Patients p ON (t.pat_id = p.pat_id)
WHERE t.treat_date BETWEEN '2013-12-01' AND '2013-12-31'
ORDER BY t.treat_date ASC;
```

**문제 4** 간호사ID가 05로 시작하는 모든 간호사 정보를 출력하시오.

```
SELECT *
FROM Nurses
WHERE nur_id LIKE '05%';
```