# DATABASE MANAGEMENT SYSTEMS PROJECT

Scenario

A doctor examines different patients. A doctor can write prescriptions to patients. There is a nurse attached to each doctor. People are also inherited. A software system is created for a hospital. The manager wants to store and manage some information such as doctors, patients, prescriptions in the hospital. The database is expected to contain information about people, doctors, patients, nurses, drugs, prescriptions, and appointments.

Business Rules

-Each doctor has name,e-mail,telephonenumber,specialty.

-Prescription has an id.

-A drug has an id, name, drug type(syrup, pill,injection).

-Appointment has an ID ,appointment date, description.

-Each patient has a name, e-mail, telephone number.

-Each room has an id, name.

-Each clinic has an id, name.

-Each nurse has an name, e-mail, phone number.

-Inthesystem,eachpersonhaspersonid,name,e-mail,telephonenumberinformation.Also,apersoncan exist in just three categories: a doctor, a nurse and a patient. One person can not be all three.

Peoplearedistinguishedfromeachotherbytheirid.

-Adoctorcanmakemorethanoneappointment.Onlyonedoctorcanbevisitedforanappointment.

-Aprescriptioniscreated byadoctor.Adoctormaynot writeaprescription ormaywritemorethan one prescription.

-Adrugmaybeprescribedmorethanonce,andaprescriptionmaycontainmorethanonedrug.

-A patient can be at an appointment, but a patient can have more than one appointment.

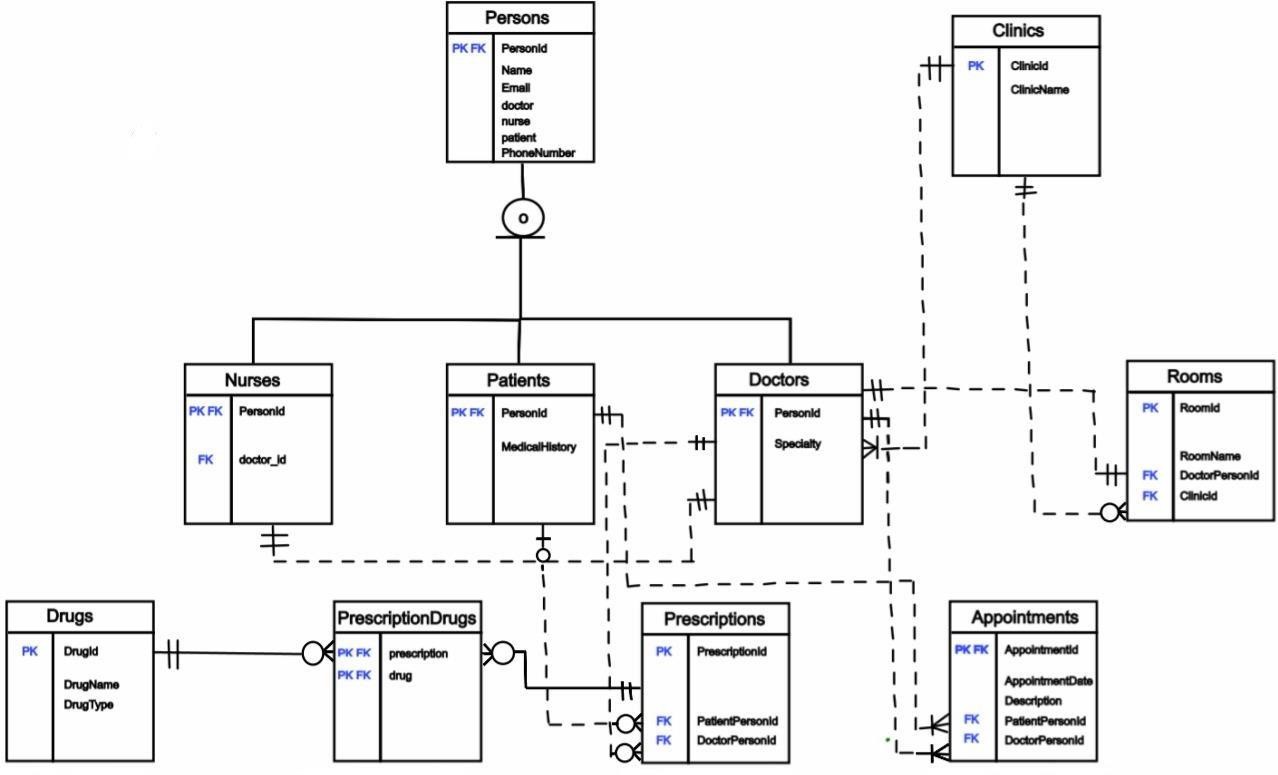
-A patient may receive no prescription a tall, or may receive a prescription. A prescription may be written for a patient, or may not be written at all.

