Project Step 7: Portfolio Assignment

Project Name: A Hospital Operations Database
Group Members: Keaton Hartman & Lawrence Yeh

URL of hosted, functioning project website:
http://flip3.engr.oregonstate.edu:8099/index

Executive Summary:

The Hospital Operations Database has made many changes starting from beginning to end. One of the biggest changes early on was adding the nursesPatients conjunction to satisfy the many-to-many relationship in our database. We originally had nursesPatients as an entity incorrectly, but our peer feedback made note that it is supposed to be a conjunction to form the relationship, having only nurseID and patientID as it's attributes. Our peers later recommended we add a search filter and a navigation bar, which were key components in our database website. The navigation bar makes it easy to move from each of the entity pages, while the search filter allows the user to quickly easily find information.

Later on in our project progression, we ran multiple into issues with the database setup (DDL query). Our peer feedback mentioned they were not able to run our DDL query through phpMyAdmin. Initially we thought our Schema's relationship web was incorrect, making the tables unable to connect to each other properly. We thought we had found the root cause, being our foreign keys were not set correctly because phpMyAdmin kept giving us this error: "foreign key constraint is incorrectly formed." However, after several step submissions, discussing with the TA, and debugging, we found that the database provided by OSU was somehow keeping track of deleted foreign key constraints, even though the original table these constraints were attached to was deleted. This prevented a re-creation of both the tables and the foreign keys, when they otherwise should have been valid to use. When trying fix foreign keys, we renamed 'nursesPatients' as 'nursesPatients2', and were unable to go back to naming nursesPatients because of this issue. The most important change to our project was during Steps 4 and 5, when we properly implemented JOIN clause to connect tables together (per TA recommendation). This was advantageous in allowing our database to properly display all the relevant data in our tables on the appropriate webpages.

Lastly, in order to implement the many-to-many relationship and display it to the user on a webpage basis, we found a creative alternative solution. On the final column of the Nurses and Patients tables, we added a "Manage" button; the manage button leads to the nursesPatients conjunction portion of our database. On this subpage, you can select between two drop downs tabs, one for patient and one for nurse, to create the nurse-patient many-to-many relationship (which will then be added to the conjunction table showing the relationship exists). Due to the conjunction table, these newly formed relationships are now displayed in both the Nurses and Patients tables. All of these changes from beginning to end allowed us to create a fully functioning database website for hospital operations.

Project Outline and Database Outline, ERD and Schema:

<u>Project Outline:</u>

      Kaiser Unpermanente is a small healthcare company that owns several 500-bed hospitals across the United States. Recently the hospital directors have had trouble keeping track of their 200+ medical staff, the average of 400-500 active patients, and current prescriptions with just paper folders from day-to-day. They decided to create a simple internal website on a per-hospital basis to help keep track of each hospital's patients, nurses, doctors, and even the prescriptions.

      The database driven site is meant for use by the 70 or so doctors and clinical secretaries per hospital to keep track of which of the 100+ nurses are currently working with which patients, and which doctors are working with which nurses. It also gives each doctor the ability to keep track of each patient they are caring for and which prescriptions they've given out. The database includes enough data to facilitate contact between nurses, doctors, and their patients/prescriptions. It is meant for organizing day-to-day operations of up to 1000 person hospitals but is still detailed enough to be used as a starter for other more permanent patient records if needed.

Database Outline:

Additional Database Details:
- Numbers 1 through 4 are entities (Doctors, Patients, Nurses, Prescriptions) but number 5 describes the many to many relationship (nursesPatients)
- For simplicity, prescriptions are attached/tied to patients rather than being the other way around
- Patient's connection to prescription is automatically NULL'd on deletion of prescription (in place of UPDATE to avoid possible errors)

1. **Doctors** – records the details of employed Doctors
- doctorID: int, unique, not NULL, PK
- doctorSpecialty: varchar
- doctorName: varchar, not NULL
- doctorPhone: int, unique, not NULL
- Relationship: 1:M between Doctors and Patients, doctorID as a FK inside of Patients
- Relationship: 1:M between Doctors and Prescriptions, doctorID as a FK inside of Prescriptions

