

ITIS-6120 –Applied Databases

Final Project

A Database for an Electronic Medical Record System

submitted by

Shweta Yashdeep Narkhede

At

University Of North Carolina at Charlotte

System Requirements

An Electronic Medical Records (EMRs) or Electronic Health Records (EHRs) is a digital version of a patients paper chart system which collects the patients clinical data and build of a patients care. It contains the medical and treatment histories of patients and makes the information available quickly and securely. EMR system contains patients medical history, medications, diagnosis, treatments, immunizations dates, test results, bills, and visiting information. The aim of this project is to collect and store the patient medical past and current history into database and display through the EMR system using APIs. Displaying the records will help both patient and provider to process further in an efficient and safe way. Patients health care and routine checkup can be easily schedule and maintained with this system.

For a clinic the doctors information is recorded, such as their specification, work time and other significant details. Facility provided by clinic should be recorded, supplies, bill, departments. Patients information is recorded with insurance details and emergency contact. Medicine prescribed by provider and prescription is recorded and bill information is stored. Availability of hospital supplies could be maintained. Briefly, we are storing the information for hospital management system which maintains the records of patients and make it simple to schedule and manage the appointments for patients. New record for patient visiting for the first time is recorded which assign a medical id for that patient in the system. This can be updated in case of any information updates. Visiting patient information can be retrieve from the existing record. The information stored can be search on patient name, id. Specialist can be search by their name and id.

Functional requirements:

- The system will maintain the medical history of patients.
 - system will manage the demographic information about the current and new patients.
 - It will also stores the visit records of the person.
 - It will also manage the test, diagnosis, immunizations, etc. records.
 - It will store patient contact number and email address to inform about the visit and to maintain the track of medical check-up.
- The system will maintain the insurance information of each patients and generate the bill which can be informed to patients on their email.
- The new record should be created for a patient visiting for the first time.
- If it is not the first visit of the patient the previous record for that patient should be editable in case of any change.
- The patients records can be searchable with their medical record ID or by their name to pull up their information.
- The system should maintain the Doctors records with their specialty and available timings with time and day.
- The system will provides drugs to the patients from their tie-up pharmacy's and will maintain the drugs names and any cation if required.
- Overall bill is generated and maintain to the patient's record.

Storyboards:

Clinic management:

A doctors records is maintain in the system with their specialty and available day and time. The status form check in information represents that the patient has checked in or not which helps the receptionist to track how many appointments are their for the day.

New patient:

A patient visiting for the first time in the clinic will fill up the check in form which include the demographic information, insurance information about that person and the appointment time with the doctor. New record is created in the system assigning a medical id for the that patient and check in schedule. A department is set for the check in and patient procedure will be done once it gets available. The examination by the doctor is done in the clinical department assigned as per availability during the check in time. The specialist examine the person and provide prescription or the required tests as per the diagnosis and patients medical records. The provider can recommend the patient to see other specialist if required. The prescriptions are provided by the supplier and then the bill is generated and sent on email of the patient.

Follow up or revisit:

A patient visiting for a daily check up need to fill the check in form again to assure the previous stored personal details has not changed. The provider can see the previous records of the patient and do diagnosis and prescribe the prescription. If the patient comes for the test results the doctor will check the reports and provide further prescriptions and tests.

