Team Members: Abel Lagonell, Dallas DeSimones, Gregory Jans, Jaleel Rogers

Professor Elish

COP3730: Database 1

Hospital Database

Abel Lagonell
Dallas DeSimones
Gregory Jans
Jaleel Rogers

Organization Overview:

The Hospital is made up of many buildings. The Hospital in question is made up of administrators that take care of the Teams, Departments, and Organization of the whole Hospital. The different Doctors and Nurses are allowed partial access to the Database to input the different medications and procedures that a patient might need or schedule. Managers would also have access to their respective employees to put in what type of job they have. The Team leads under the managers can edit and control the teams themselves. Teams are also able to be set to different procedures by the department heads such that two different teams of surgeons don't have the same procedure to do at the same time.

Business Rules:

A **department** is located in several **buildings**
Each **building** houses one **department**
A **department** oversees several **teams**
Each **team** is managed by one **department**
A **team** operates several **buildings**
Each **building** is operated by only one **team**
A **building** has many **rooms**
Each **room** is located in only one **building**
A **room** can have many **appointments** inside
A **room** does not need to have **appointments** scheduled in it
Each **appointment** is held in one room
An **appointment** may have a **procedure** take place during it
Each **procedure** happens during one **appointment**
A **procedure** is performed by one **team**
A **team** may conduct several **procedures**
A **team** is not required to conduct any **procedures**
A **patient** can have many **procedures** performed on them
A **patient** is not required to have **procedures** performed on them
Each **procedure** happens to only one **patient**
A **patient** can attend several **appointments**
A **patient** is not required to attend any **appointments**
Each **appointment** is only for one **patient**
A **patient** may be given many **prescriptions**
A **patient** is not required to receive a **prescription**
Each **prescription** is given to only one **patient**
Each **prescription** contains only one **medication**
A **medication** can be prescribed in many **prescriptions**

Abel Lagonell
Dallas DeSimones
Gregory Jans
Jaleel Rogers

A **medication** does not need to be prescribed in any **prescriptions**

An **employee** may lead one **team**

A **team** is led by only one **employee**

An **employee** is assigned to one **team**

A **team** has many **employees**

An **employee** may chair a **department**

A **department** must be chaired by one **employee**

An **employee** has one **job**

A **job** may be assigned to many **employees**

A **job** does not need to be assigned to any **employees**

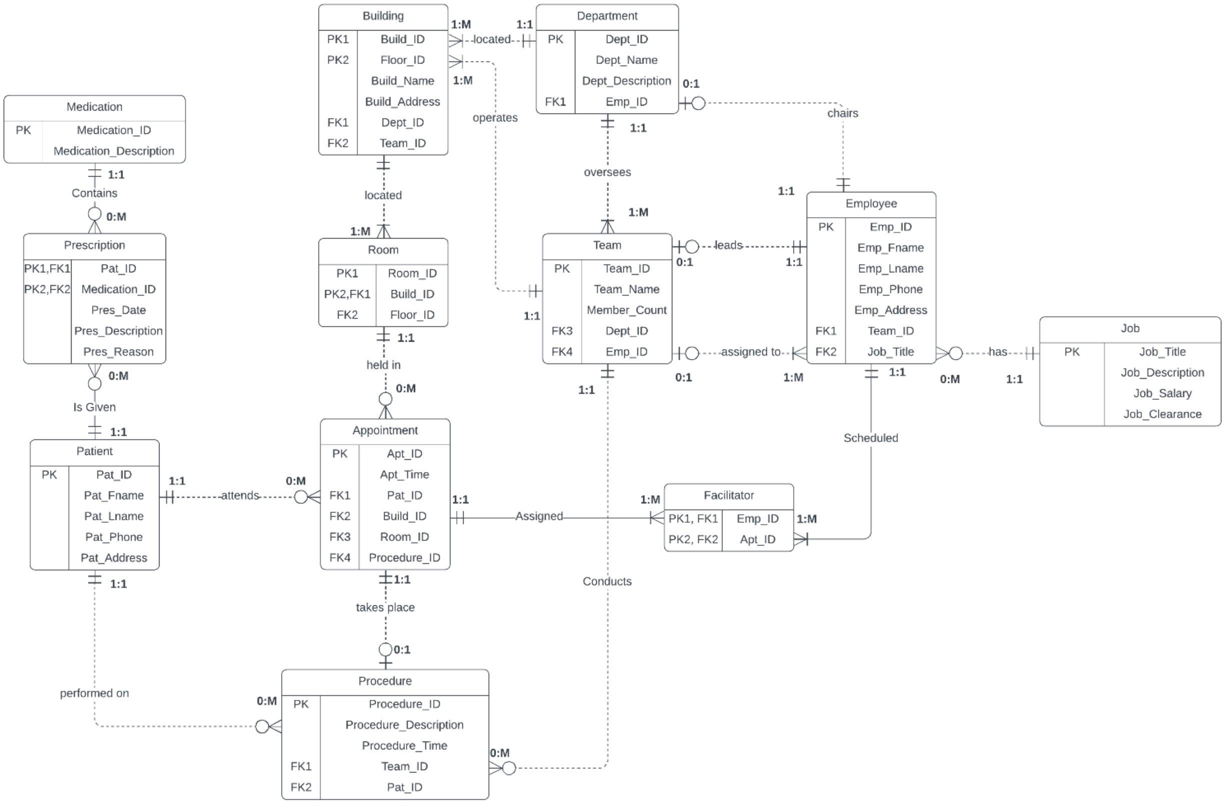
An **employee** is scheduled to be a **facilitator** many times

Each **facilitator** is one **employee**

A **facilitator** is assigned to one **appointment**

An **appointment** can be assigned to many **facilitators**

Entity Relationship Model



The whole ERD is in 3NF, because there are no transient dependencies or partial dependencies in any of the entities in the diagram.

