PHASE 2: SUBMISSION CHECKLIST/SIGNOFF SHEET

GROUP#: 1 **GROUP NAME:** Supreme Nerds

Deliverables:

- [Revised] Description of the organization
- [Revised] ER diagram with min/max specifications
- [Revised] Constraints not in ER diagram
- [Revised] Relational Schema with Referential Integrity
- Queries with brief description
- DDL statements
- Completed PostgreSQL Implementation (Attached Team SQL script)

Group Assessment

- Group Status Report

We have each reviewed the contents of this deliverable.

Phase Leader	Delaney Sauer	____________________________
Phase Recorder	Kristin Fasiang	____________________________
Phase Checker	Anthony Orso	____________________________
Technical Advisor	Natalie Stegman	____________________________

1. Introduction

Provide here a general description of the organization and the contact person.

2. Requirements Description

The HOSPITAL DATABASE (HDB) helps store information related to patient care, including medications, treating physicians, diagnostic labels, and the location that service was rendered. Relationships among these entities further help organize instances of patient care into meaningful patterns for querying and data analysis.

- Each patient is stored in the database with crucial information related to their identity. The Patient entity includes a PID that acts a primary key in relational database design, as well as important demographic information such as a multivalued address entry, gender, date of birth, and their first and last name. To ensure the database has some quantitative variables for interesting queries, we've also added attributes for height, weight, and blood pressure. The patient entity interacts with the diagnosis, medication, hospital room, and provider entity. Due to the complexities of hospital systems, we've taken into consideration that patients can interact with hospitals in outpatient and inpatient settings. This is an important consideration because primary care involves fewer interactions with medical professionals and, thus, a patient may not take medication or have a diagnosis. We've included zero as a minimum value in most relationships to capture this nuance. On the other hand, patients receiving inpatient treatment are guaranteed to have one or more diagnoses, take medication, and potentially see many doctors and stay in multiple rooms during a single stay. The only instances where we mandate participation in a relationship are with the Room and Provider entities. Patients must be in private rooms due to HIPAA, and the very definition of being a patient is receiving care from a healthcare professional. However, because this database is designed to hold historical records versus updating current ones, we must allow the patient to receive care in multiple rooms and see more than one doctor over time.
- Provider is an entity that represents the healthcare workers who are responsible for delivering the main service to the patient. For example, we wouldn't list as the provider a tech who draws blood during a physical with the patient's primary care physician. However, if a patient were to go to an individually run lab in the hospital with a doctor's order for certain tests, the phlebotomist who draws the blood in that individual interaction would be logged as the provider in that instance of the relationship. The primary key for the provider entity is PrID, which is a unique integer that belongs to one and only one provider. We also have as attributes the first name and last name (PrFName and PrLName), type (e.g. MD, DO, NP, RN), and specialty (e.g. anesthesia, gastroenterology). The provider entity has a one-to-many relationship with patients, as they can