2. **Patients** – records the details of admitted Patients
- patientID: int, unique, not NULL, PK
- patientName: varchar, not NULL
- patientPhone: int, unique, not NULL
- patientRoom: int
- patientInsuranceNo: int
- doctorID: int, unique, not NULL, FK
- prescriptionID: int, unique, FK
- Relationship: 1:M between Patients and Doctors, patientID as a FK inside of Doctors
- Relationship: M:M between Patients and Nurses, patientID as a FK inside of nursesPatients
- Relationship: 1:1 between Patients and Prescriptions

3. **Nurses** – records the details of employed Nurses
- nurseID: int, unique, not NULL, PK
- nurseName: varchar, not NULL
- nursePhone: int, unique, not NULL
- nurseFloor: int, not NULL
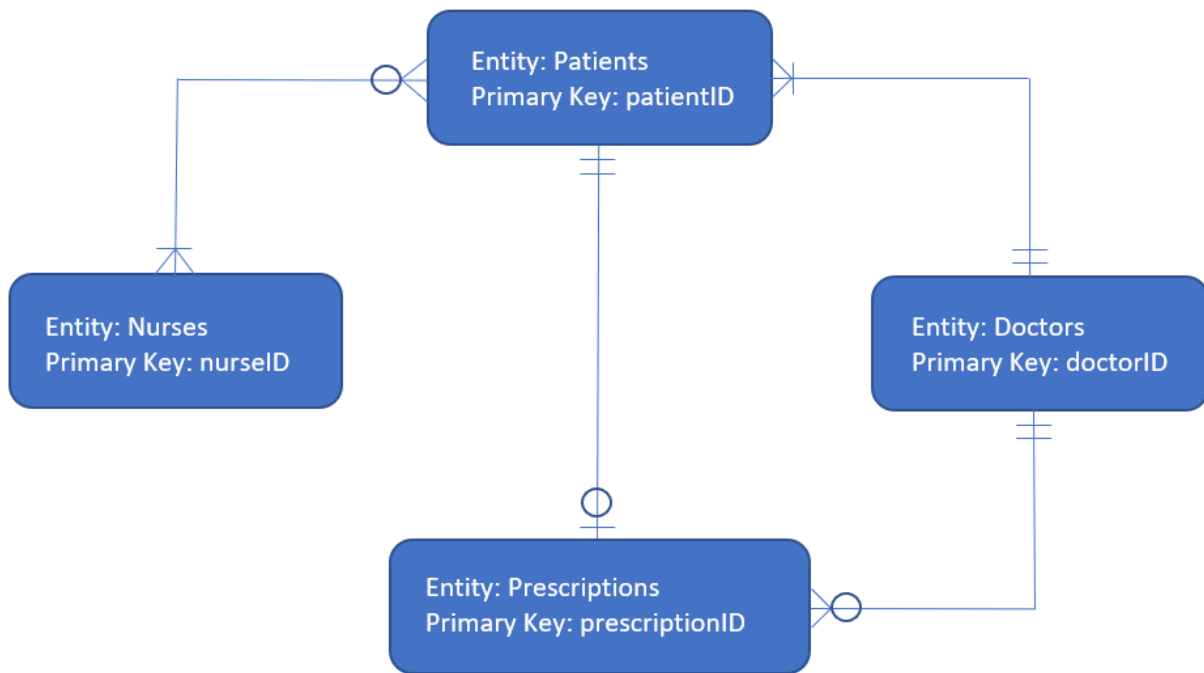- Relationship: M:M between Nurses and Patients, nurseID as a FK inside of nursesPatients