Abel Lagonell
Dallas DeSimone
Gregory Jans
Jaleel Rogers

DDL Statements:

```
create table MEDICATION (
MEDICATION_ID                    number                          constraint medication_pk
primary key,
MEDICATION_DESCRIPTION   varchar2(35) not null
);

create table PATIENT (
PAT_ID                   number                          constraint patient_pk primary key,
PAT_FNAME      varchar2(20),
PAT_LNAME      varchar2(20),
PAT_PHONE      varchar2(12),
PAT_ADDRESS varchar2(35)
);

create table PRESCRIPTION (
PAT_ID                           number                          references PATIENT (PAT_ID),
MEDICATION_ID            number                                  references MEDICATION
(MEDICATION_ID),
PRES_DATE                date not null,
PRES_DESCRIPTION  varchar2(35) not null,
PRES_REASON              varchar2(35) not null,
constraint prescription_pk primary key (PAT_ID, MEDICATION_ID)
);

create table JOB (
JOB_TITLE            varchar2(35)            constraint job_pk primary key,
JOB_DESCRIPTION varchar2(35),
JOB_SALARY           number,
JOB_CLEARANCE        number not null
);

create table EMPLOYEE (
EMP_ID                   number                          constraint employee_pk primary key,
EMP_FNAME      varchar2(20),
EMP_LNAME      varchar2(20),
EMP_PHONE      varchar2(12),
EMP_ADDRESS varchar2(35),
TEAM_ID        number,
JOB_TITLE      varchar2(35)            references JOB (JOB_TITLE)
);

create table DEPARTMENT (
DEPT_ID                          number                          constraint department_pk
primary key,
DEPT_NAME                       varchar2(20),
DEPT_DESCRIPTION    varchar2(35),
EMP_ID                                  number                              references EMPLOYEE
(EMP_ID)
);

create table TEAM (
TEAM_ID                 number                                      constraint team_pk primary key,
TEAM_NAME               varchar2(20),
MEMBER_COUNT            number,
DEPT_ID                 number                          references DEPARTMENT (DEPT_ID) not null,
EMP_ID                          number                              references EMPLOYEE (EMP_ID)
);

alter table EMPLOYEE add constraint emp_team_fk foreign key (TEAM_ID) references TEAM
(TEAM_ID);

create table BUILDING (
BUILD_ID                number,
FLOOR_ID                number,
BUILD_NAME              varchar2(20),
BUILD_ADDRESS           varchar2(35) not null,
DEPT_ID references DEPARTMENT (DEPT_ID),
TEAM_ID references TEAM (TEAM_ID),
constraint building_pk primary key (BUILD_ID, FLOOR_ID)
);

create table ROOM (
ROOM_ID         number,
BUILD_ID        number,
FLOOR_ID        number not null,
constraint room_build_fk foreign key (BUILD_ID, FLOOR_ID) references BUILDING (BUILD_ID,
FLOOR_ID),
constraint room_pk      primary key (ROOM_ID, BUILD_ID)
);

create table PROCEDURE (
PROCEDURE_ID                            number                              constraint procedure_pk primary
key,
PROCEDURE_DESCRIPTION   varchar2(35)    not null,
PROCEDURE_TIME                          timestamp               not null,
TEAM_ID                                 number                              references TEAM (TEAM_ID),
PAT_ID                                          number                      references PATIENT (PAT_ID)
);

create table APPOINTMENT (
APT_ID                          number                      constraint appointment_pk primary key,
APT_TIME                timestamp       not null,
PAT_ID                          number          references PATIENT (PAT_ID),
ROOM_ID                 number,
BUILD_ID                number,
PROCEDURE_ID            number          references PROCEDURE (PROCEDURE_ID),
constraint apt_room_fk foreign key (ROOM_ID, BUILD_ID) references ROOM (ROOM_ID, BUILD_ID)
);

create table FACILITATOR (
EMP_ID          number references EMPLOYEE (EMP_ID),
APT_ID          number references APPOINTMENT (APT_ID),
constraint facilitator_pk primary key (EMP_ID, APT_ID)
);
```

Data Dictionary:

MEDICATION

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
MEDICATION_ID	Medication ID	INTEGER	#####	NA	Y	PK	
MEDICATION_DESCRIPTION	Medication Description	VARCHAR(35)	Xxxxxxxxx	NA	Y		

PRESCRIPTION

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
PAT_ID	Patient ID	INTEGER	#####	NA	Y	PK & FK	PATIENT
MEDICATION_ID	Medication ID	INTEGER	#####	NA	Y	PK & FK	MEDICATION
PRES_DATE	Prescription Date	DATE	DD-MON-YYYY	NA	Y		
PRES_DESCRIPTION	Prescription Description	VARCHAR(35)	Xxxxxxxxx	NA	Y		
PRES_REASON	Prescription Reason	VARCHAR(35)	Xxxxxxxxx	NA	Y		

PATIENT

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
PAT_ID	Patient ID	INTEGER	#####	NA	Y	PK	PRESCRIPTIO N
PAT_FNAME	Patient First Name	VARCHA R(20)	Xxxxxxxx x	NA	N		
PAT_LNAME	Patient Last Name	VARCHA R(20)	Xxxxxxxx x	NA	N		
PAT_PHONE	Patient Phone Number	CHAR	999-999-9999	NA	N		
PAT_ADDRESS	Patient Address	VARCHA R(35)	Xxxxxxxx x	NA	N		