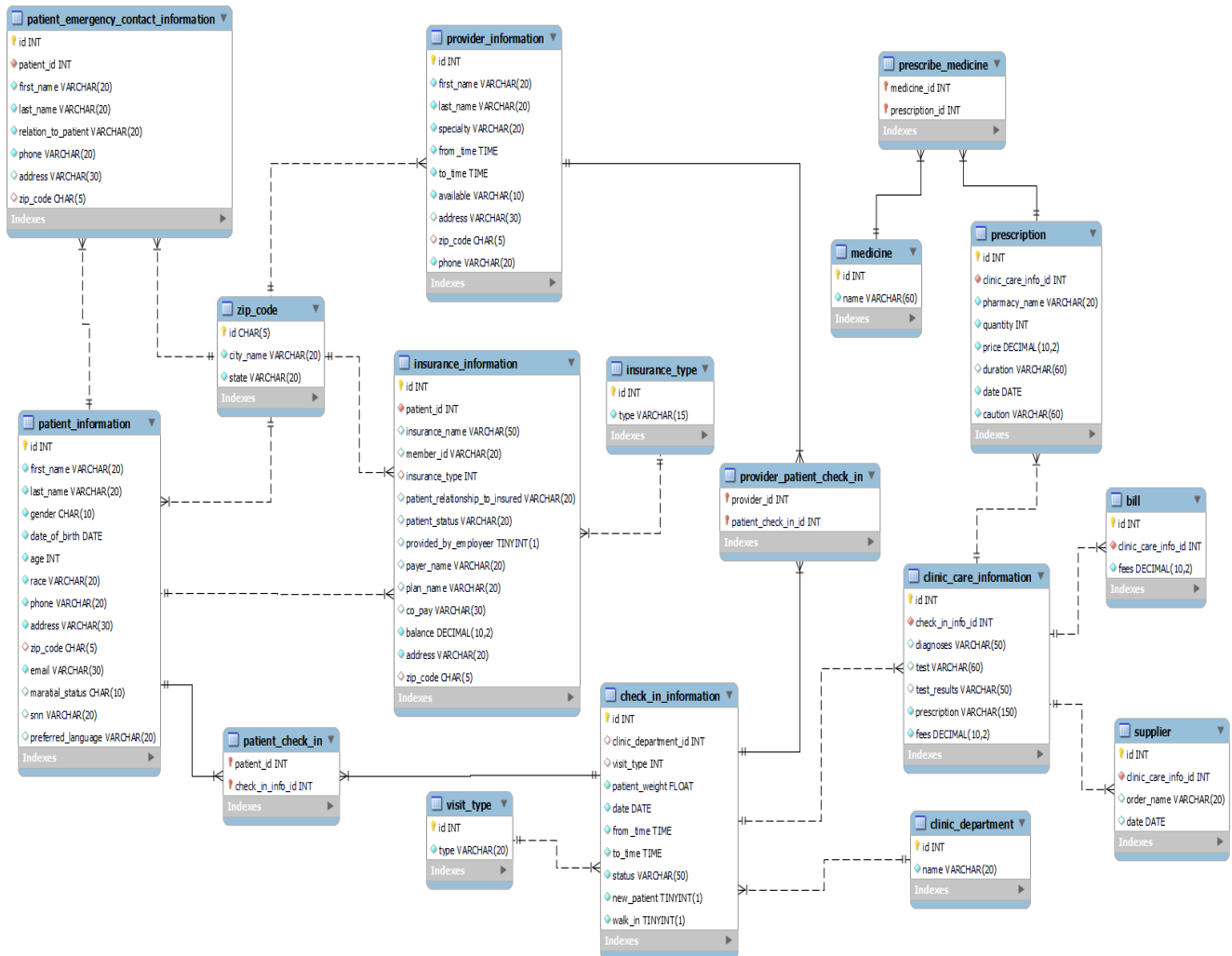
Track the clinic report:

The clinic report can be track on the basis on daily, weekly, monthly and yearly. It can be track with the daily visiting patients. And can be shown statistically through the counts.

Prescriptions for the patients:

Once doctor prescribe the medicine along with course of medicine with provided caution to the patient. The patient will get the prescribed medicine from the pharmacy and the final bill is generated.

ER – Diagram:

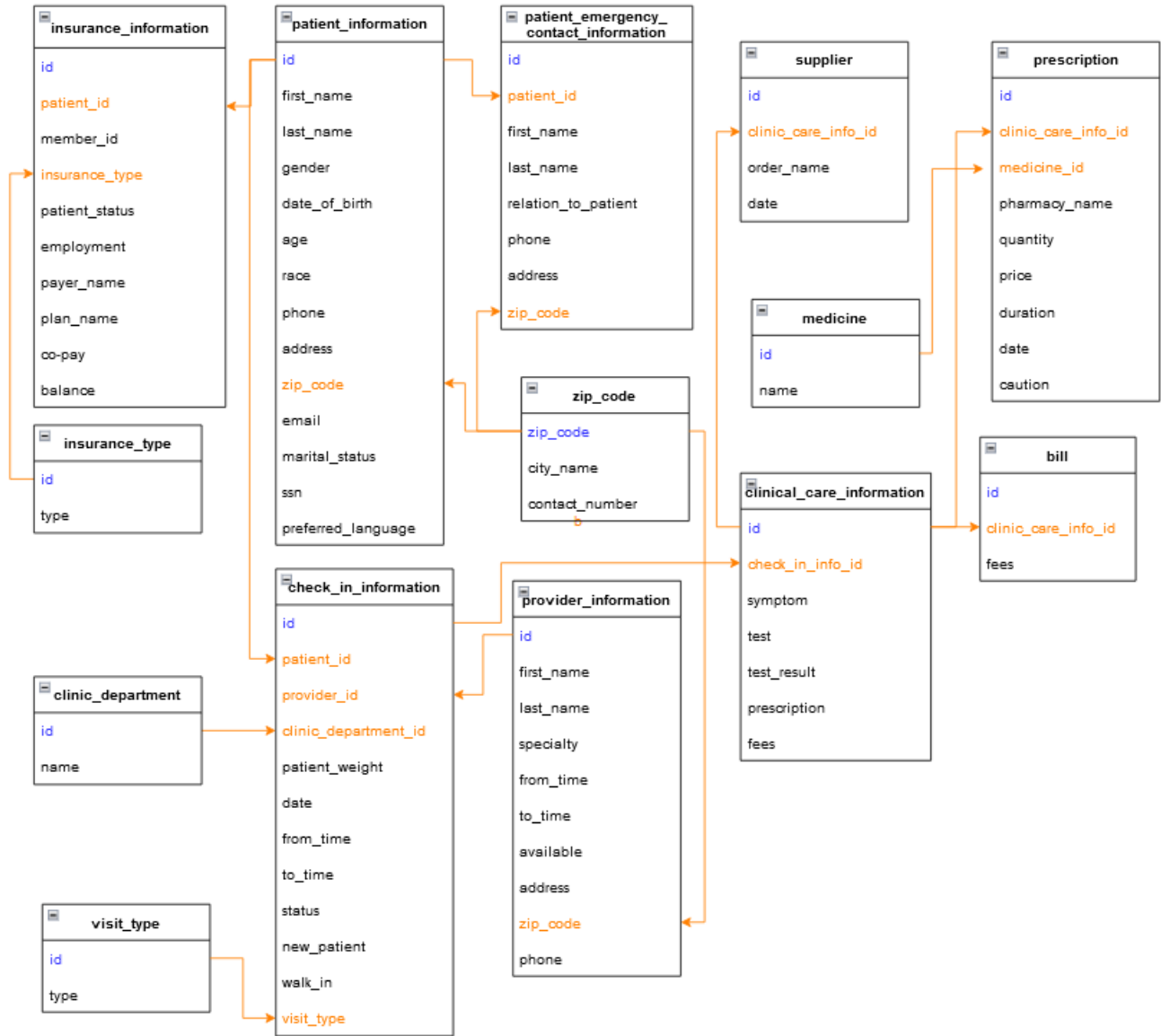


The diagram illustrates the following tables and their attributes:

- insurance_information**: id, patient_id, member_id, insurance_type, patient_status, employment, payer_name, plan_name, co-pay, balance
- insurance_type**: id, type
- patient_information**: id, first_name, last_name, gender, date_of_birth, age, race, phone, address, zip_code, email, marital_status, ssn, preferred_language
- patient_emergency_contact_information**: id, patient_id, first_name, last_name, relation_to_patient, phone, address, zip_code
- zip_code**: zip_code, city_name, contact_number
- clinic_department**: id, name
- visit_type**: id, type
- check_in_information**: id, patient_id, provider_id, clinic_department_id, patient_weight, date, from_time, to_time, status, new_patient, walk_in, visit_type
- provider_information**: id, first_name, last_name, specialty, from_time, to_time, available, address, zip_code, phone
- supplier**: id, clinic_care_info_id, order_name, date
- medicine**: id, name
- clinical_care_information**: id, check_in_info_id, symptom, test, test_result, prescription, fees
- prescription**: id, clinic_care_info_id, medicine_id, pharmacy_name, quantity, price, duration, date, caution
- bill**: id, clinic_care_info_id, fees

Relationships (Foreign Keys):

- insurance_information (patient_id) to patient_information (id)
- insurance_information (insurance_type) to insurance_type (id)
- patient_information (zip_code) to zip_code (zip_code)
- patient_emergency_contact_information (patient_id) to patient_information (id)
- patient_emergency_contact_information (zip_code) to zip_code (zip_code)
- check_in_information (patient_id) to patient_information (id)
- check_in_information (provider_id) to provider_information (id)
- check_in_information (clinic_department_id) to clinic_department (id)
- check_in_information (visit_type) to visit_type (id)
- clinical_care_information (check_in_info_id) to check_in_information (id)
- clinical_care_information (supplier) to supplier (id)
- clinical_care_information (medicine) to medicine (id)
- clinical_care_information (prescription) to prescription (id)
- clinical_care_information (bill) to bill (id)



Glossary

Glossary represents the data dictionary having additional details in text format, which includes the name synonyms and description of an entities described in above data model diagram. Following entities may include more tables and data based on the scope of requirements and their business rules.

Entity Type – **patient_information**

strong entity, super-type

Patient information table store demographic information about the person who is visiting the clinic. The information can be used for future reference and to contact the person.

Attribute Type –

id 1-1(1)

first_name M-1(1)

last_name M-1(1)

Gender M-1(1)

date_of_birth M-1(1)

age M-1(1)

race M-1(1)

phone 1-1(1)

address M-1(1)

zip_code M-1(0)

email 1-1(1)

marital_status (0)

ssn 1-1(0)

preferred_language
M-M(0)

Relationship – patient_information-holds-zip_code, patient_information-has-insurance_information, patient_information-has-emergency_contact_information.

Entity Type – **patient_emergency_contact_information**

weak entity, sub-type

This is a table use to store the information about the person to contact in an emergency about the person admitted or going through the procedures in the clinic.