-A room has only one doctor. A doctor has only one room.

-A clinic must have at least one doctor, but there can also be more than one doctor but a doctor can only be at a clinic.

-A nurse can only work with one doctor, a doctor can only work with one nurse.

-There are one or more rooms in a clinic. A room be longs to only one clinic.



Relational Schema (Textual Representation):

* Persons(PersonId:bigint,Name:varchar(30),Email:varchar(50),PhoneNumber:bigint, doctor:bool, nurse:bool, patient:bool)
* Nurses(PersonId:bigint,DoctorId:bigint)
* Patients(PersonId:bigint,MedicalHistory:varchar(100))
* Doctors(PersonId:bigint,Speciality:varchar(30))
* Clinics(ClinicId:bigint,ClinicName:varchar(30))
* Rooms(RoomId:bigint,RoomName:varchar(30),DoctorPersonId:bigint,ClinicId:bigint)
* Appointments(AppoinmentId:bigint,AppoinmentDate:timestamp,Description:VARCHAR(60), PatientPersonId:bigint, DoctorPersonId:bigint)
* Prescriptions(PrescriptionId:bigint,PatientPersonId:bigint,DoctorPersonId:bigint)
* PrescriptionDrugs(Prescription:bigint,Drug:bigint)
* Drugs(DrugId:bigint,DrugName:varchar(50),DrugType:varchar(30))

SQLstatementstocreatethedatabasewiththedatainit

--CreationProcesses

CREATETABLEIFNOTEXISTS"EFMigrationsHistory"( "MigrationId" character varying(150) NOT NULL, "ProductVersion" character varying(32) NOT NULL,

CONSTRAINT"PK EFMigrationsHistory"PRIMARYKEY("MigrationId")

);

CREATETABLE"Clinics"(

"ClinicId"bigintGENERATEDBYDEFAULTASIDENTITY, "ClinicName" VARCHAR(30) NOT NULL,

CONSTRAINT"PK\_Clinic"PRIMARYKEY("ClinicId")

);

CREATETABLE"Drugs"(

"DrugId"bigintGENERATEDBYDEFAULTASIDENTITY, "DrugName" VARCHAR(50) NOT NULL,

"DrugType" VARCHAR(30) NOT NULL, CONSTRAINT"PK\_Drug"PRIMARYKEY("DrugId")

);

CREATETABLE"Persons"(

"PersonId"bigintGENERATEDBYDEFAULTASIDENTITY, "Name" VARCHAR(30) NOT NULL,

"Email"VARCHAR(50)NOTNULL,

"PhoneNumber"textNOTNULL, doctor boolean NOT NULL, patient boolean NOT NULL,nurse boolean NOT NULL,

CONSTRAINT"PK\_Person"PRIMARYKEY("PersonId")

);

CREATETABLE"Doctors"(

"PersonId" bigint NOT NULL, "Specialty"VARCHAR(30)NOTNULL,

"ClinicId"bigintNOTNULL,

CONSTRAINT"PK\_Doctor"PRIMARYKEY("PersonId"),

CONSTRAINT"FK\_Doctor\_Person\_PersonId"FOREIGNKEY("PersonId")REFERENCES "Persons" ("PersonId"),

CONSTRAINT"FK\_Doctor\_Clinic\_ClinicId"FOREIGNKEY("ClinicId")REFERENCES "Clinics" ("ClinicId")