4. **Prescriptions** – records the details of Patient prescriptions
- prescriptionID: int, unique, not NULL, PK
- prescriptionDetails: varchar, not NULL
- prescriptionDate: date, not NULL
- doctorID: int, unique, not NULL, FK
- patientID: int, unique, not NULL, FK
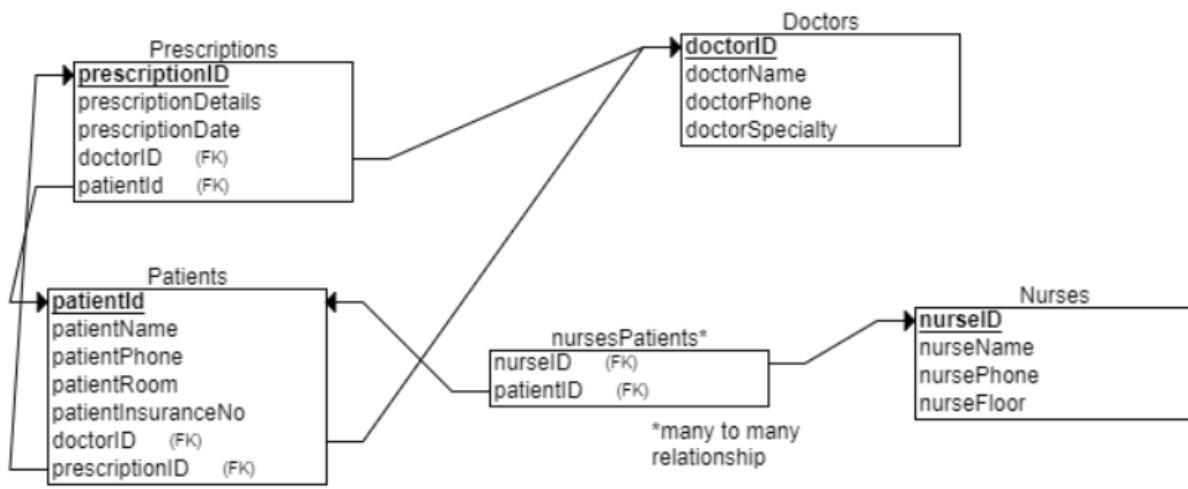- Relationship: 1:1 between Prescriptions and Patients

5. **nursesPatients** – M:M relationship, a conjunction not an entity
- nurseID: int, unique, not NULL, FK
- patientID: int, unique, not NULL, FK

Entity Relationship Diagram:

Entity: Patients
Primary Key: patientID

Entity: Nurses
Primary Key: nurseID

Entity: Doctors
Primary Key: doctorID

Entity: Prescriptions
Primary Key: prescriptionID

Schema:

Prescriptions
prescriptionID
prescriptionDetails
prescriptionDate
doctorID (FK)
patientId (FK)

Doctors
doctorID
doctorName
doctorPhone
doctorSpecialty

Patients
patientId
patientName
patientPhone
patientRoom
patientInsuranceNo
doctorID (FK)
prescriptionID (FK)

nursesPatients*
nurseID (FK)
patientID (FK)

*many to many relationship

Nurses
nurseID
nurseName
nursePhone
nurseFloor

Screen Captures of UI Pages:

Homepage for Hospital Operations Database

DISPLAY and ADD and DELETE and SEARCH Doctors

UPDATE Doctors

Update Doctors

Add Doctor

Name: [Johnny Eyeri update test]

Specialty: [Pediatrician]

Phone Number: [2147345345]

[Submit]

DISPLAY and ADD and DELETE and SEARCH Nurses

Nurses

Add Nurse

Name: [   ]

Floor: [   ]

Phone Number: [   ]

[Submit]

NOTE: (To add a nurse-patient relationship, click the 'Manage..' button in the table below)

View Nurses:

[Search...]

	Nurse ID	Nurse Name	Nurse Floor	Nurse Phone	Nurse's Patients [Manage..]
UPDATE DELETE	8	Laura Holmes	2	2147483647	Tiffany Jackson, Jackie Anderson,
UPDATE DELETE	9	Diana Fickes	1	2147483647	
UPDATE DELETE	11	Leela	2	2147483647	Jake Paul, Andy Wang,

UPDATE Nurses

Update Nurses

┌─ Add Nurse ───
  Name: [Laura Holmes]

  Floor: [2]

  Phone Number: [2147483647]

  [Submit]

  NOTE: (To add a nurse-patient relationship, click the 'Manage..' button in the table below)
└──

DISPLAY and ADD and DELETE and SEARCH Patients

Patients

┌─ Add Patient ───
  Name: [   ]

  Insurance Number: [   ]

  Room: [   ]

  Phone Number: [   ]

  Patient's Doctor [-- select an option -- ▼]

  [Submit]

  NOTE: (To add a nurse-patient relationship, click the 'Manage..' button in the table below)
└──

View Patients:

[Search...]