BUILDING

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
BUILD_ID	Building ID	INTEGER	#####	NA	Y	PK	
FLOOR_ID	Floor ID	INTEGER	#####	NA	Y	PK	
BUILD_NAME	Building Name	VARCHA R(20)	Xxxxxxxx x	NA	N		
BUILD_ADDRESS	Building Address	VARCHA R(35)	Xxxxxxxx x	NA	Y		
DEPT_ID	Department ID	INTEGER	#####	NA	N	FK	DEPARTMENT
TEAM_ID	Team ID	INTEGER	#####	NA	N	FK	TEAM

ROOM

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
ROOM_ID	Room ID	INTEGER	#####	NA	Y	PK	
BUILD_ID	Building ID	INTEGER	#####	NA	Y	PK & FK	BUILDING
FLOOR_ID	Floor ID	INTEGER	#####	NA	Y	FK	BUILDING

APPOINTMENT

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
APT_ID	Appointment ID	INTEGER	#####	NA	Y	PK	
APT_TIME	Appointment Time	TIMESTAMP(0)	DD-MMM-YYYY HH:MM:SS	NA	Y		
PAT_ID	Patient ID	INTEGER	#####	NA	N	FK	PATIENT
BUILD_ID	Building ID	INTEGER	#####	NA	N	FK	BUILDING
ROOM_ID	Room ID	INTEGER	#####	NA	N	FK	ROOM
PROCEDURE_ID	Procedure ID	INTEGER	#####	NA	Y	FK	PROCEDURE

PROCEDURE

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
PROCEDURE_ID	Procedure ID	INTEGER	#####	NA	Y	PK	
PROCEDURE_DESCRIPTION	Procedure Description	VARCHAR(35)	Xxxxxxxxx	NA	Y		
PROCEDURE_TIME	Procedure Time	TIMESTAMP(0)	DD-MMM-YYYY HH:MM:SS	NA	Y		
TEAM_ID	Team ID	INTEGER	#####	NA	N	FK	TEAM
PAT_ID	Patient ID	INTEGER	#####	NA	N	FK	PATIENT

DEPARTMENT

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
DEPT_ID	Department ID	INTEGER	#####	NA	Y	PK	
DEPT_NAME	Department Name	VARCHAR(20)	Xxxxxxxxx	NA	N		

DEPT_DESCRIPTION	Department Description	VARCHAR(35)	Xxxxxxxxx	NA	N		
EMP_ID	Employee ID	INTEGER	#####	NA	N	FK	EMPLOYEE

TEAM

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
TEAM_ID	Team ID	INTEGER	#####	NA	Y	PK	
TEAM_NAME	Team Name	VARCHAR(20)	Xxxxxxxxx	NA	N		
MEMBER_COUNT	Member Count	INTEGER	#####	NA	N		
DEPT_ID	Department ID	INTEGER	#####	NA	Y	FK	DEPARTMENT
EMP_ID	Employee ID	INTEGER	#####	NA	N	FK	EMPLOYEE

FACILITATOR

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
EMP_ID	Employee ID	INTEGER	#####	NA	Y	PK & FK	EMPLOYEE
APT_ID	Appointment ID	INTEGER	#####	NA	Y	PK & FK	APPOINTMENT

EMPLOYEE

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
EMP_ID	Employee ID	INTEGER	#####	NA	Y	PK	
EMP_FNAME	Employee First Name	VARCHAR (20)	Xxxxxxxxx	NA	N		
EMP_LNAME	Employee Last Name	VARCHAR (20)	Xxxxxxxxx	NA	N		
EMP_PHONE	Employee Phone Number	CHAR	999-999-9999	NA	N		
EMP_ADDRESS	Employee Address	VARCHAR (35)	Xxxxxxxxx	NA	N		

TEAM_ID	Team ID	INTEGER	#####	NA	N	FK	TEAM
JOB_TITLE	Job Title	VARCHAR (35)	Xxxxxxxxx	NA	N	FK	JOB