);

CREATE TABLE "Patients" ( "PersonId"bigintNOTNULL,

"MedicalHistory" VARCHAR(100) NOT NULL, CONSTRAINT"PK\_Patient"PRIMARYKEY("PersonId"),

CONSTRAINT"FK\_Patient\_Person\_PersonId"FOREIGNKEY("PersonId")REFERENCES "Persons" ("PersonId")

);

CREATETABLE"Nurses"(

"PersonId" bigint NOT NULL, "DoctorPersonId"bigintNOTNULL,

CONSTRAINT"PK\_Nurse"PRIMARYKEY("PersonId"),

CONSTRAINT"FK\_Nurse\_Person\_PersonId"FOREIGNKEY("PersonId")REFERENCES

"Persons"("PersonId"),

CONSTRAINT"FK\_Nurse\_Doctor\_DoctorPersonId"FOREIGNKEY("DoctorPersonId") REFERENCES "Doctors" ("PersonId")

);

CREATETABLE"Rooms"(

"RoomId"bigintGENERATEDBYDEFAULTASIDENTITY, "RoomName" VARCHAR(30) NOT NULL,

"DoctorPersonId"bigintNOTNULL, "ClinicId" bigint NOT NULL,

CONSTRAINT"PK\_Room"PRIMARYKEY("RoomId"),

CONSTRAINT"FK\_Room\_Clinic\_ClinicId"FOREIGNKEY("ClinicId")REFERENCES "Clinics" ("ClinicId"),

CONSTRAINT"FK\_Room\_Doctor\_DoctorPersonId"FOREIGNKEY("DoctorPersonId") REFERENCES "Doctors" ("PersonId")

);

CREATETABLE"Appointments"(

"AppointmentId"bigintGENERATEDBYDEFAULTASIDENTITY, "AppointmentDate" timestamp with time zone NOT NULL, "Description" VARCHAR(60) NOT NULL,

"PatientPersonId"bigintNOTNULL, "DoctorPersonId"bigintNOTNULL,

CONSTRAINT"PK\_Appointment"PRIMARYKEY("AppointmentId"),

CONSTRAINT"FK\_Appointment\_Doctor\_DoctorPersonId"FOREIGNKEY("DoctorPersonId") REFERENCES "Doctors" ("PersonId"),

CONSTRAINT"FK\_Appointment\_Patient\_PatientPersonId"FOREIGNKEY("PatientPersonId") REFERENCES "Patients" ("PersonId")

);

CREATETABLE"Prescriptions"(

"PrescriptionId"bigintGENERATEDBYDEFAULTASIDENTITY, "DoctorPersonId" bigint NOT NULL,

"PatientPersonId"bigintNOTNULL,

CONSTRAINT"PK\_Prescription"PRIMARYKEY("PrescriptionId"),

CONSTRAINT"FK\_Prescription\_Doctor\_DoctorPersonId"FOREIGNKEY("DoctorPersonId") REFERENCES "Doctors" ("PersonId"),

CONSTRAINT"FK\_Prescription\_Patient\_PatientPersonId"FOREIGNKEY("PatientPersonId") REFERENCES "Patients" ("PersonId")

);

CREATETABLE"PrescriptionDrugs"( prescription bigint NOT NULL, drug bigint NOT NULL,

CONSTRAINT"PK\_PrescriptionDrug"PRIMARYKEY(prescription,drug),

CONSTRAINT"FK\_PrescriptionDrug\_Drug\_drug"FOREIGNKEY(drug)REFERENCES "Drugs" ("DrugId"),

CONSTRAINT"FK\_PrescriptionDrug\_Prescription\_prescription"FOREIGNKEY(prescription) REFERENCES "Prescriptions" ("PrescriptionId")

);

# --INSERTIONPROCESSES

--Clinics

INSERTINTO"Clinics"("ClinicName")VALUES ('General Medicine Clinic'),

('Pediatrics Clinic'), ('Cardiology Center'), ('DermatologyClinic'),

('Neurology Institute'), ('Orthopedic Hospital'), ('OncologyDepartment'), ('ENT Clinic'), ('Psychiatry Clinic'), ('GastroenterologyUnit');