Attribute Type –

Id 1-1(1)

patient_id 1-1(1)

first_name M-1(1)

last_name M-1(1)

relation_to_patient M-1(1)

phone M-1(1)

address M-1(0)

zip_code (required) M-1(0)

Relationship – patient_emergency_contact_information-holds-patient-information,
patient_emergency_contact_information-holds-zip_code

Entity Type – **insurance_information**

weak entity, sub-type

This table store the information about the persons insurance policies. The details about the insurance company and the plan of insurance and what type of insurance the person have. It also store the employment status of the patient. This table in future can be used to deduct the fees applying the insurance policy.

Attribute Type –

id 1-1(1)

patient_id 1-1(1)

insurance_name M-1(0)

member_id 1-1(0)

insurance_type 1-1(0)

patient_relationship_to_insured 1-1(0)

patient_status 1-1(0)

employment 1-1(0)

payer_name 1-1(0)

plan_name 1-1(0)

Co-pay 1-1(0)

balance 1-1(1)
address M-1(1)
zip_code M-1(0)

Relationship – insurance_information-holds-patient_information,
insurance_information-has-insurance_type.

Entity Type – **check_in_information**

weak entity, sub-type

This table is stores the information about the appointment schedules along with the patient information. It has the status which shows the patient has checked in or had scheduled an appointment.

Attribute Type –

id 1-1(1)
clinic_department_id M-1(1)
patient_weight M-1(1)
date M-1(1)
from_time M-1(1)
to_time M-1(1)
status M-1(1)
new_patient M-1(1)
walk_in M-1(1)
visit_type M-1(1)

Relationship – check_in_information-has-visit_type, check_in_information-has-clinic-department, check_in_information-holds-patient_information, check_in_information-holds-provider-information.

Entity Type – **zip_code**

strong entity, super-type weak entity, sub-type

This table stores the state and cities in that state with its zip code. It will use to record the address of the patient, provider.

Attribute Type –

zip_code 1-1(1)
city_name 1-1(1)
state 1-1(1)

Entity Type – **insurance_type**

strong entity, super-type

This table store the type of insurance patient have.

Attribute Type –

id 1-1(1)
type 1-1(1)

Entity Type – **clinic_department**

strong entity, super-type

This table has the information about the departments in the hospital. For instance, Exam room, checkup room, emergency room, etc.

Attribute Type –

id 1-1(1)
name 1-1(1)

Entity Type – **visit_type**

strong entity, super-type

This table store the information about the appointment type. For instance, follow up, sick visit, blood test, etc.

Attribute Type –

id 1-1(1)
type 1-1(1)

Entity Type – **provider_information**

strong entity, super-type

This table will store information about the provider and their availability with time and day in the clinic.

Attribute Type –

id 1-1(1)
first_name M-1(1)
last_name M-1(1)
specialty M-M(1)
from_time M-1(1)
to_time M-1(1)
available M-M(1)
address M-1(0)
zip_code M-1(0)
phone 1-1(1)

Relationship – provider_information-has-zip_code

Entity Type – **clinic_care_information**

weak entity, sub-type

This table records the information about the appointments scheduled for a patient with the provider and store the prescription given by the doctor. It also stores the information about the tests to be done for the patients and its diagnosis. It also stores visiting fees.

Attribute Type –

id 1-1(1)
check_in_info_id 1-1(1)
diagnoses M-1(0)
test M-1(0)
test_result M-1(0)
prescription M-1(1)
fees M-1(1)

Relationship – clinic_care_information holds check_in_information.

Entity Type – **provider_patient_check_in**

weak entity, sub-type

This table is use to maintain the previous records of the patient and specialist, which will be use in follow up or the future visits of patients.

Attribute Type –

clinic_care_info_id M-M(1)

provider_id 1-1(1)

Entity Type – **medicine**

strong entity, super-type

This table is use to store the records of available medicines in the clinic store.

Attribute Type –

Id 1-1(1)

name 1-1(1)

Entity Type – **supplier**

weak entity, sub-type

This table store the information about the supplies provided to the clinic. It store the information about the beds, emergency rooms, etc.

Attribute Type –

id 1-1(1)

clinic_care_info_id M-M(1)

order_name M-1(1)

date M-1(1)

Relationship – supplier-holds-clinic_care_information.

Entity Type – **prescription**

weak entity, sub-type

This table records the information about the prescription provided by the clinic doctors and pharmacy from where it is. It store the information about the medicine quantity for the patient and duration of the dose.

Attribute Type –

Id 1-1(1)
clinic_care_info_id M-1(1)
pharmacy_name M-M(1)
quantity M-M(1)
price M-M(1)
duration M-M(0)
date M-1(1)
caution M-1(1)

Relationship – prescription-has-medicine, prescription-holds-clinic_care_information.

Entity Type – **bill**

weak entity, sub-type

This table store the total amount for the medical of the patient for their visit to the clinic

Attribute Type –

id 1-1(1)
clinic_care_info_id 1-1(1)
Fees 1-1(1)

Relationship – bill-holds-clinic_care_information.

