Project 2
Josh Crail, Sathvik Chereddy

Table of Contents

Forward Engineering

Person(persID, name)
Patient(pID, persID)
- persID is FK to Person

Worker(wID, type, pID)
- pID is FK to Person

Doctor(dID, wID)
- wID is FK to Worker

Medication(mID, medName)
Prescription(pID, dID, mID, date, doctype, dosage)
- pID is FK to Patient
- dID is FK to Doctor
- mID is FK to Medication

Shift(sID, startTime, endTime, date)
Nurse(nID, wID)
- wID is FK to Worker

Phone(persID, pNum)
- persID is FK to Person

Visits(sID, pID, result)
- sID is FK to Shift
- pID is FK to Patient

WorksIn(sID, wID)
- sID is FK to Shift
- wID is FK to Worker

DDL

```
CREATE TABLE Person (
        persID INT UNSIGNED,
        name VARCHAR(20),
PRIMARY KEY (persID));

CREATE TABLE Patient (
        pID INT UNSIGNED,
        persID INT UNSIGNED,
        PRIMARY KEY (pID),
        FOREIGN KEY (persID) REFERENCES Person(persID));

CREATE TABLE Worker (
        wID INT UNSIGNED,
        persID INT UNSIGNED,
        workertype VARCHAR(20),
        PRIMARY KEY (wID),
        FOREIGN KEY (persID) REFERENCES Person(persID),
CHECK (workertype IN ("doctor", "nurse")) );

CREATE TABLE Doctor (
        dID INT UNSIGNED,
        wID INT UNSIGNED,
        PRIMARY KEY (dID),
        FOREIGN KEY (wID) REFERENCES Worker(wID));

CREATE TABLE Medication (
        mID INT UNSIGNED,
        medName VARCHAR(20 ),
        PRIMARY KEY (mID));

CREATE TABLE Prescription (
        pID INT UNSIGNED,
        dID INT UNSIGNED,
        mID INT UNSIGNED,
        date DATE,
        doctype VARCHAR(20),
dosage VARCHAR(20),
PRIMARY KEY (pID, dID, mID),
FOREIGN KEY (pID) REFERENCES Patient(pID),
FOREIGN KEY (dID) REFERENCES Doctor(dID),
FOREIGN KEY (mID) REFERENCES Medication(mID),
CHECK (doctype IN ("triage", "case")) );
```

```
CREATE TABLE Shift (
        sID INT UNSIGNED,
        startTime TIME,
        endTime TIME,
        date DATE,
        PRIMARY KEY (sID));

CREATE TABLE Nurse (
        nID INT UNSIGNED,
        wID INT UNSIGNED,
        PRIMARY KEY (nID),
        FOREIGN KEY (wID) REFERENCES Worker(wID));

CREATE TABLE Phone (
        pnum VARCHAR(20),
        persID INT UNSIGNED,
PRIMARY KEY (pnum, persID),
FOREIGN KEY (persID) REFERENCES Person(persID));



CREATE TABLE Visits (
        sID INT UNSIGNED,
        pID INT UNSIGNED,
        result VARCHAR(20) NOT NULL,
PRIMARY KEY (sID, pID),
FOREIGN KEY (sID) REFERENCES Shift(sID),
FOREIGN KEY (pID) REFERENCES Patient(pID),
CHECK (result IN ("sent", "admitted")) );

CREATE TABLE Works_In (
        sID INT UNSIGNED,
        wID INT UNSIGNED,
        PRIMARY KEY (sID, wID),
FOREIGN KEY (sID) REFERENCES Shift(sID),
FOREIGN KEY (wID) REFERENCES Worker(wID));
```