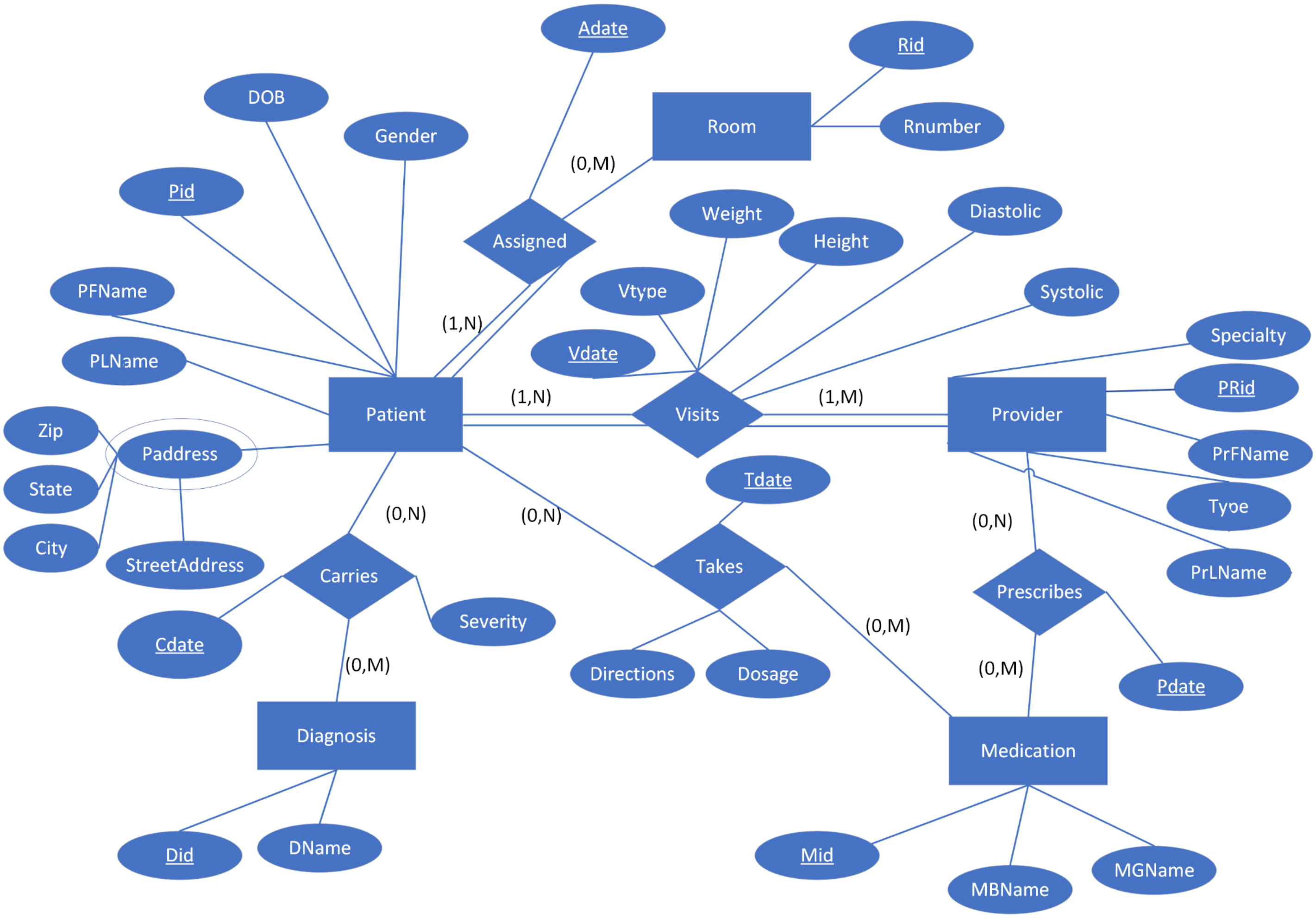
see as many patients as they'd like and are mandated by conditions of employment to see patients. The provider is also connected to the Medication entity with a zero-to-many cardinality. Some providers legally cannot prescribe medication, such as psychotherapists and nurses, while physicians are, in theory, free to prescribe an infinite combination of medications.

- As mentioned earlier, the Room entity is needed to trace the location of patient-provider interactions. This entity has the attributes RID (primary key integer) and RNumber (room number). In theory, a room could be unused, so the minimum in the cardinality ratio is zero. At first, our instinct was to set that maximum at one due to the constraint that only one patient at a time can be in the room due to HIPAA requirements around patient confidentiality. However, the purpose of the database is to keep historical information, so each instance of Room can occur many times. To maintain integrity constraints, we've added an attribute to the assigned relationship that identifies the date, patient ID, and RID as a composite key to prevent duplicate entries.
- Diagnosis is another entity important in hospital database design.The attributes are DID (unique integer primary key attached to a single diagnosis), Dname (name of diagnosis), and Severity (low, medium, or high), Because some diagnoses are very rare, it's possible a hospital could never see a patient with a one-in-a-million diagnosis. For this reason, the cardinality ratio for Diagnosis in its relationship to Patient starts at zero. Of course, there is no finite limit to how many unique diagnoses can be given as well, so the maximum is many.
- The final entity is Medication, which stores information related to medications. The primary key is MID, which is a unique integer that identifies the medication. The other attributes are MDname (drug name) and MGname (generic name). This entity is related to the Patient who takes a given medication and Provider who prescribes it.
- The Visits relation connects Patient and Provider and is a many-to-many relationship. For this reason, the primary key is a composite of the primary keys of Patient and Provider--PID and PrID, respectively--as well as the Date of the visit. Since this database collects historical records, the possibility of a patient seeing a provider more than once is likely, the Date is added to prevent duplicate entries. The addition of Date to the primary key is predicated on the integrity constraint that a patient cannot see the same provider more than once a day.
- The Assigned relation connects the Patient to a room for their visit with the Provider. The many-to-many relationship between the Patient and Room entities means the Assigned relational schema must use its primary keys to make a composite primary key. The attribute Adate that captures the date for which the room is assigned must also be part of the composite primary key since, in theory, a patient could be in that same room for future appointments.