Synonyms: person – patient, clinic – hospital, procedure – diagnosis – treatment,
check_in – visiting – appointment, exam – test, specialist – provide – doctor, specialty –
specialization.

patient_information has-a patient_emergency_contact_information
patient_information has-a insurance_information
insurance_information has-a insurance_type
check_in_information has-a clinic_care_information
clinic_care_information has-a prescription and supplier\
prescription has-a medicines

Independent tables:

zip_code

visit_type

insurance_type

medicine

clinic_department

insurance_type

Changes to the model

Added following tables

Entity Type – **patient_check_in**

To maintain the patient and check in information

Attribute Type

patient_id 1-1(1)

check_in_info_id 1-1(1)

Entity Type – **provider_patient_check_in**

To maintain the provider and patient check in information

Attribute Type

provider_id 1-1(1)

patient_check_in_id 1-1(1)

Entity Type – **prescribe_medicine**

To maintain the prescription and medicine records

Attribute Type

medicine_id 1-1(1)

prescription_id 1-1(1)

Functional dependencies and BCNF normalization

Boyce–Codd Normal Form (BCNF) is based on functional dependencies that take into account all candidate keys in a relation; however, BCNF also has additional constraints compared with the general definition of 3NF.

In this model new tables are designed to make the database tables in BCNF.

Functional dependence from Super table is removed and decomposed into different tables.

From table *patient_information* and *check_in_information* the functional dependent attribute *patient_id* and *check_in_information* is removed and placed in the third table *patient_check_in*.

From table *provider_information* and *check_in_information* the functional dependent attribute *provider_id* and *patient_check_in_information* is removed and placed in the third table *provider_patient_check_in*.

From table *medicine* and *prescription* the functional dependent attribute *medicine_id* and *prescription_id* is removed and placed in the third table *prescribe_medicine*.

Assessment of the final design

EMR system table designed with test data values.

zip_code

	id	city_name	state
▶	73301	Austin	Texas
	73344	Austin	Texas
	75001	Addison	Texas
	75006	Carrollton	Texas
	75080	Richardson	Texas
	76301	Wichita Falls	Texas
	78833	Camp Wood	Texas

visit_type

	id	type
▶	1	Follow up
	2	Sick visit
	3	Check up
	4	Well child check
	5	Physical exam
	6	Blood test
	7	Immunization shot
	8	Other

insurance_type

	id	type
▶	1	Medicare
	2	Medicaid
	3	Tri care
	4	Champva
	5	Feca
	6	Other

medicine

	id	name
▶	1	Renzo's Picky Eater Multi with Iron daily
	2	Thorne Kids Multi+
	3	Delta D3
	4	Cholecalciferol
	5	Flu Shot
	6	Atovaquone
	7	Tylenol
	8	DTaP Shot

clinic_department

	id	name
▶	1	Exam room
	2	Check up room no. 1
	3	Check up room no. 2
	4	Check up room no. 3
	5	Check up room no. 4
	6	Emergency room
	7	Other

patient_information

	id	first_name	last_name	gender	date_of_birth	age	race	phone	address	zip_code	email	marital_status	snn	preferred_language
▶	1	Jack	Peterson	male	2015-07-02	6	america	801 234 9867	234 hunts ln	73301	jackpeterson@gmail.com	0		English
	2	Emma	Watson	female	2010-03-08	12	america	401 675 4675	112 star ln	73344	emmawatson@yahoo.com	0		English
	3	John	Smith	male	2019-10-05	3	america	771 234 4567	345 creek ln	75001	johnsmith@hotmail.com	0		
	4	Dayana	Jones	female	2005-04-20	16	america	601 834 4867	789 little stone ln	76301	dayanajones@yahoo.com	0		
	5	David	Wilson	male	1990-11-17	32	america	301 234 0860	121 stream ln	78833	davidwilson@gmail.com	1		English

insurance_information

	id	patient_id	insurance_name	member_id	insurance_type	patient_relationship_to_insured	patient_status	provided_by_employee	payer_name	plan_name	co_pay	balance	address	zip_code
▶	1	1	United Health	123 456 697	4	Son	not working	1		NULL		400.00	12 South Ln	78833
	2	2	CIGNA	567 989 690	3	Daughter	not working	1		NULL		568.78	43 East Ln	75006
	3	3	Metropolitan	245 780 261	2	Son	not working	1		NULL		100.50	78 North Ln	75080
	4	4	Point32Health	789 346 900	3	Self	working	1		NULL		670.00	99 Wood walk Ln	73301
	5	5	Metropolitan	879 258 878	1	Self	working	0		NULL		250.78	55 Stream Ln	75080

patient_emergency_contact_information

	id	patient_id	first_name	last_name	relation_to_patient	phone	address	zip_code
▶	1	1	Lesly	Peterson	Mother	801 234 9867	234 hunts ln	73301
	2	2	Bob	Watson	Father	401 675 4676	112 star ln	73344
	3	3	Henly	Smith	Mother	771 234 4567	345 creek ln	75001
	4	4	Emily	Jones	Mother	601 834 6767	789 little stone ln	76301
	5	5	Marry	Wilson	Wife	301 234 1243	121 stream ln	78833