Time Log

Who	Location	Date	Start	End	Length	Topic	Action items
Josh C.	Benton	11/3	10:00am	10:30pm	0.50	Forward engineering project ERD	Get consensus on forward engineering with Sathvik C.
Sathvik C.	Home	11/7	12:00pm	12:30pm	0.50	Forward engineering ERD given in project specification	Get consensus on forward engineered relation model with Josh C.
Josh C. Sathvik C.	Home Zoom	11/7	7:30pm	8:30pm	1.00	Comparing & verifying forwarded engineered relation model, and converting relational model to MySQL DDL	Implement Database in Google Cloud
Josh C. Sathvik C.	Benton	11/9	4:10pm	6:10pm	2.00	Implementing the database in Google Cloud. Creating data for the database. Writing queries for so that data demands and mock data can be checked for accuracy	Adding mock data into the database and checking the data demands
Sathvik C.	Home	11/9	7:00pm	9:00pm	2.00	Creating mock data for relations in database that satisfy minimum 5 tuple requirement	Adding mock data into database and verifying that requirements are met
Josh C.	Home	11/9	11:00pm	12:00am	1.00	Adding mock data into the database and checking the data demands	Checking the data demands and adding more data so that all will return at least five tuples.
Josh C. Sathvik C.	Home Zoom	11/10	6:30pm	9:30pm	3.00	Checking to make sure that data demands are satisfied. Checking to make sure that queries for data demands return the proper tuples.	N/A
Total length:	Josh C.	1.5	Sathvik C.	2.5	Team	6.0	

Table Values

Doctor Table:

```
1 •  SELECT * FROM MidCityEr.Doctor;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

dID	wID
4	0
0	2
1	4
3	5
2	8
5	11
NULL	NULL

Medication Table:

```
1 •  SELECT * FROM MidCityEr.Medication;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

mID	medName
0	naproxen
1	advil
2	alavert
3	palifermin
4	xigris
5	galsulfate
NULL	NULL

Nurse Table:

```
1 •  SELECT * FROM MidCityEr.Nurse;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

nID	wID
0	1
1	3
2	6
3	7
4	9
5	10
NULL	NULL

Patient Table:

```
1 •  SELECT * FROM MidCityEr.Patient;
```

pID	persID
7	0
2	2
8	4
0	5
1	6
6	7
9	8
4	9
10	10
3	13
5	15
NULL	NULL

Person Table:

```
1 •  SELECT * FROM MidCityEr.Person;
```

persID	name
1	Boris Angeloni
2	Toni Babochini
3	Jacobi Krakowski
4	Homer Simpson
5	Peter Parker
6	Benjamin Tennyson
7	Jack Black
8	Chris Pratt
9	Jill Manning
10	Matt Penner
11	Jane Doe
12	Alexander Hamilton
13	Forrest Gump
14	David Oppenheimer
15	Jim Bob

Phone Table:

```
1 •  SELECT * FROM MidCityEr.Phone;
```

pnum	persID
258-530-4812	0
813-592-3773	1
390-507-9214	2
752-734-5882	3
250-659-4323	4
864-361-0175	5
445-456-3089	6
303-506-1142	7
250-554-6483	8
937-393-5718	9
318-156-2129	10
395-438-1968	11
568-561-5974	12
120-528-9349	13
140-898-1816	14

Prescription Table:

```
1 • SELECT * FROM MidCityEr.Prescription;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

pID	dID	mID	date	doctype	dosage
0	4	1	2021-11-09	case	20 mg
2	0	0	2021-11-09	triage	30 g
4	1	0	2021-11-09	triage	65 ml
5	5	0	2021-11-09	triage	30 mg
6	3	0	2021-11-09	triage	100 mg
7	3	0	2021-11-09	triage	0.1 l
8	3	5	2021-11-09	triage	10.0 g
9	3	4	2021-11-09	triage	10 ml
10	3	2	2021-11-09	case	125 g
NULL	NULL	NULL	NULL	NULL	NULL

Shift Table:

```
1 • SELECT * FROM MidCityEr.Shift;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

sID	startTime	endTime	date
0	09:00:00	10:00:00	2021-11-09
1	10:00:00	11:00:00	2021-11-09
2	11:00:00	12:00:00	2021-11-09
3	12:00:00	13:00:00	2021-11-09
4	13:00:00	14:00:00	2021-11-09
8	17:00:00	18:00:00	2021-11-09
9	09:00:00	10:00:00	2020-01-09
10	10:00:00	11:00:00	2020-02-09
11	11:00:00	12:00:00	2020-03-09
12	12:00:00	13:00:00	2020-04-09
13	13:00:00	14:00:00	2020-05-09
14	17:00:00	18:00:00	2020-06-09
NULL	NULL	NULL	NULL

Visits Table:

```
1 • SELECT * FROM MidCityEr.Visits;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

sID	pID	result
0	0	admitted
1	9	admitted
1	10	admitted
2	2	sent
3	7	sent
3	8	admitted
4	4	sent
4	5	sent
8	6	sent
NULL	NULL	NULL

Worker Table:

```
1 •  SELECT * FROM MidCityEr.Worker;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

wID	persID	workertype
0	5	doctor
1	6	nurse
2	2	doctor
3	13	nurse
4	9	doctor
5	0	doctor
6	12	nurse
7	14	nurse
8	1	doctor
9	3	nurse
10	11	nurse
11	15	doctor
NULL	NULL	NULL

Works_In:

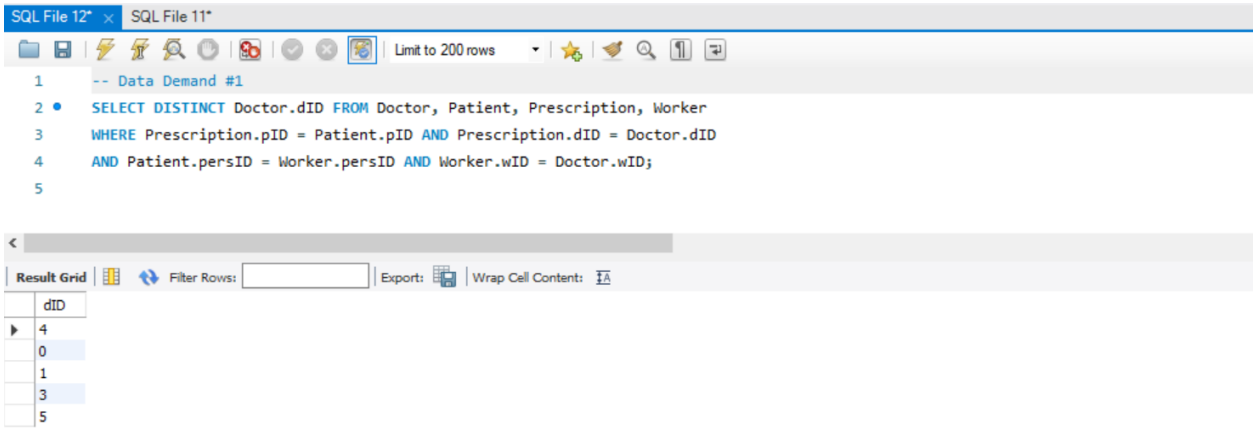
```
1 •  SELECT * FROM MidCityEr.Works_In;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

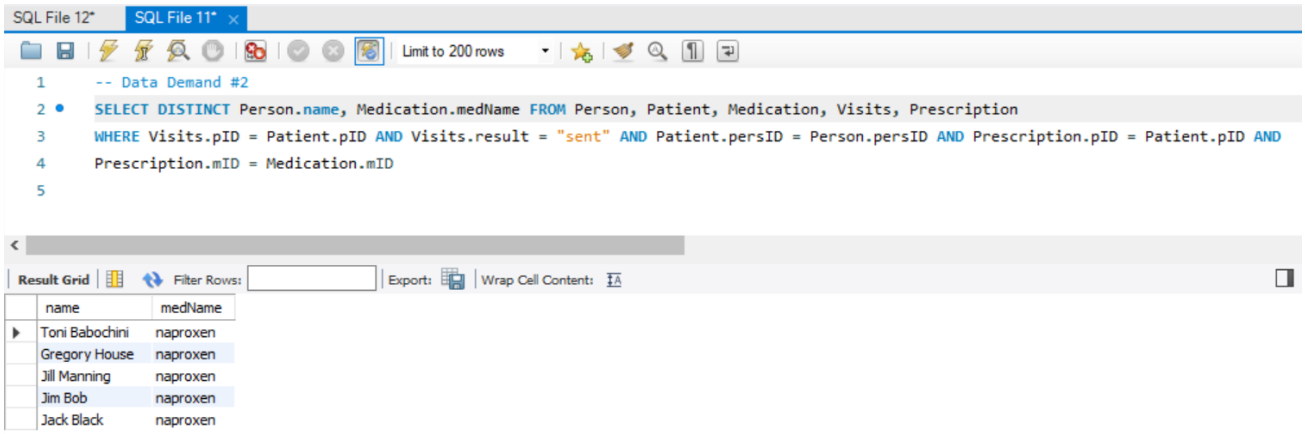
sID	wID
9	0
9	1
14	1
4	2
1	3
2	4
0	5
1	5
3	5
4	5
8	5
12	6
13	7
10	8
12	9
13	10