JOB

ATTRIBUTE NAME	DESCRIPTION	TYPE	FORMAT	RANGE	REQUIRED	PK or FK	FK REFERENCED TABLE
JOB_TITLE	Job Title	VARCHAR (35)	Xxxxxxxxx	NA	Y	PK	
JOB_DESCRIPTION	Job Description	VARCHAR (35)	Xxxxxxxxx	NA	N		
JOB_SALARY	Job Salary	INTEGER	#####	NA	N		
JOB_CLEARANCE	Job Security Clearance	INTEGER(2)	##	0-9	Y		

```
-- 1) Scripts to populate data into all your tables

create table MEDICATION (
MEDICATION_ID                   number                  constraint medication_pk primary key,
MEDICATION_DESCRIPTION  varchar2(35) not null
);

create table PATIENT (
PAT_ID          number                          constraint patient_pk primary key,
PAT_FNAME       varchar2(20),
PAT_LNAME       varchar2(20),
PAT_PHONE       varchar2(12),
PAT_ADDRESS varchar2(35)
);

create table PRESCRIPTION (
PAT_ID                          number                          references PATIENT (PAT_ID),
MEDICATION_ID           number                          references MEDICATION (MEDICATION_ID),
PRES_DATE                       date not null,
PRES_DESCRIPTION        varchar2(35) not null,
PRES_REASON             varchar2(35) not null,
constraint prescription_pk primary key (PAT_ID, MEDICATION_ID)
);


create table JOB (
JOB_TITLE               varchar2(35)            constraint job_pk primary key,
JOB_DESCRIPTION varchar2(35),
JOB_SALARY              number,
JOB_CLEARANCE   number not null
);

create table EMPLOYEE (
EMP_ID          number                          constraint employee_pk primary key,
EMP_FNAME       varchar2(20),
EMP_LNAME       varchar2(20),
EMP_PHONE       varchar2(12),
EMP_ADDRESS varchar2(35),
TEAM_ID         number,
JOB_TITLE       varchar2(35)            references JOB (JOB_TITLE)
);

create table DEPARTMENT (
DEPT_ID                         number                          constraint department_pk primary key,
DEPT_NAME                       varchar2(20),
DEPT_DESCRIPTION        varchar2(35),
EMP_ID                          number                          references EMPLOYEE (EMP_ID)
);

create table TEAM (
TEAM_ID                 number                          constraint team_pk primary key,
TEAM_NAME               varchar2(20),
MEMBER_COUNT    number,
DEPT_ID              number                         references DEPARTMENT (DEPT_ID) not null,
EMP_ID               number                         references EMPLOYEE (EMP_ID)
);

alter table EMPLOYEE add constraint emp_team_fk foreign key (TEAM_ID) references TEAM (TEAM_ID);

create table BUILDING (
BUILD_ID             number,
FLOOR_ID             number,
BUILD_NAME              varchar2(20),
BUILD_ADDRESS   varchar2(35) not null,
DEPT_ID references DEPARTMENT (DEPT_ID),
TEAM_ID references TEAM (TEAM_ID),
constraint building_pk primary key (BUILD_ID, FLOOR_ID)
);

create table ROOM (
ROOM_ID        number,
BUILD_ID       number,
FLOOR_ID       number not null,
constraint room_build_fk foreign key (BUILD_ID, FLOOR_ID) references BUILDING (BUILD_ID, FLOOR_ID),
constraint room_pk      primary key (ROOM_ID, BUILD_ID)
);

create table PROCEDURE (
PROCEDURE_ID                number                constraint procedure_pk primary key,
PROCEDURE_DESCRIPTION   varchar2(35)    not null,
PROCEDURE_TIME                timestamp             not null,
TEAM_ID                             number                references TEAM (TEAM_ID),
PAT_ID                              number                references PATIENT (PAT_ID)
);

create table APPOINTMENT (
APT_ID               number         constraint appointment_pk primary key,
APT_TIME               timestamp       not null,
PAT_ID               number         references PATIENT (PAT_ID),
ROOM_ID                number,
BUILD_ID              number,
PROCEDURE_ID    number          references PROCEDURE (PROCEDURE_ID),
constraint apt_room_fk foreign key (ROOM_ID, BUILD_ID) references ROOM (ROOM_ID, BUILD_ID)
);

create table FACILITATOR (
EMP_ID  number  references EMPLOYEE (EMP_ID),
APT_ID  number  references APPOINTMENT (APT_ID),
constraint facilitator_pk primary key (EMP_ID, APT_ID)
);


insert into MEDICATION values (57001, 'Tylenol');
insert into MEDICATION values (57002, 'Advil');
insert into MEDICATION values (57003, 'Ibuprofen');
insert into MEDICATION values (57004, 'Aspirin');
insert into MEDICATION values (57005, 'Motrin');
insert into PATIENT values (40001, 'Deven', 'Smith', '123-456-7890', '753 Miami St');
insert into PATIENT values (40002, 'Zachary', 'Chair', '234-567-8901', '159 Lake Rd');
insert into PATIENT values (40003, 'Abigail', 'Stucki', '345-678-9012', '456 Orlando Blvd');
insert into PATIENT values (40004, 'Ethan', 'Jones', '456-789-0123', '852 Tampa Dr');
insert into PATIENT values (40005, 'Johnny', 'Bravo', '567-890-1234', '012 Jacksonville Ave');
insert into PATIENT values (40006, 'Avery', 'Wilson', '678-901-2345', '123 Main St');
insert into PATIENT values (40007, 'Hannah', 'Parker', '789-012-3456', '456 Elm Rd');
insert into PATIENT values (40008, 'Liam', 'Adams', '890-123-4567', '789 Maple Blvd');
insert into PATIENT values (40009, 'Emma', 'Cook', '901-234-5678', '012 Oak Dr');
insert into PATIENT values (40010, 'Noah', 'Baker', '012-345-6789', '345 Pine Ave');
insert into PATIENT values (40011, 'Sophia', 'Green', '123-123-1234', '678 Cedar Ln');
insert into PATIENT values (40012, 'William', 'Collins', '234-234-2345', '901 Birch St');
insert into PATIENT values (40013, 'Isabella', 'Taylor', '345-345-3456', '234 Spruce Rd');

insert into PRESCRIPTION values (40013, 57003, timestamp'2023-04-06 10:00:00', 'once per day', 'headache');
insert into PRESCRIPTION values (40011, 57004, timestamp'2023-02-13 1:35:00', 'twice per day', 'toothache');
insert into PRESCRIPTION values (40001, 57002, timestamp'2023-05-01 15:00:00', 'thrice per week', 'stomachache');