provider_information

	id	first_name	last_name	specialty	from_time	to_time	available	address	zip_code	phone
▶	1	Danial	Hall	Pediatric	08:30:00	17:00:00	Mon-Fri	123 Tom Hl	78833	342 544 6665
	2	Kevin	Lee	general practitioner	08:30:00	17:00:00	Mon-Fri	233 Back St	75080	567 134 9800
	3	Matthew	Scott	Family Medicine	08:30:00	17:00:00	Mon-Fri	677 Park St	76301	324 121 7875
	4	Sandra	Clark	Ophthalmology	08:30:00	17:00:00	Mon-Fri	567 River Rd	75006	213 633 3212
	5	Amanda	White	gynecology	08:30:00	17:00:00	Mon-Fri	678 Peak Ln	73344	785 533 2334

check_in_information

	id	clinic_department_id	visit_type	patient_weight	date	from_time	to_time	status	new_patient	walk_in
▶	1	3	4	20	2022-02-23	13:30:00	15:00:00	Checked In	1	1
	2	2	1	30	2022-03-13	09:30:00	11:00:00	Checked In	1	1
	3	1	7	10	2022-04-08	10:30:00	11:00:00	Checked In	1	1
	4	4	2	38	2022-05-06	09:00:00	11:00:00	Checked In	1	1
	5	5	5	60	2022-06-07	14:30:00	16:00:00	Checked In	1	1
	6	1	4	21	2022-09-12	13:30:00	15:00:00	Checked In	0	1
	7	1	7	11	2022-10-08	10:30:00	11:00:00	Checked In	0	1
	8	3	2	40	2022-11-21	13:30:00	15:00:00	Checked In	0	1
	9	4	2	65	2022-11-25	15:30:00	16:00:00	Checked In	0	1
	10	1	3	12	2023-01-08	10:00:00	11:00:00	Scheduled Appointment	0	1
	11	5	5	60	2022-06-07	14:30:00	16:00:00	Scheduled Appointment	0	1

patient_check_in

	patient_id	check_in_info_id
▶	1	1
	2	2
	3	3
	4	4
	5	5
	1	6
	3	7
	3	8
	5	9
	4	10
	5	11

provider_patient_check_in

	provider_id	patient_check_in_id
▶	1	1
	3	2
	1	3
	3	4
	2	5
	1	6
	1	7
	2	8
	2	9
	2	10
	2	11

clinic_care_information

	id	check_in_info_id	diagnoses	test	test_results	prescription	fees
▶	1	1	Iron deficiency	Blood Screening	All good	Renzo's Picky Eater Multi with Iron daily and Thorne Kids Multi+ each tablet daily till next visit	250.00
	2	2	Vitamin D deficiency	Vitamin D	20-Hydroxyvitamin D	Take Delta D3 and cholecalciferol regularly for a 6 months and recommended sun exposure daily for atleast 1 hour.	300.00
	3	3	NULL	NULL	NULL	Flu Immunization Shot	150.00
	4	4	Cold and Fever	Blood Screening	Malaria positive	Take Atovaquone tablet daily morning and evening for 7 days	350.00
	5	5	Joint pain	Imaging Test...	Diagnosed Arthritis	Tylenol to help ease the pain and exercise daily walking or low-impact aerobic exercise is the best	600.00
	6	6	Iron deficiency	NULL	NULL	Continue the Thorne Kids Multi+ tablet	100.00
	7	7	NULL	NULL	NULL	DTaP Immunization Shot	200.00
	8	8	Covid	Covid-19	positive	Take Tylenol daily morning and evening for 10 days and quarantine for 10 days	150.00
	9	9	Covid	Covid-19	positive	Take Tylenol daily morning and evening for 10 days and quarantine for 10 days	150.00

prescription

	id	clinic_care_info_id	pharmacy_name	quantity	price	duration	date	caution
▶	1	1	CVS	250	35.00	Regularly for 6-7 months	2022-02-23	With Doctor prescription only
	2	1	CVS	250	40.00	Regularly for 6-7 months	2022-02-23	With Doctor prescription only
	3	2	CVS	180	15.00	6 months	2022-03-13	With Doctor prescription only
	4	2	CVS	180	20.00	6 months	2022-03-13	With Doctor prescription only
	5	3	Walgreens	1	80.00	NULL	2022-04-08	With Doctor prescription only
	6	4	CVS	14	10.00	7 days	2022-05-06	With Doctor prescription only
	7	5	Walgreens	30	20.00	1 month	2022-06-07	With Doctor prescription only
	8	6	CVS	356	40.00	Regularly for around 1 year	2022-09-12	With Doctor prescription only
	9	7	Walgreens	1	120.00	NULL	2022-10-08	With Doctor prescription only
	10	8	Walgreens	20	50.00	10 days	2022-11-21	With Doctor prescription only
	11	9	Walgreens	20	50.00	10 days	2022-11-25	With Doctor prescription only