--Drugs

INSERTINTO"Drugs"("DrugName","DrugType")VALUES ('Drug 5', 'Tablet'),

('Drug9','Injection'),

('Drug10','Syrup'),

('Drug8','Capsule'),

('Drug5','Tablet'),

('Drug1','Tablet'),

('Drug3','Injection'),

('Drug3','Syrup'),

('Drug8','Capsule'),

('Drug9','Injection');

--Persons

INSERTINTO"Persons"("Name","Email","PhoneNumber",doctor,patient,nurse)VALUES ('Person 1', ['person1@example.com',](mailto:%27person1@example.com)'1234567891', True, False, False),

('Person 2', ['person2@example.com',](mailto:%27person2@example.com)'1234567892', True, False, False), ('Person 3', ['person3@example.com',](mailto:%27person3@example.com)'1234567893', True, False, False), ('Person 4', ['person4@example.com',](mailto:%27person4@example.com)'1234567894', False, False, True), ('Person 5', ['person5@example.com',](mailto:%27person5@example.com)'1234567895', False, True, False), ('Person 6', ['person6@example.com',](mailto:%27person6@example.com)'1234567896', True, False, False), ('Person 7', ['person7@example.com',](mailto:%27person7@example.com)'1234567897', False, False, True), ('Person 8', ['person8@example.com',](mailto:%27person8@example.com)'1234567898', False, False, True), ('Person 9', ['person9@example.com',](mailto:%27person9@example.com)'1234567899', False, False, True), ('Person10',['person10@example.com',](mailto:%27person10@example.com)'12345678910',False,False,True);

--Doctors

INSERTINTO"Doctors"("PersonId","Specialty","ClinicId")VALUES (21, 'Dermatology', 7),

(22,'Dermatology',7),

(23,'Dermatology',10),

(24,'Pediatrics', 7),

(25,'Dermatology',10);

--Patients

INSERTINTO"Patients"("PersonId","MedicalHistory")VALUES (26, 'History 9'),

(27,'History1'),

(28,'History2'),

(29,'History9'),

(30,'History7');

--Nurses

INSERTINTO"Nurses"("PersonId","DoctorPersonId")VALUES (6, 5),

(7, 1),

(8, 4),

(9, 1),

(10, 2);

--Rooms

INSERTINTO"Rooms"("RoomName","DoctorPersonId","ClinicId")VALUES ('Room 3', 5, 7),

('Room4',4,5),

('Room4',5,10),

('Room8',3,7),

('Room3',2,3),

('Room2',3,4),

('Room3',5,3),

('Room4',2,8),

('Room4',1,4),

('Room3',3,7);

--Appointments

INSERTINTO"Appointments"("AppointmentDate","Description","PatientPersonId", "DoctorPersonId") VALUES

(NOW()+INTERVAL'1days','Description4',7,3),

(NOW()+INTERVAL'2days','Description6',10,3),

(NOW()+INTERVAL'3days','Description2',7,3),

(NOW()+INTERVAL'4days','Description10',10,5),

(NOW()+INTERVAL'5days','Description7',7,4),

(NOW()+INTERVAL'6days','Description9',9,1),

(NOW()+INTERVAL'7days','Description6',8,3),

(NOW()+INTERVAL'8days','Description6',10,1),

(NOW()+INTERVAL'9days','Description2',9,3),

(NOW()+INTERVAL'10days','Description7',7,5);

--Prescriptions

INSERTINTO"Prescriptions"("DoctorPersonId","PatientPersonId")VALUES (1, 6),

(3, 9),

(4, 9),

(5, 9),

(1, 7),

(1, 9),

(3, 6),

(4, 6),

(3, 7),

(3, 10);

--PrescriptionDrugs

INSERTINTO"PrescriptionDrugs"(prescription,drug)VALUES (15, 14),

(12, 17),

(10, 15),

(19, 16),

(16, 14),

(17, 13),

(11, 17),

(12, 18),

(12, 19),

(15, 15);