- The Carries relation connects Patient and Diagnosis and is a nonrequired relationship that has no limit to the number of relations that can exist between the two entities. Not all patients will have diagnoses, and it's possible for a diagnosis to not be paired with a patient if it's an exceedingly rare one. There's also an attribute for the relation called Cdate, which is the date the diagnosis is provided or re-entered in the system during a follow-up appointment for chronic illness. This attribute is part of the composite primary key that also contains the primary keys of Patient and Diagnosis.
- The Takes relation connects Patient to Medication. It also is a nonrequired relationship that has no limit to the number of relations that can exist between the two entities. Patients do not need to be prescribed medication and some medications that get imported from exhaustive lists into databases may not be prescribed. The relation has attribute Tdate, dosage, and directions, with Tdate being the primary key. Tdate represents the date the medication is prescribed to the patient, and that combined with the primary keys of Patient and Medication comprise the primary key.
- The final relation in our ER diagram is Prescribes, which captures the fact that the Provider prescribes Medication to the Patient entity. Some providers cannot prescribe medication while others can prescribe as many as they see fit for patient care, so it has a zero-to-many cardinality in the direction of medication. Also, it's possible a medication is never prescribed at a hospital even if it is logged in the database even though others can be prescribed without limit, so it too has a zero-to-many cardinality. The Pdate, which is the date the provider prescribes the medication, is taken together with the primary keys of the Provider and Medication entities to form a composite key due to the relation schema constraint of many-to-many relations.

3. ER Diagram

The figure below shows the ER diagram of the Healthcare Database



4. ER Diagram Uncaptured Constraints

The following is a list of constraints that are not captured by the ER diagram of HDB:

- The dosage for medication and the values for weight, height, and blood pressure must be positive numbers
    - Dosages are in milligrams, weight is in kilograms, height in centimeters, and blood pressure in mmHg
- A provider cannot visit more than 30 patients in a day, and cannot visit more than one patient at the same time.
- A patient cannot visit the same provider twice on the same day
- A patient can only be actively assigned to one room at a time during a Provider visit due to privacy concerns.

5. Relational Schema

This section provides the relational schema with referential integrity and the relational table details.

5.1 Relational Schema with Referential Integrity

Patient (Pid, DOB, Pfname, Plname, Paddress, Gender, BP, Height, Weight)
Provider (PRid, PRfname, PRlname, Type, Specialty)
Medication (Mid, MBname, MGname)
Diagnosis (Did, severity, Dname)
Room(Rid, Rnumber)
Visits (Pid, PRid, Vdate, Vtype)
Foreign key (Pid) references Patient (Pid)
Foreign key (PRid) references Provider (PRid)
Takes (Pid, Mid, Tdate, Directions, Dosage)
Foreign key (Pid) references Patient (Pid)
Foreign key (Mid) references Med (Mid)
Prescribes (Mid, PRid, Pdate)
Foreign key (PRid) references Provider (PRid)
Foreign key (Mid) references Med (Mid)
Carry (Did, Pid, Cdate)
Foreign key (Did) references Diagnosis (Did)
Foreign key (Pid) references Patient (Pid)
Assigned(Rid, Pid, Adate)
Foreign key (Rid) references Room (Rid)

Foreign key (Pid) references Patient (Pid)

5.2 Relational Table Details

The relational schema given in Section 5.1 was mapped into the following tables in the HOSPITAL DATABASE. Primary keys have been underlined. Tables that have multiple attributes underlined represent composite keys.

Table Name	Attribute	Description
Patient	Pid	Unique patient ID
	Plname	Last name of patient
	Pfname	First name of patient
	DOB	Date of birth of patient (MM/DD/YYYY)
	Paddress	Multilevel attribute that is the address of the patient, broken into street address, zip, city, and state
	Gender	Gender of the patient: F, M, or NB
Provider	PRid	Unique provider ID
	PRfname	First name of provider
	PRlname	Last name of provider
	Ptype	Type of provider (nurse, doctor, PA)
	specialty	Medical specialty of provider
Medication	Mid	Unique course ID
	MGname	General name of medication
	MBname	Brand name of medication
Diagnosis	Did	Unique diagnosis ID
	Dname	Name of diagnosis
Room	Rid	Unique room ID
	Rnumber	Labeled number of room in hospital
Visits	PID	Unique patient ID
	PrID	Unique provider ID
	Vtype	Type of visit (annual physical, surgery, PT) delineated in CPT code
	Vdate	Date and time of the visit (MM/DD/YYYY)
	Rid	Unique room ID

	BP	Blood pressure of patient: multilevel attribute that includes diastolic and systolic
	Height	Height in cm of patient
	Weight	Weight in Kg of patient
Takes	PID	Unique patient ID
	MID	Unique medication ID
	Tdate	Start date of taking medication (MM/DD/YYYY)
	Dosage	Amount of the medication to be taken
	Directions	Instructions for frequency and timing of medication
Prescribes	Prid	Unique provider ID
	MID	Unique medication ID
	Pdate	Date of when medication was prescribed (MM/DD/YYYY)
Carries	PID	Unique patient ID
	DID	Unique diagnosis ID
	Cdate	Date of when diagnosis was made (MM/DD/YYYY)
	severity	Indicates the severity of diagnosis: low, medium, high
Assigned	Adate	Date of when patient is assigned to room (MM/DD/YYYY)
	Rid	Unique room ID
	Pid	Unique patient ID

6. Queries

The following table summarizes the queries in the HOSPITAL DATABASE.