prescribe_medicine

	medicine_id	prescription_id
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	2	8
	8	9
	7	10
	7	11

supplier

	id	clinic_care_info_id	order_name	date
▶	1	1	Blood test Lab	2022-02-23
	2	2	Blood test Lab	2022-03-13
	3	4	Blood test Lab	2022-05-06
	4	5	X-ray Lab	2022-06-07

bill

	id	clinic_care_info_id	fees
▶	1	1	425.00
	2	2	335.00
	3	3	230.00
	4	4	360.00
	5	5	620.00
	6	6	140.00
	7	7	320.00
	8	8	170.00
	9	9	170.00

Following are few of the functionality that the system will perform.

The EMR database system will provide the functionality to get the patients information along with the appointment scheduled and the patient who had checked in.

	first_name	last_name	status	date	from_time	to_time
▶	Dayana	Jones	Scheduled Appointment	2023-01-08	10:00:00	11:00:00
	David	Wilson	Scheduled Appointment	2022-06-07	14:30:00	16:00:00

	first_name	last_name	status	date	from_time	to_time
▶	Jack	Peterson	Checked In	2022-02-23	13:30:00	15:00:00
	Emma	Watson	Checked In	2022-03-13	09:30:00	11:00:00
	John	Smith	Checked In	2022-04-08	10:30:00	11:00:00
	Dayana	Jones	Checked In	2022-05-06	09:00:00	11:00:00
	David	Wilson	Checked In	2022-06-07	14:30:00	16:00:00
	Jack	Peterson	Checked In	2022-09-12	13:30:00	15:00:00
	John	Smith	Checked In	2022-10-08	10:30:00	11:00:00
	John	Smith	Checked In	2022-11-21	13:30:00	15:00:00
	David	Wilson	Checked In	2022-11-25	15:30:00	16:00:00

Functionality to get the total amount of bill generated for the patient(s).

	first_name	last_name	date	diagnoses	fees
▶	Emma	Watson	2022-03-13	Vitamin D deficiency	335.00
	John	Smith	2022-04-08	NULL	230.00
	John	Smith	2022-10-08	NULL	320.00
	John	Smith	2022-11-21	Covid	170.00
	David	Wilson	2022-06-07	Joint pain	620.00
	David	Wilson	2022-11-25	Covid	170.00

Functionality to maintain the regular and new patients

	first_name	last_name	date	diagnoses	fees
▶	Emma	Watson	2022-03-13	Vitamin D deficiency	335.00
	John	Smith	2022-04-08	NULL	230.00
	John	Smith	2022-10-08	NULL	320.00
	John	Smith	2022-11-21	Covid	170.00
	David	Wilson	2022-06-07	Joint pain	620.00
	David	Wilson	2022-11-25	Covid	170.00

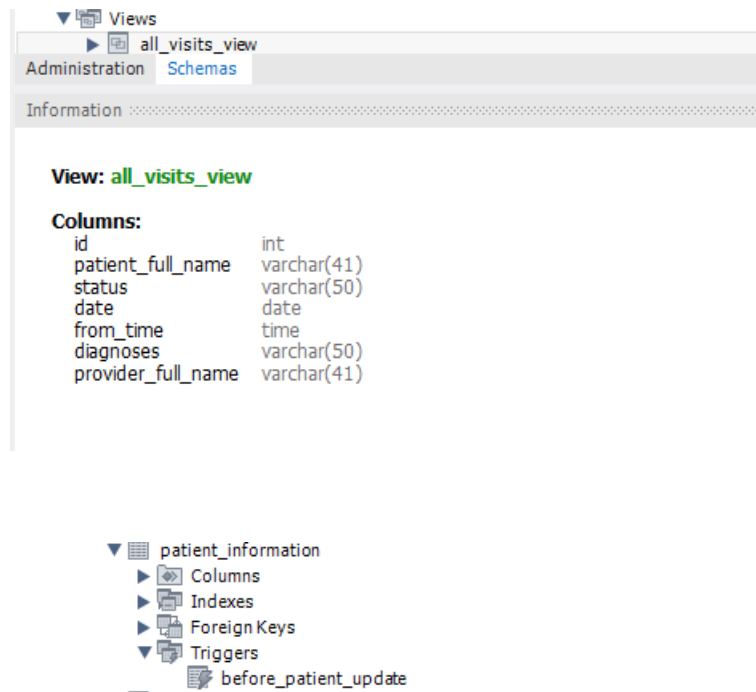
	first_name	last_name	date	diagnoses	fees
▶	Emma	Watson	2022-03-13	Vitamin D deficiency	335.00
	John	Smith	2022-04-08	NULL	230.00
	John	Smith	2022-10-08	NULL	320.00
	John	Smith	2022-11-21	Covid	170.00
	David	Wilson	2022-06-07	Joint pain	620.00
	David	Wilson	2022-11-25	Covid	170.00

Database Design

Stored procedure are added to perform the API functionality. The create, insert tables queries are converted into stored procedures for prior to the execution of the application the sql scripts for the procedures should be run to create the tables and insert the test data into the tables. View `sp_get_all_visits_view` is added to get list of all visits to the clinic including the status of checked in patients.