//UPDATEProcess

# --TRIGGERS

--TRIGER1

--TrigerTABLE

CREATETABLE"AppointmentLog"( "LogId" SERIAL PRIMARY KEY,

"AppointmentId"bigintNOTNULL, "LogMessage" text NOT NULL,

"LogTime"timestampNOTNULLDEFAULTNOW()

);

CREATEORREPLACEFUNCTIONlog\_appointment() RETURNS TRIGGER AS $$

# BEGIN

INSERTINTO"AppointmentLog"("AppointmentId","LogMessage") VALUES (NEW."AppointmentId", 'New appointment added.'); RETURN NEW;

# END;

$$LANGUAGEplpgsql;

CREATETRIGGERafter\_appointment\_insert BEFORE INSERT ON "Appointments"

# FOREACHROW

EXECUTEFUNCTIONlog\_appointment();

INSERTINTO"Persons"("Name","Email","PhoneNumber",doctor,patient,nurse)VALUES ('Person 11', ['person12@example.com',](mailto:%27person12@example.com)'1234567891', False, True, False);

INSERTINTO"Persons"("Name","Email","PhoneNumber",doctor,patient,nurse)VALUES ('Person 12', ['person13@example.com',](mailto:%27person13@example.com)'1234567891', True, False, False);

--Patient

INSERTINTO"Patients"("PersonId","MedicalHistory")VALUES (13, 'History 11');

--Doctor

INSERTINTO"Doctors"("PersonId","Specialty","ClinicId")VALUES (14, 'Dermatology', 7);

--Addanew appointment

INSERTINTO"Appointments"("AppointmentDate","Description","PatientPersonId", "DoctorPersonId")

VALUES(NOW()+INTERVAL'2days','Routinecheck-up',13, 14);

-- Log tablosundaki kaydı kontrol et SELECT\*FROM"AppointmentLog";

# --TRIGER2

CREATEORREPLACEFUNCTIONcascade\_delete\_patient\_appointments() RETURNS TRIGGER AS $$

# BEGIN

DELETEFROM"Appointments"WHERE"PatientPersonId"=OLD."PersonId"; RETURN OLD;

# END;

$$LANGUAGEplpgsql;

CREATETRIGGERafter\_patient\_delete BEFORE DELETE ON "Patients"

# FOREACHROW

EXECUTEFUNCTIONcascade\_delete\_patient\_appointments();

--Let'strytodeletethe"Patients"

DELETEFROM"Persons"WHERE"PersonId"=3; DELETEFROM"Patients"WHERE"PersonId"=3;

# --TRIGER3

--TrigerTABLE

CREATETABLE"DoctorNameLog"( "LogId" SERIAL PRIMARY KEY,

"DoctorPersonId" bigint NOT NULL, "OldName" VARCHAR(50) NOT NULL, "NewName"VARCHAR(50)NOTNULL,

"ChangeTime"timestampNOTNULLDEFAULTNOW()

);

CREATEORREPLACEFUNCTIONlog\_doctor\_name\_changes() RETURNS TRIGGER AS $$

# BEGIN

--Ifthenamehaschanged,savetheoldandnewnamestothelogtable. IF NEW."Name" IS DISTINCT FROM OLD."Name" THEN

INSERTINTO"DoctorNameLog"("DoctorPersonId","OldName","NewName","ChangeTime") VALUES (OLD."PersonId", OLD."Name", NEW."Name", NOW());

END IF;

RETURNNEW;

END;

$$LANGUAGEplpgsql;

CREATETRIGGERdoctor\_name\_update\_trigger BEFORE UPDATE ON "Persons"

# FOREACHROW

WHEN(OLD."doctor"=TRUEANDNEW."doctor"=TRUE)

EXECUTEFUNCTIONlog\_doctor\_name\_changes();

--Let's update

UPDATE"Persons"SET"Name"='Memati'WHERE"PersonId"=3;

--Let'scheckthelogtable

SELECT\*FROM"DoctorNameLog";