	Patient ID	Patient Name	Patient Insurance No.	Patient Room	Patient Phone	Patient's Doctor	Patient's Prescription Details	Patient's Nurses [Manage..]
[UPDATE] [DELETE]	6	Tiffany Jackson	998845	215	2147483647	Johnny Eyeri update test	Every third Tuesday, be sure to eat half of apple and chug 12oz root beer	Laura Holmes,
[UPDATE] [DELETE]	7	Jackie Anderson	985622	352	2147483647	Bart Simpson	None	Laura Holmes,

UPDATE Patients

Home   Doctors   Nurses   Patients   Prescriptions

Update Patients

Add Patient

Name: [Tiffany Jackson]

Insurance Number: [998845]

Room: [215]

Phone Number: [2147483647]

Patient's Doctor [Johnny Eyeri update test ∨]

[Submit]

NOTE: (To add a nurse-patient relationship, click the 'Manage..' button in the table below)

DISPLAY and ADD and DELETE and SEARCH Prescriptions

Home   Doctors   Nurses   Patients   Prescriptions

Prescriptions

Add Prescription

Details: [   ]

Date: [mm/dd/yyyy 📅]

Prescription's Patient [-- select an option -- ∨]

Prescription's Doctor [-- select an option -- ∨]

[Submit]

View Prescriptions:

[Search...]

	Prescription ID	Prescription Details	Prescription Date	Prescription's Patient	Prescription's Doctor
UPDATE DELETE	6	See recommended Physical Therapist	2020-11-10	Nate Snyder	Herb Skinson
UPDATE DELETE	7	Tylenol and Pedialyte	2020-12-24	Linda Reynolds	Johnny Eyeri update test
UPDATE DELETE	8	Every third Tuesday, be sure to eat half of apple and chug 12oz root beer	None	Tiffany Jackson	None

UPDATE Prescriptions

Home   Doctors   Nurses   Patients   Prescriptions

Update Prescriptions

Add Prescription

Details: [See recommended Physic]

Date: [11/10/2020 📅]

Prescription's Patient [Nate Snyder ∨]

Prescription's Doctor [Herb Skinson ∨]

[Submit]

DISPLAY and ADD and DELETE and SEARCH nursesPatients (UPDATE not required)

| Home | Doctors | Nurses | Patients | Prescriptions |

Manage nursesPatients

Add Nurse-Patient Relationship
Nurse [-- select an option -- ▼] Patient [-- select an option -- ▼] [Submit]

View Nurses:

	Nurse	Patient
DELETE	(ID: 8) Laura Holmes	(ID: 6) Tiffany Jackson
DELETE	(ID: 8) Laura Holmes	(ID: 7) Jackie Anderson
DELETE	(ID: 11) Leela	(ID: 9) Jake Paul
DELETE	(ID: 11) Leela	(ID: 10) Andy Wang
DELETE	(ID: 18) Jackie Peyton	(ID: 8) Nate Snyder

CS 340 TEAM EVALUATION FORM
DECEMBER 8, 2020

RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

1 = Strongly Disagree   2 = Disagree   3 = Agree   4 = Strongly Agree

GROUP NUMBER		
NAME OF GROUP TEAM MEMBERS:		
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	Very prepared, did research and reading on before meeting. Assignments always completed.
HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	Very responsive, we each have our strengths and we work together well because we update each other via discord or text message all the time.
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY Contributed best academic ability	4	Yes, to our best academic ability!
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? Were both team members both positive and productive?	4	Yes, we use Discord to work on all of our group assignments. Everything was followed.

Are there any suggestions for improvement for your team and what are your goals moving forward?
(Better communication, follow the contract better, modify the initial team contract, more contribution, etc?)?

No suggestions, we've completed the project and fixed all of our issues in past steps, mission complete.