Trigger `before_patient_update` is add to maintain the logs for patient information updates.





Web Application Implementation:

The database application is further developed with the basic API implementation to perform CRUD operation patient_information table. Which will add Patient data in patient information table, It will also update, retrieve and delete the data from the same table. Stored procedure are developed to perform the API functionality. To implement the EMR application I have used Python Django web framework.

The application will contain the screen as shown in the following figure. It has Register Patients button to put the patient information into the database table patient_information (insurance information is not added now on UI, can be implemented in advance feature). Patient information table is shown through which we can edit or delete the patients records. Also the list of all visits to the clinic is shown along with the status of patients who are checked in. This table can be implemented in advance feature considering more information with editing and deleting functionality.



EMR Systems

[Register Patient](#)

Patients Information

Medical Id	First name	Last name	Gender	Date of birth	Age	Race	Phone	Address	Zip code	Email	Maratial status	SNN	Preferred language	Action
1	Jack	Peterson	male	July 2, 2015	8	america	801 234 9867	234 hunts ln	73301	jackpeterson@gmail.com	1	None	English	Edit Delete
2	Emma	Watson	female	March 8, 2010	12	america	401 675 4675	112 star ln	73344	emmawatson@yahoo.com	0		English	Edit Delete
3	John	Smith	male	Oct 5, 2019	3	america	771 234 4567	345 creek ln	75001	johnsmith@hotmail.com	0			Edit Delete
4	Dayana	Jones	female	April 20, 2005	16	america	601 834 4867	789 little stone ln	76301	dayanajones@yahoo.com	0			Edit Delete
5	David	Wilson	male	Nov. 17, 1990	32	america	301 234 0860	121 stream ln	78833	davidwilson@gmail.com	1		English	Edit Delete

List of all visits to the clinic

Medical Id	Patient name	Status	Date of visit	Time	Diagnoses	Doctor name
1	Jack Peterson	Checked In	Feb. 23, 2022	1:30 p.m.	Iron deficiency	Danial Hall
2	Emma Watson	Checked In	March 13, 2022	9:30 a.m.	Vitamin D deficiency	Matthew Scott
3	John Smith	Checked In	April 8, 2022	10:30 a.m.	None	Danial Hall
4	Dayana Jones	Checked In	May 6, 2022	9 a.m.	Cold and Fever	Matthew Scott
5	David Wilson	Checked In	June 7, 2022	2:30 p.m.	Joint pain	Kevin Lee
1	Jack Peterson	Checked In	Sept. 12, 2022	1:30 p.m.	Iron deficiency	Danial Hall
3	John Smith	Checked In	Oct. 8, 2022	10:30 a.m.	None	Danial Hall
3	John Smith	Checked In	Nov. 21, 2022	1:30 p.m.	Covid	Kevin Lee
5	David Wilson	Checked In	Nov. 25, 2022	3:30 p.m.	Covid	Kevin Lee

Patient information can be added with the following form by clicking on Register Patients.

EMR Systems

First name:

Last name:

Gender:

Date of birth:

Age:

Race:

Phone:

Address:

Zip code:

Email:

Maratial status:

Snn:

Preferred language:

Conclusion

The EMR application is designed to maintain the medical records. The patients appointment scheduling and check in with the Doctors is maintain with time and date. Working on the database design rework has been done as per the functionality requirement and after finding the entities required relationships between the tables are established and cardinality has been defined. The the database has been designed consider the normalization forms. Working on the this project has enhance the hands-on working on real life application development along with the procedure of information gathering discussion on the information and selecting the proper entity writhing the Story-Board and Glossary. Working on this project helps me learn to build end to end application from scratch along with the information gathering, database designing and application deveploment.

The application is implemented in python language with user interface to perform the CURD operation to add patient information. And to show other clinic details such as patient visits data and time and their diagnoses with the provider name. There is future enhancement for this project by implementing advance functionality of calculation of bill with the insurance covered policy and to inform the patient regarding the bills. Advance functionality can be added on the UI to make the application easy to use. Future scope could be adding the functionality of per-notifying the patients for their regular check-up routine. The advance functionality for this system could be the Doctors for children's and adults which will be handle separately during the check-in or appointment booking. In addition, the patients and providers records can be maintain separately based on the specialty of the doctors which will make the system more robust to maintain the medical records.