# --TRIGER4

-- Triger TABLECREATETABLE"ClinicReport"(

"ReportId"SERIALPRIMARYKEY,

"ClinicId"bigintNOTNULL,

"ClinicName"VARCHAR(50)NOTNULL,

"LastUpdated"timestampNOTNULLDEFAULTNOW()

);

CREATEORREPLACEFUNCTIONupdate\_clinic\_reports() RETURNS TRIGGER AS $$

# BEGIN

--Updatereporttableifclinicnameorinformationhas changed

IFNEW."ClinicName"ISDISTINCTFROMOLD."ClinicName"THEN UPDATE "ClinicReport"

SET"ClinicName"=NEW."ClinicName", "LastUpdated" = NOW()

WHERE"ClinicId"=NEW."ClinicId";

END IF;

RETURNNEW;

END;

$$LANGUAGEplpgsql;

CREATETRIGGERclinic\_update\_trigger BEFORE UPDATE ON "Clinics"

# FOREACHROW

EXECUTEFUNCTIONupdate\_clinic\_reports();

--Let'supdatetheclinicname

UPDATE"Clinics"SET"ClinicName"='CityHealthCenter'WHERE"ClinicId"=6;

--Let'schecktheclinicalreporttable SELECT \* FROM "ClinicReport";

# --FUNCTIONS

--Function1

CREATE OR REPLACE FUNCTION get\_patients\_by\_doctor(doctor\_id BIGINT) RETURNSTABLE(PersonIdBIGINT,PersonNameTEXT,MedicalHistoryTEXT)AS$$ BEGIN

# RETURNQUERY

SELECTp."PersonId",p."Name",pa."MedicalHistory" FROM "Persons" p

JOIN"Patients"paONp."PersonId"= pa."PersonId"

JOIN"Appointments"aONa."PatientPersonId"=pa."PersonId" WHERE a."DoctorPersonId" = doctor\_id;

# END;

$$LANGUAGEplpgsql;

SELECT\*FROMget\_patients\_by\_doctor(3);--3numaralıdoktoraaithastalar

--Function2

CREATEORREPLACEFUNCTIONget\_rooms\_by\_clinic(clinic\_idBIGINT) RETURNS TABLE (RoomName TEXT, DoctorName TEXT) AS $$

# BEGIN

RETURNQUERY

SELECTr."RoomName",p."Name"ASDoctorName FROM "Rooms" r

JOIN"Doctors"dONr."DoctorPersonId"=d."PersonId" JOIN "Persons" p ON d."PersonId" = p."PersonId" WHERE r."ClinicId" = clinic\_id;

# END;

$$LANGUAGEplpgsql;

SELECT\*FROMget\_rooms\_by\_clinic(7);--7numaralıkliniğinodaları

--Function3

CREATEORREPLACEFUNCTIONget\_drugs\_by\_patient(patient\_idBIGINT) RETURNS TABLE (DrugName TEXT, PrescriptionId BIGINT) AS $$

# BEGIN

RETURNQUERY

SELECTd."DrugName",pd."prescription"ASPrescriptionId FROM "PrescriptionDrugs" pd

JOIN"Prescriptions"pONpd."prescription"=p."PrescriptionId" JOIN "Drugs" d ON pd."drug" = d."DrugId"

WHEREp."PatientPersonId"=patient\_id; END;

$$LANGUAGEplpgsql;

SELECT\*FROMget\_drugs\_by\_patient(6);--6numaralıhastanınilaçları

--Function4

CREATEORREPLACEFUNCTIONget\_appointment\_count\_by\_doctor(doctor\_idBIGINT) RETURNS INT AS $$

# DECLARE

appointment\_countINT; BEGIN

# SELECTCOUNT(\*)

INTOappointment\_count FROM "Appointments"

WHERE"DoctorPersonId"=doctor\_id;

RETURNappointment\_count; END;

$$LANGUAGEplpgsql;

SELECTget\_appointment\_count\_by\_doctor(3);--3numaralıdoktoruntoplamrandevuları