Query Name	Description	Output	Relations Accessed
surplusRx	For each provider who visited patients or wrote a prescription on a given day (in this example, 10-10-2022), print the surplus number of prescriptions written (defined as the number of prescriptions written that was greater than the number of visits performed) in descending order.	● prID ● prFname ● prLname ● surplusRx	● Provider ● Visits ● Prescribes
healthDiabetic	We want to select patients who have diabetes and stratify by Type 1 and Type 2 to measure which ones report better health outcomes. This is measured through reporting the average weight, diastolic blood pressure, and systolic blood pressure.	● Did ● Dname ● avg_weight ● avg_systolic ● avg_diastolic	● visits ● carry ● diagnosis
mostDiabeticPx	This is a more complex query where we first want to join visits and carry to create a view with provider, patient, and diagnostic information. We'll group by provider and the condition where the diagnosis is Type 1 or Type 2 diabetes. Then we will use the count function to see how many diabetic patients each provider is seeing. The second query will pull the name, provider ID, and the maximum count of patients from the view.	● PRid ● PRfname ● PRlname ● numPatients	● View t ● visits ● carry ● provider
mostRecentVisit	For each patient, print the date of the most recent visit they had and what provider it was with. Sort the output by the date of the most recent visit (ascending order).	● pID ● Pfname ● Plname ● MostRecentVdate ● Vtype ● PRid ● PRlname ● Ptype ● Pspecialty	● Patient ● Visits ● Provider

7. DDL + SQL

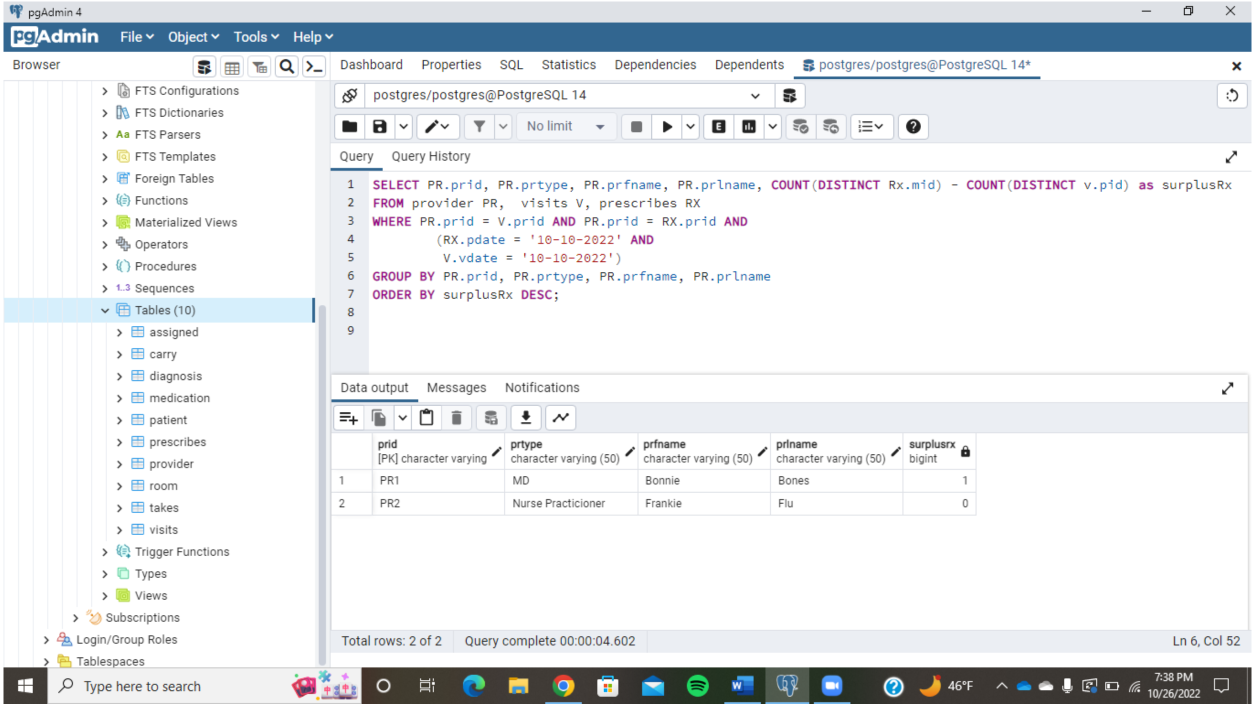
The following is an SQL definition of the tables for the HOSPITAL DATABASE.

```
CREATE TABLE patient
(Pid varchar PRIMARY KEY,
 Pdob date, Pfname varchar (50),
 Plname varchar (50),
 Paddress varchar (75),
 Pcity varchar (50),
 Pstate varchar (2),
 Pzip varchar (10),
 Pgender varchar(2)
);

CREATE TABLE provider
(PRid varchar PRIMARY KEY,
 PRfname varchar(50),
 PRlname varchar(50),
 PRtype varchar(50),
 PRspecialty varchar(50)
);

CREATE TABLE medication
(Mid varchar PRIMARY KEY,
 MBname varchar(50),
 MGname varchar(50)
);

CREATE TABLE diagnosis
(Did varchar PRIMARY KEY,
 Dname varchar (50)
);

CREATE TABLE room
(Rid varchar PRIMARY KEY,
 Rnumber varchar (5)
);

CREATE TABLE visits
(Vdate date,
 Vtype varchar(50),
 Pid varchar NOT NULL,
 PRid varchar NOT NULL,
 systolic numeric,
 diastolic numeric,
 height numeric,
 weight numeric,
PRIMARY KEY (Pid,PRid,Vdate),
 FOREIGN KEY (Pid)
        REFERENCES patient(Pid),
 FOREIGN KEY (PRid)
```

```
                              REFERENCES provider(PRid)
);

CREATE TABLE takes
(Pid varchar NOT NULL,
 Mid varchar NOT NULL,
 Tdate date,
 directions varchar (50),
 dosage numeric,
 PRIMARY KEY (Pid, Mid, Tdate),
 FOREIGN KEY (Pid)
                              REFERENCES patient(Pid),
 FOREIGN KEY (Mid)
                              REFERENCES medication(Mid)
);

CREATE TABLE prescribes
(Pdate date,
 PRid varchar NOT NULL,
 Mid varchar NOT NULL,
 PRIMARY KEY (PRid, Mid, Pdate),
 FOREIGN KEY (PRid)
                              REFERENCES provider(PRid),
 FOREIGN KEY (Mid)
                             REFERENCES medication(Mid)
);

CREATE TABLE carry
(Cdate date,
 Did varchar NOT NULL,
 Pid varchar NOT NULL,
 severity varchar (10),
 PRIMARY KEY (Did,Pid,Cdate),
 FOREIGN KEY (Did)
                              REFERENCES diagnosis(Did),
 FOREIGN KEY (Pid)
                              REFERENCES patient(Pid)
);

CREATE TABLE assigned
(Adate date,
 Rid varchar NOT NULL,
 Pid varchar NOT NULL,
 PRIMARY KEY (Rid,Pid,Adate),
 FOREIGN KEY (Rid)
                              REFERENCES room(Rid),
 FOREIGN KEY (Pid)
                              REFERENCES patient(Pid)
);
```