insert into PRESCRIPTION values (40004, 57003, timestamp'2023-01-01 00:00:00', 'once per week', 'backache');
insert into PRESCRIPTION values (40004, 57005, timestamp'2023-03-15 12:00:00', 'twice biweekly', 'fever');

insert into JOB values ('Doctor', 'Medical', 100000, 5);
insert into JOB values ('Nurse', 'Medical', 50000, 4);
insert into JOB values ('Janitor', 'Cleaning', 30000, 1);
insert into JOB values ('Security', 'Security', 40000, 3);
insert into JOB values ('Manager', 'Management', 80000, 5);

insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10001, ' John', 'Smith', '123-456-7890', '123 Main St', 'Nurse');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10002, ' Emma', 'Johnson', '234-567-8901', '456 Secondary Road', 'Doctor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10003, ' Michael', 'Williams', '345-678-9012', '789 Third Ave', 'Janitor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10004, ' Emily', 'Jones', '456-789-0123', '012 Fourth Dr', 'Janitor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10005, ' David', 'Brown', '567-890-1234', '345 Fifth Blvd', 'Security');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10006, ' Olivia', 'Davis', '678-901-2345', '678 Sixth St', 'Security');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10007, ' James', 'Miller', '789-012-3456', '901 Seventh Rd', 'Security');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10008, ' Isabella', 'Wilson', '890-123-4567', '234 Eighth Ave', 'Manager');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10009, ' Robert', 'Moore', '901-234-5678', '567 Ninth Dr', 'Manager');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10010, ' Sophia', 'Taylor', '012-345-6789', '890 Tenth Blvd', 'Security');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10011, ' Ava', 'Anderson', '234-567-8901', '456 Maple Street', 'Doctor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10012, ' Daniel', 'Thomas', '345-678-9012', '789 Oak Avenue', 'Manager');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10013, ' Mia', 'Jackson', '456-789-0123', '123 Pine Lane', 'Doctor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
```

```
values (10014, ' Joseph', 'Lee', '567-890-1234', '234 Elm Road', 'Doctor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10015, ' Grace', 'Garcia', '678-901-2345', '567 Cedar Court', 'Janitor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10016, ' William', 'Martin', '789-012-3456', '1212 First Street', 'Janitor');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10017, ' Charlotte', 'Hall', '890-123-4567', '2323 Second Avenue', 'Nurse');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10018, ' Ethan', 'Allen', '901-234-5678', '3434 Third Boulevard', 'Security');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10019, ' Abigail', 'Wright', '012-345-6789', '4545 Fourth Drive', 'Nurse');
insert into EMPLOYEE (EMP_ID, EMP_FNAME, EMP_LNAME, EMP_PHONE, EMP_ADDRESS, JOB_TITLE)
values (10020, ' Benjamin', 'Scott', '234-567-8901', '5656 Fifth Boulevard', 'Manager');

insert into DEPARTMENT values (11, 'Emergency', 'ER', 10001);
insert into DEPARTMENT values (22, 'Neurology', 'Brain', 10002);
insert into DEPARTMENT values (33, 'Cardiology', 'Heart', 10003);
insert into DEPARTMENT values (44, 'Surgery', 'Surgery', 10004);
insert into DEPARTMENT values (55, 'Pediatrics', 'Children', 10005);

insert into TEAM values (20001, 'Emergency 1', 5, 11, 10013);
insert into TEAM values (20002, 'Emergency 2', 7, 11, 10013);
insert into TEAM values (20003, 'Neurology', 3, 22, 10018);
insert into TEAM values (20004, 'Cardiology', 9, 33, 10010);
insert into TEAM values (20005, 'Surgery 1', 2, 44, 10018);
insert into TEAM values (20006, 'Surgery 2', 4, 44, 10010);
insert into TEAM values (20007, 'Pediatrics', 2, 55, 10003);

insert into BUILDING values (101, 1, 'R.C.H North', '123 RDJ Rd', 11, 20001);
insert into BUILDING values (101, 2, 'R.C.H North', '123 RDJ Rd', 11, 20002);
insert into BUILDING values (101, 3, 'R.C.H North', '123 RDJ Rd', 22, 20003);
insert into BUILDING values (101, 4, 'R.C.H North', '123 RDJ Rd', 33, 20004);
insert into BUILDING values (101, 5, 'R.C.H North', '123 RDJ Rd', 33, 20004);
insert into BUILDING values (102, 1, 'Baptiste Center', '10-24 CH Blvd', 44, 20005);
insert into BUILDING values (102, 2, 'Baptiste Center', '10-24 CH Blvd', 44, 20005);
insert into BUILDING values (102, 3, 'Baptiste Center', '10-24 CH Blvd', 44, 20006);
insert into BUILDING values (103, 1, 'Overwatch HQ', '705 That St', 55, 20007);

insert into ROOM values (1101, 101, 1);
insert into ROOM values (1102, 101, 1);
insert into ROOM values (1103, 101, 1);
insert into ROOM values (1201, 101, 2);
insert into ROOM values (1202, 101, 2);
insert into ROOM values (1203, 101, 2);
insert into ROOM values (1301, 101, 3);
insert into ROOM values (1302, 101, 3);
insert into ROOM values (1401, 101, 4);
insert into ROOM values (1501, 101, 5);
insert into ROOM values (1502, 101, 5);
insert into ROOM values (1503, 101, 5);
insert into ROOM values (2101, 102, 1);
insert into ROOM values (2102, 102, 1);
insert into ROOM values (2104, 102, 1);
insert into ROOM values (2201, 102, 1);
insert into ROOM values (2202, 102, 2);
insert into ROOM values (2203, 102, 2);
insert into ROOM values (2204, 102, 2);
insert into ROOM values (2205, 102, 2);
insert into ROOM values (2301, 102, 2);
insert into ROOM values (2302, 102, 3);
insert into ROOM values (2303, 102, 3);
insert into ROOM values (3101, 102, 3);
insert into ROOM values (3102, 103, 1);
insert into ROOM values (3103, 103, 1);
insert into ROOM values (3104, 103, 1);
insert into ROOM values (3105, 103, 1);

insert into PROCEDURE values (58401, 'Appendectomy', timestamp'2023-04-06 10:00:00', 20005, 40006);