Data Demands

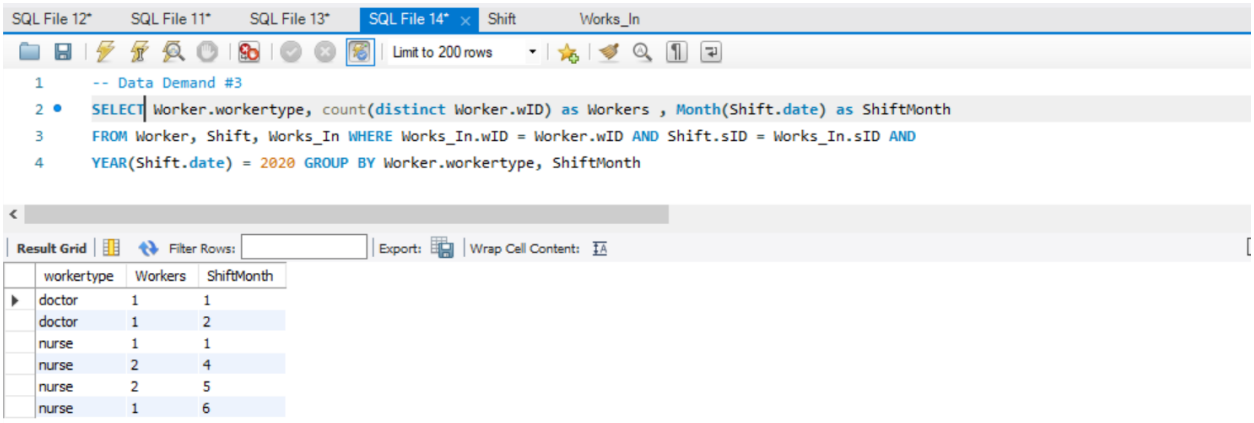
Data Demand #1



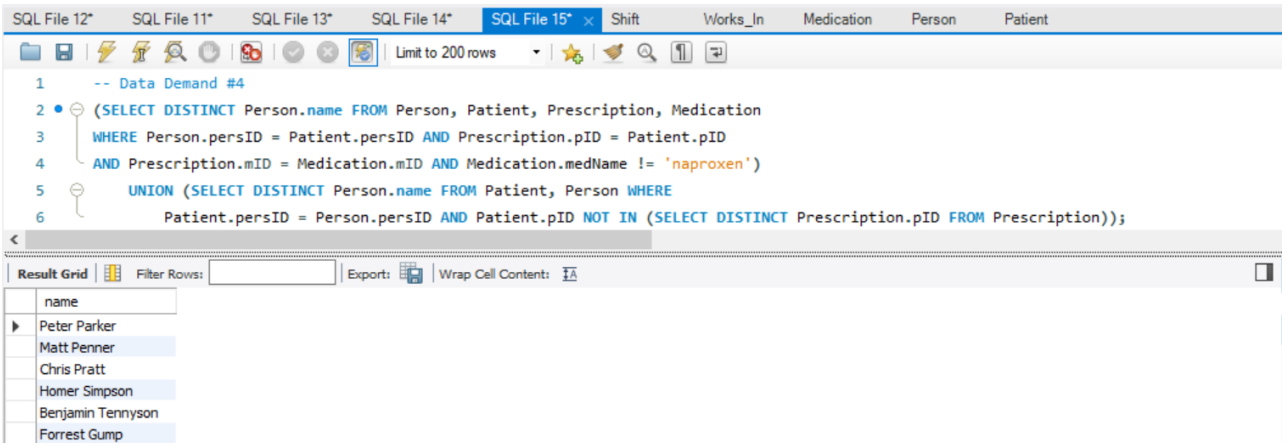
Data Demand #2



Data Demand #3



Data Demand #4



Data Demand #5