Q1. surplusRx

```
SELECT PR.prid, PR.prtype, PR.prfname, PR.prlname, COUNT(DISTINCT Rx.mid) - COUNT(DISTINCT v.pid) as
surplusRx
FROM provider PR,  visits V, prescribes RX
WHERE PR.prid = V.prid AND PR.prid = RX.prid AND
                    (RX.pdate = '10-10-2022' AND
                     V.vdate = '10-10-2022')
GROUP BY PR.prid, PR.prtype, PR.prfname, PR.prlname
ORDER BY surplusRx DESC;
```



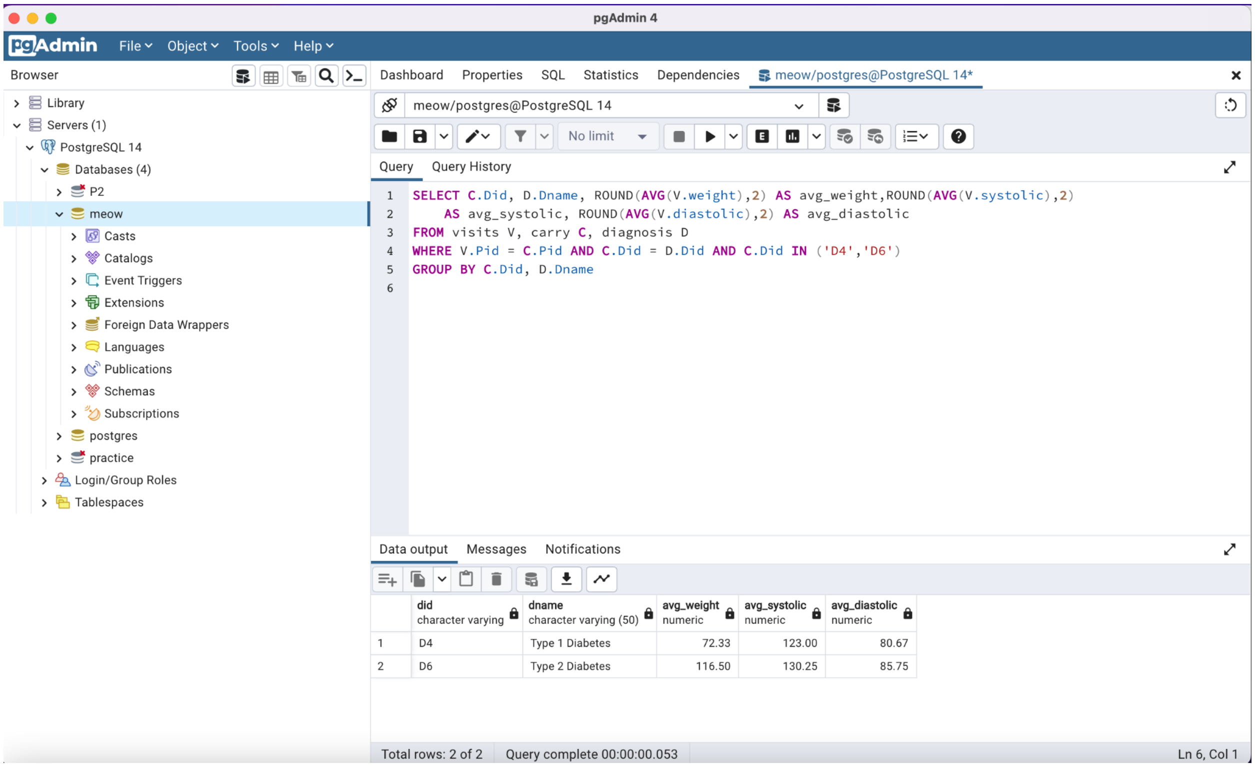
```
SELECT PR.prid, PR.prtype, PR.prfname, PR.prlname, COUNT(DISTINCT Rx.mid) - COUNT(DISTINCT v.pid) as surplusRx
FROM provider PR,  visits V, prescribes RX
WHERE PR.prid = V.prid AND PR.prid = RX.prid AND
        (RX.pdate = '10-10-2022' AND
         V.vdate = '10-10-2022')
GROUP BY PR.prid, PR.prtype, PR.prfname, PR.prlname
ORDER BY surplusRx DESC;
```