insert into PROCEDURE values (58402, 'Blood transfusion', timestamp'2023-02-13 1:35:00', 20002, 40002);
insert into PROCEDURE values (58403, 'X-ray', timestamp'2023-05-01 15:00:00', 20007, 40008);
insert into PROCEDURE values (58404, 'Tonsillectomy', timestamp'2023-01-01 00:00:00', 20005, 40008);
insert into PROCEDURE values (58405, 'Vaccination', timestamp'2023-03-15 12:00:00', 20007, 40013);

insert into APPOINTMENT values (50001, timestamp'2023-04-06 10:00:00', 40012, 1502, 101, 58401);
insert into APPOINTMENT values (50002, timestamp'2023-02-13 1:35:00', 40013, 3104, 103, null );
insert into APPOINTMENT values (50003, timestamp'2023-05-01 15:00:00', 40002, 1301, 101, 58402);
insert into APPOINTMENT values (50004, timestamp'2023-01-01 00:00:00', 40009, 2203, 102, null );
insert into APPOINTMENT values (50005, timestamp'2023-03-15 12:00:00', 40011, 1103, 101, 58403);
insert into APPOINTMENT values (50006, timestamp'2023-06-22 11:30:00', 40010, 1101, 101, null );
insert into APPOINTMENT values (50007, timestamp'2023-07-09 14:15:00', 40010, 3102, 103, null );
insert into APPOINTMENT values (50008, timestamp'2023-08-16 09:45:00', 40006, 1301, 101, null );
insert into APPOINTMENT values (50009, timestamp'2023-09-03 16:20:00', 40012, 2204, 102, 58404);
insert into APPOINTMENT values (50010, timestamp'2023-10-11 13:00:00', 40008, 2201, 102, 58405);

insert into FACILITATOR values (10019, 50001);
insert into FACILITATOR values (10015, 50002);
insert into FACILITATOR values (10019, 50003);
insert into FACILITATOR values (10003, 50004);
insert into FACILITATOR values (10002, 50005);

select * from MEDICATION;
select * from PATIENT;
select * from PRESCRIPTION;
select * from JOB;
select * from EMPLOYEE;
select * from DEPARTMENT;
select * from TEAM;
select * from BUILDING;
select * from ROOM;
select * from PROCEDURE;
select * from APPOINTMENT;
select * from FACILITATOR;

-- 2) Scripts to update data (update (SQL DML) statements should be in various complexity, at least 2 per table)

update MEDICATION set MEDICATION_DESCRIPTION = 'Pepto' where MEDICATION_ID = 57002;
update MEDICATION set MEDICATION_DESCRIPTION = 'Adderall' where MEDICATION_ID = 57005;
update PATIENT set PAT_PHONE = '346-728-8367' where PAT_ID = 40001;
update PATIENT set PAT_ADDRESS = '63 Sw loop' where PAT_ID = 40002;

update PRESCRIPTION set PRES_DESCRIPTION = 'Twice per day' where PAT_ID = 40013;
update PRESCRIPTION set PRES_REASON = 'fever' where PAT_ID = 40011;

update JOB set JOB_SALARY = 35000 where JOB_TITLE = 'Janitor';
update JOB set JOB_CLEARANCE = 6 where JOB_TITLE = 'Doctor';

update EMPLOYEE set JOB_TITLE = 'Manager' where EMP_ID = 10002;
update EMPLOYEE set EMP_PHONE = '356-236-3780' where EMP_ID = 10001;

update DEPARTMENT set DEPT_DESCRIPTION = 'Emergency Room' where DEPT_ID = 11;
update DEPARTMENT set DEPT_DESCRIPTION = 'Family Doctor' where DEPT_ID = 55;

update TEAM set MEMBER_COUNT = 6 where DEPT_ID = 20001;
update TEAM set MEMBER_COUNT = 2 where DEPT_ID = 20003;

update BUILDING set TEAM_ID = 20002 where BUILD_ID = 101;
update BUILDING set DEPT_ID = 22, TEAM_ID = 20003 where BUILD_ID = 102;

update ROOM set FLOOR_ID = 2 where ROOM_ID = 1101;
update ROOM set FLOOR_ID = 1 where ROOM_ID = 1102;

update PROCEDURE set PROCEDURE_DESCRIPTION = 'CT scan' where PROCEDURE_ID = 58403;
update PROCEDURE set PROCEDURE_TIME = '01-MAY-23 09.30.00.000000 AM' where PROCEDURE_ID =
58401;

update APPOINTMENT set APT_TIME = '01-MAY-23 02.00.00.000000 PM' where APT_ID = 50001;
update APPOINTMENT set PROCEDURE_ID = 58402 where APT_ID = 50002;

update FACILITATOR set APT_ID = 50004 where EMP_ID = 10002;
update FACILITATOR set APT_ID = 50005 where EMP_ID = 10003;

-- 3) At least 5 query scripts to answer questions about your organization and its operations

-- I want to know how many of each medication have been prescribed
select count(PAT_ID), MEDICATION_ID from PRESCRIPTION group by MEDICATION_ID;
-- I want to know how many employees of each job there are
select JOB_TITLE, count(JOB_TITLE) as "# of employees" from EMPLOYEE group by JOB_TITLE order by
COUNT(JOB_TITLE) desc;
-- I want to know who manages each team
select TEAM.TEAM_NAME, EMPLOYEE.EMP_FNAME as "Managers First Name", EMPLOYEE.EMP_LNAME as
"Managers Last Name" from TEAM left join EMPLOYEE on TEAM.EMP_ID = EMPLOYEE.EMP_ID;
-- I want to know what medication Deven Smith was prescibed and when
select PATIENT.PAT_FNAME, PATIENT.PAT_LNAME, MEDICATION.MEDICATION_DESCRIPTION,
PRESCRIPTION.PRES_DATE from PATIENT right join PRESCRIPTION on PRESCRIPTION.PAT_ID =
PATIENT.PAT_ID right join MEDICATION on PRESCRIPTION.MEDICATION_ID =
MEDICATION.MEDICATION_ID where PAT_FNAME = 'Deven';
-- I want to know what, when, where William Collins will be having a procedure
select PATIENT.PAT_FNAME, PATIENT.PAT_LNAME, PROCEDURE.PROCEDURE_DESCRIPTION,
PROCEDURE.PROCEDURE_TIME, ROOM.ROOM_ID, ROOM.BUILD_ID from APPOINTMENT inner join
PATIENT on APPOINTMENT.PAT_ID = PATIENT.PAT_ID inner join ROOM on APPOINTMENT.ROOM_ID =
ROOM.ROOM_ID inner join PROCEDURE on APPOINTMENT.PROCEDURE_ID =
PROCEDURE.PROCEDURE_ID where PATIENT.PAT_FNAME = 'William';

-- 4) At least 4 scripts (update or insert) to demonstrate that you receive errors from the database


-- Tries to insert a duplicate MEDICATION_ID
insert into MEDICATION values (57001, 'Metoprolol');
-- Tries to update BUILD_NAME to a string that's longer than 20 characters
update BUILDING set BUILD_NAME = '111111111111111111111111111111';
-- References a TEAM_ID that doesn't exist
insert into PROCEDURE values (58406, 'Appendectomy', timestamp'2023-04-06 10:00:00', 2, 40006);
-- Tries to insert a null value into MEDICATION_DESCRIPTION
insert into MEDICATION values (57006 , NULL);
```