	prid [PK] character varying	prtype character varying (50)	prfname character varying (50)	prlname character varying (50)	surplusrx bigint
1	PR1	MD	Bonnie	Bones	1
2	PR2	Nurse Practicioner	Frankie	Flu	0

Total rows: 2 of 2 Query complete 00:00:04.602 Ln 6, Col 52

Q2. healthDiabetic

```
SELECT C.Did, D.Dname, ROUND(AVG(V.weight),2) AS avg_weight,ROUND(AVG(V.systolic),2)
          AS avg_systolic, ROUND(AVG(V.diastolic),2) AS avg_diastolic
FROM visits V, carry C, diagnosis D
WHERE V.Pid = C.Pid AND C.Did = D.Did AND C.Did IN ('D4','D6')
GROUP BY C.Did, D.Dname;
```



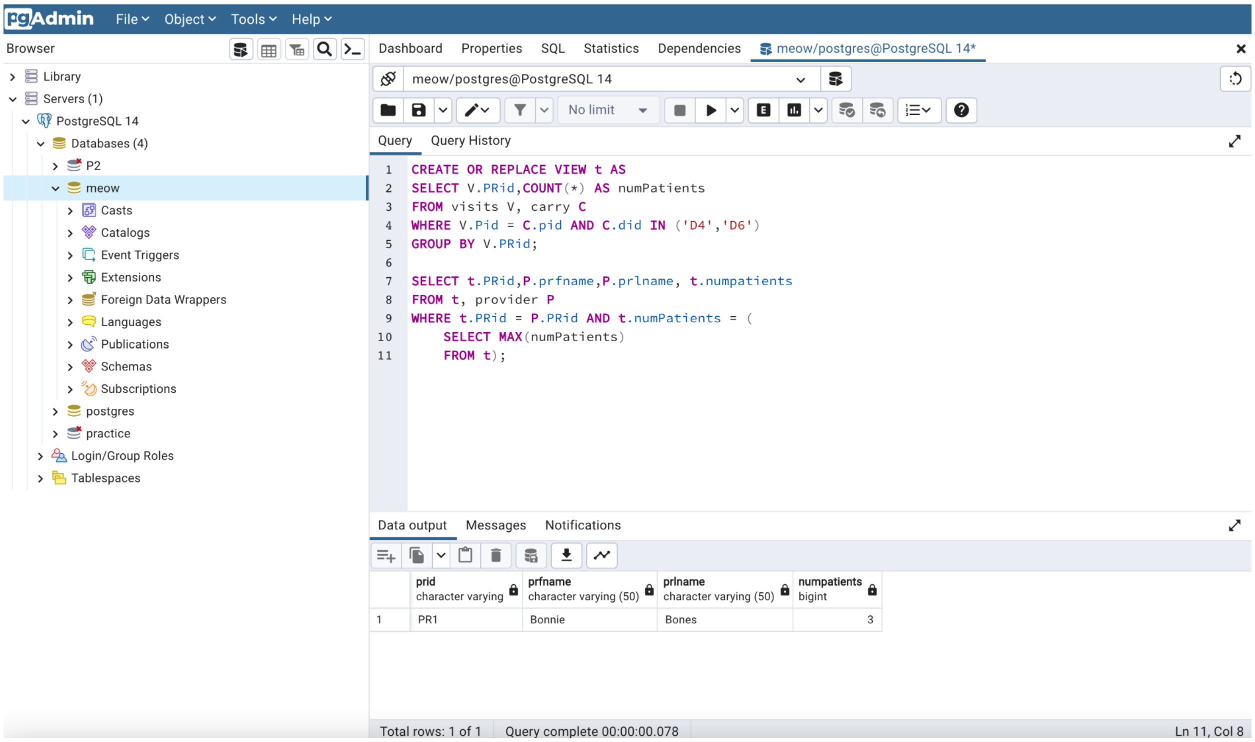
```
SELECT C.Did, D.Dname, ROUND(AVG(V.weight),2) AS avg_weight,ROUND(AVG(V.systolic),2)
    AS avg_systolic, ROUND(AVG(V.diastolic),2) AS avg_diastolic
FROM visits V, carry C, diagnosis D
WHERE V.Pid = C.Pid AND C.Did = D.Did AND C.Did IN ('D4','D6')
GROUP BY C.Did, D.Dname
```