HOSPITAL DATABASE

Team Members: Abel Lagonell, Dallas DeSimone, Greg Jans, Jaleel Rogers

ORGANIZATION

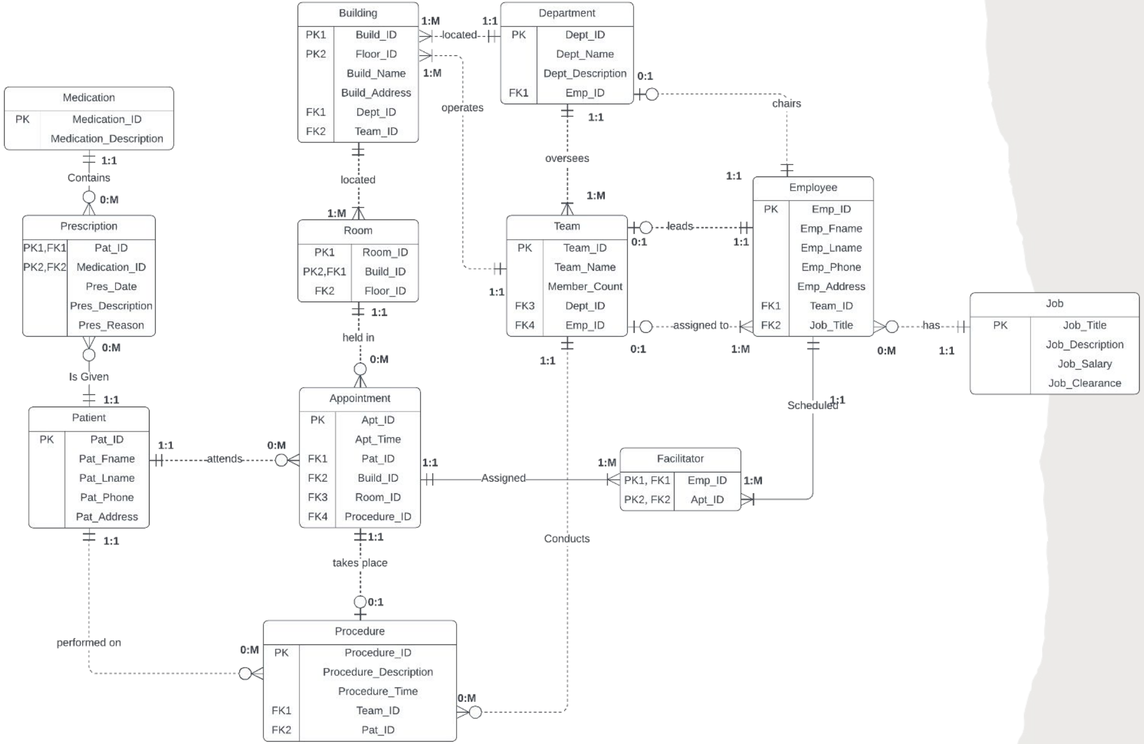
- Hospital Management Database
  - Medium-Sized Hospital
  - Assumes Hospital consists of several buildings and departments
- Why?
  - Extensively relevant to healthcare settings
  - Most complex & largest database
  - Security regulation
    - HIPPA and GDPR

BUSINESS RULES

- A majority of entities have some sort of optional connection
  - Prescription, Procedure, Appointment, etc.
- Attempted to minimize complexity
  - Simplified & limited tables
- Bridge entities
  - Allows for m:n relationships
  - Facilitator connects many employees to many appointments
  - Prescription connects many medications to many patients

ER DIAGRAM

1. We start with the physical entities which include Building and Room.
2. We branch off to the right with the hospital's internal structure, which includes Department, Team, Employee, Job, and Facilitator.
3. We branch off to the left where the patients and their related information is stored, including Patient, Prescription, and Medication.
4. We create the Appointment and Procedure entities which are the core of the daily operations and connect the physical, Internal, and patient entity groups.