	did character varying	dname character varying (50)	avg_weight numeric	avg_systolic numeric	avg_diastolic numeric
1	D4	Type 1 Diabetes	72.33	123.00	80.67
2	D6	Type 2 Diabetes	116.50	130.25	85.75

Total rows: 2 of 2 Query complete 00:00:00.053 Ln 6, Col 1

Q3. mostDiabeticPx

```
CREATE OR REPLACE VIEW t AS
SELECT V.PRid,COUNT(*) AS numPatients
FROM visits V, carry C
WHERE V.Pid = C.pid AND C.did IN ('D4','D6')
GROUP BY V.PRid;

SELECT t.PRid,P.prfname,P.prlname, t.numpatients
FROM t, provider P
WHERE t.PRid = P.PRid AND t.numPatients = (
        SELECT MAX(numPatients)
        FROM t);
```



```sql
CREATE OR REPLACE VIEW t AS
SELECT V.PRid,COUNT(*) AS numPatients
FROM visits V, carry C
WHERE V.Pid = C.pid AND C.did IN ('D4','D6')
GROUP BY V.PRid;

SELECT t.PRid,P.prfname,P.prlname, t.numpatients
FROM t, provider P
WHERE t.PRid = P.PRid AND t.numPatients = (
    SELECT MAX(numPatients)
    FROM t);
```

	prid character varying	prfname character varying (50)	prlname character varying (50)	numpatients bigint
1	PR1	Bonnie	Bones	3

Total rows: 1 of 1 Query complete 00:00:00.078 Ln 11, Col 8

Q4: mostRecentVisit

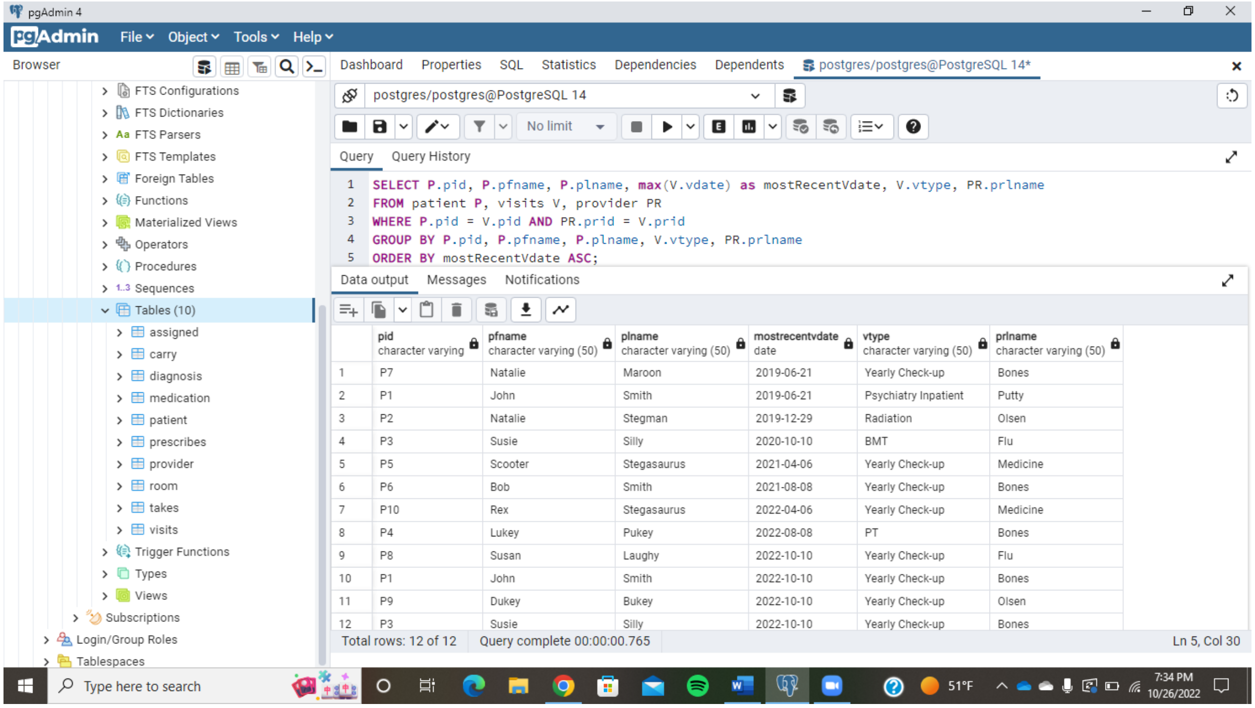
SELECT P.pid, P.pfname, P.plname, max(V.vdate) as mostRecentVdate, V.vtype, PR.prlname

FROM patient P, visits V, provider PR

WHERE P.pid = V.pid AND PR.prid = V.prid

GROUP BY P.pid, P.pfname, P.plname, V.vtype, PR.prlname

ORDER BY mostRecentVdate ASC;



```sql
SELECT P.pid, P.pfname, P.plname, max(V.vdate) as mostRecentVdate, V.vtype, PR.prlname
FROM patient P, visits V, provider PR
WHERE P.pid = V.pid AND PR.prid = V.prid
GROUP BY P.pid, P.pfname, P.plname, V.vtype, PR.prlname
ORDER BY mostRecentVdate ASC;
```

	pid character varying	pfname character varying (50)	plname character varying (50)	mostrecentvdate date	vtype character varying (50)	prlname character varying (50)
1	P7	Natalie	Maroon	2019-06-21	Yearly Check-up	Bones
2	P1	John	Smith	2019-06-21	Psychiatry Inpatient	Putty
3	P2	Natalie	Stegman	2019-12-29	Radiation	Olsen
4	P3	Susie	Silly	2020-10-10	BMT	Flu
5	P5	Scooter	Stegasaurus	2021-04-06	Yearly Check-up	Medicine
6	P6	Bob	Smith	2021-08-08	Yearly Check-up	Bones
7	P10	Rex	Stegasaurus	2022-04-06	Yearly Check-up	Medicine
8	P4	Lukey	Pukey	2022-08-08	PT	Bones
9	P8	Susan	Laughy	2022-10-10	Yearly Check-up	Flu
10	P1	John	Smith	2022-10-10	Yearly Check-up	Bones
11	P9	Dukey	Bukey	2022-10-10	Yearly Check-up	Olsen
12	P3	Susie	Silly	2022-10-10	Yearly Check-up	Bones

Total rows: 12 of 12 Query complete 00:00:00.765 Ln 5, Col 30

GROUP STATUS REPORT

GROUP #: 1 **GROUP NAME: Supreme Nerds** **PHASE #:** 2

Dates & attendance at group meetings in this phase:

Saturday, Oct 22	1:30-	2:00	All group members present
Tuesday, Oct 25	11:00-	11:30	All met before class to discuss queries and DDL
Wednesday, Oct 26	7:30-	8:00	All

CONTRIBUTIONS OF GROUP MEMBERS

Leader: Delaney Sauer

- Visited Dr. Silva office hours to review changes
- Checked that ER Diagram and DDL represented the same relationships/entities
- QUERY IMPLEMENTATION: Query 2: healthDiabetic
- TABLE DDL: patient, provider
- INSERT: patient, provider

Recorder: Kristin Fasiang

- Prepared group status report
- QUERY IMPLEMENTATION: Quarry 4: mostRecentVisit
- TABLE DDL: medication, diagnosis, assigned
- INSERT: medication, diagnosis, assigned

Phase Checker: Anthony Orso

- Met with Prof. Silva to review progress
- QUERY IMPLEMENTATION: Query 3: mostDiabeticPx
- TABLE DDL: room, visits, carry
- INSERT: room, visits, carry

Technical Advisor: Natalie Stegman

- Revised document to include verified DDL
- INSERT: visits, takes, prescribes
- TABLE DDL: visits, takes, prescribes
- QUERY IMPLEMENTATION: Query 1: surplusRx
- interviewed expert user