Building

PK1	Build_ID
PK2	Floor_ID
	Build_Name
	Build_Address
FK1	Dept_ID
FK2	Team_ID

Department

PK	Dept_ID
	Dept_Name
	Dept_Description
FK1	Emp_ID

Medication

PK	Medication_ID
	Medication_Description

Prescription

PK1,FK1	Pat_ID
PK2,FK2	Medication_ID
	Pres_Date
	Pres_Description
	Pres_Reason

Room

PK1	Room_ID
PK2,FK1	Build_ID
FK2	Floor_ID

Team

PK	Team_ID
	Team_Name
	Member_Count
FK3	Dept_ID
FK4	Emp_ID

Employee

PK	Emp_ID
	Emp_Fname
	Emp_Lname
	Emp_Phone
	Emp_Address
FK1	Team_ID
FK2	Job_Title

Job

PK	Job_Title
	Job_Description
	Job_Salary
	Job_Clearance

Patient

PK	Pat_ID
	Pat_Fname
	Pat_Lname
	Pat_Phone
	Pat_Address

Appointment

PK	Apt_ID
	Apt_Time
FK1	Pat_ID
FK2	Build_ID
FK3	Room_ID
FK4	Procedure_ID

Facilitator

PK1, FK1	Emp_ID
PK2, FK2	Apt_ID

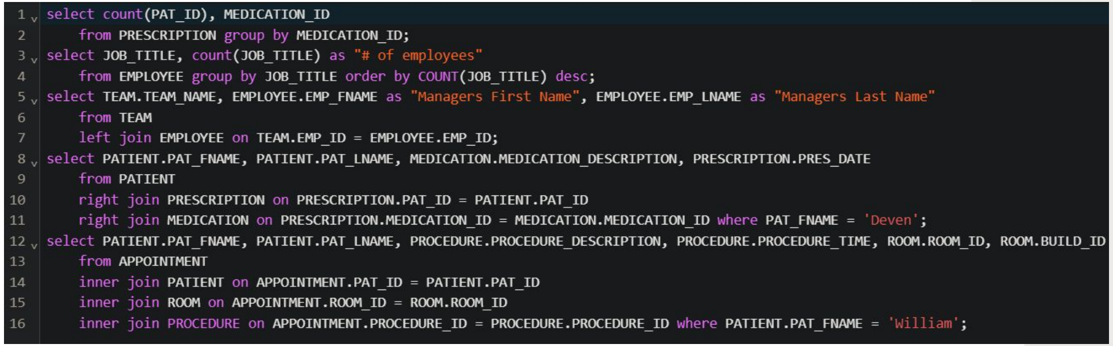
Procedure

PK	Procedure_ID
	Procedure_Description
	Procedure_Time
FK1	Team_ID
FK2	Pat_ID

QUERIES

Queries to answer questions about the organization

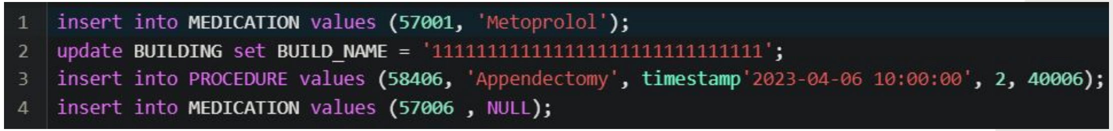
1. I want to know how many of each medication have been prescribed.
2. I want to know how many employees of each job there are.
3. I want to know who manages each team.
4. I want to know what medication Deven Smith was prescribed and when.
5. I want to know what, when, where William Collins will be having a procedure.



```sql
select count(PAT_ID), MEDICATION_ID
    from PRESCRIPTION group by MEDICATION_ID;
select JOB_TITLE, count(JOB_TITLE) as "# of employees"
    from EMPLOYEE group by JOB_TITLE order by COUNT(JOB_TITLE) desc;
select TEAM.TEAM_NAME, EMPLOYEE.EMP_FNAME as "Managers First Name", EMPLOYEE.EMP_LNAME as "Managers Last Name"
    from TEAM
    left join EMPLOYEE on TEAM.EMP_ID = EMPLOYEE.EMP_ID;
select PATIENT.PAT_FNAME, PATIENT.PAT_LNAME, MEDICATION.MEDICATION_DESCRIPTION, PRESCRIPTION.PRES_DATE
    from PATIENT
    right join PRESCRIPTION on PRESCRIPTION.PAT_ID = PATIENT.PAT_ID
    right join MEDICATION on PRESCRIPTION.MEDICATION_ID = MEDICATION.MEDICATION_ID where PAT_FNAME = 'Deven';
select PATIENT.PAT_FNAME, PATIENT.PAT_LNAME, PROCEDURE.PROCEDURE_DESCRIPTION, PROCEDURE.PROCEDURE_TIME, ROOM.ROOM_ID, ROOM.BUILD_ID
    from APPOINTMENT
    inner join PATIENT on APPOINTMENT.PAT_ID = PATIENT.PAT_ID
    inner join ROOM on APPOINTMENT.ROOM_ID = ROOM.ROOM_ID
    inner join PROCEDURE on APPOINTMENT.PROCEDURE_ID = PROCEDURE.PROCEDURE_ID where PATIENT.PAT_FNAME = 'William';
```

Queries to check validity of tables

1. Tries to insert a duplicate MEDICATION_ID.
2. Tries to update BUILD_NAME to a string that's longer than 20 characters.
3. References a TEAM_ID that doesn't exist.
4. Tries to insert a null value into MEDICATION_DESCRIPTION.



```sql
insert into MEDICATION values (57001, 'Metoprolol');
update BUILDING set BUILD_NAME = '111111111111111111111111111111';
insert into PROCEDURE values (58406, 'Appendectomy', timestamp'2023-04-06 10:00:00', 2, 40006);
insert into MEDICATION values (57006 , NULL);
```