

Executive Summary

Project Title: Patient Charting Database

Technology Used: Flask Python

Overview: The Evergreen Health Clinic is transitioning from a paper-based chart system to a digital database to efficiently manage the increasing patient base. The system aims to handle critical data such as patient contact details, diagnoses, and treatments, enhancing patient care and streamlining operations.

Database Design Process: The database focused on normalization to ensure data integrity and minimize redundancy. The primary entities include Clinics, Patients, Doctors, Diagnoses, and Procedures, each with structured relationships managed by primary and foreign keys.

Major Changes and Implementation Steps:

1. Server Migration:

- **Change:** Migrated from FLIP server to Classwork server.
- **Reason:** Addressed frequent internal server errors and improved reliability.

2. Schema Adjustments:

- **Change:** Implemented auto-increment for primary keys and enforced non-NULL constraints on critical fields.
- **Reason:** Prevented errors related to NULL values and duplicate keys, enhancing data integrity.

3. Field Mapping and Data Handling:

- **Change:** Corrected field mappings for update operations and included invisible fields for primary keys.
- **Reason:** Ensured accurate data retrieval and submission, reducing errors.

4. User Interface Enhancements:

- **Change:** Added CSS styling, dropdown menus for foreign keys, and mandatory fields for essential attributes.
- **Reason:** Improved the visual appeal and usability of the application, making it more user-friendly.

Conclusion: The project has evolved significantly, with major improvements in server stability, database integrity, and user interface. These changes ensure that Evergreen Health Clinic can efficiently manage its growing patient base, providing high-quality healthcare services and streamlining clinic operations.

a) Project Outline and Database Outline

Project Title: Patient Charting Database: A tool to manage patient information

Overview

The Evergreen Health Clinic, faced with managing the healthcare records of a growing patient base, is transitioning from paper-based chart systems. As patient numbers increase from 100 to 1000, the inefficiencies of traditional methods become evident.

They seek a modernized system capable of efficiently storing and managing patient information to address this. With 10 doctors, 1000 patients, and 1000 procedures, this system will handle crucial data such as contact details, diagnoses, and treatments.

Each patient's medical record will be linked to their name and birthdate, ensuring comprehensive profiles. Utilizing structured schemas and normalization techniques, redundancy will be minimized, and data integrity will be maintained. Relationships between data entities will be managed using primary and foreign keys.

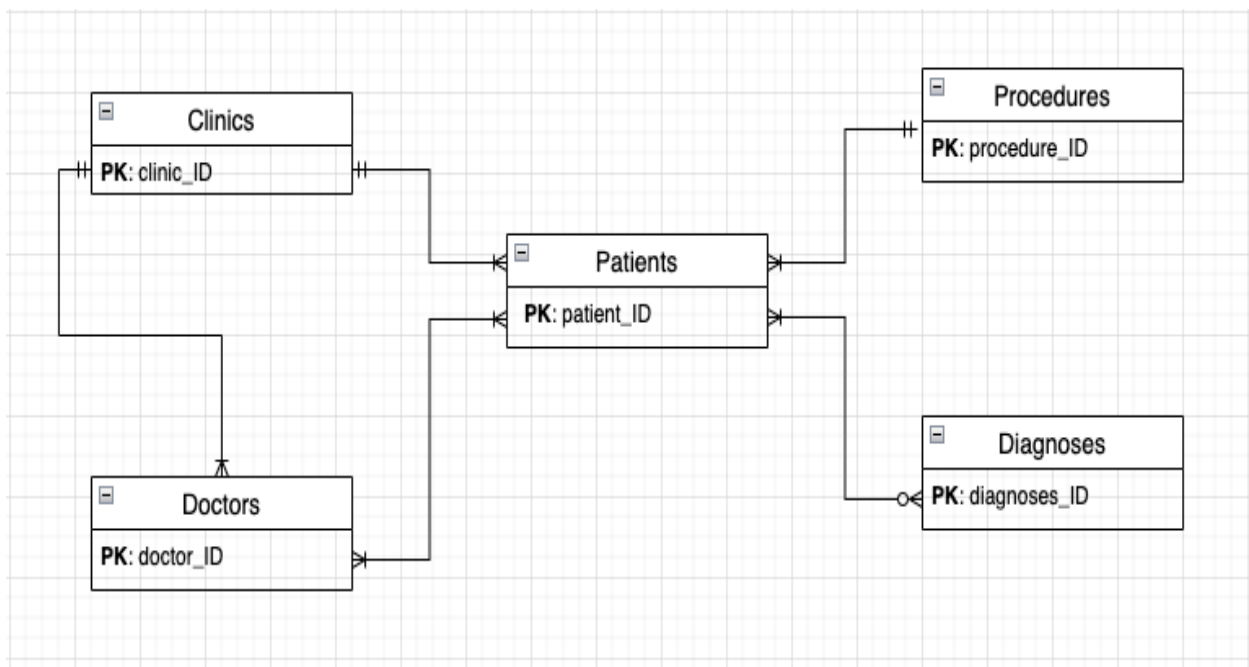
By implementing such a system, the Evergreen Health Clinic aims to enhance patient care and streamline operations, aligning with its commitment to excellence in healthcare provision.

Database Outline

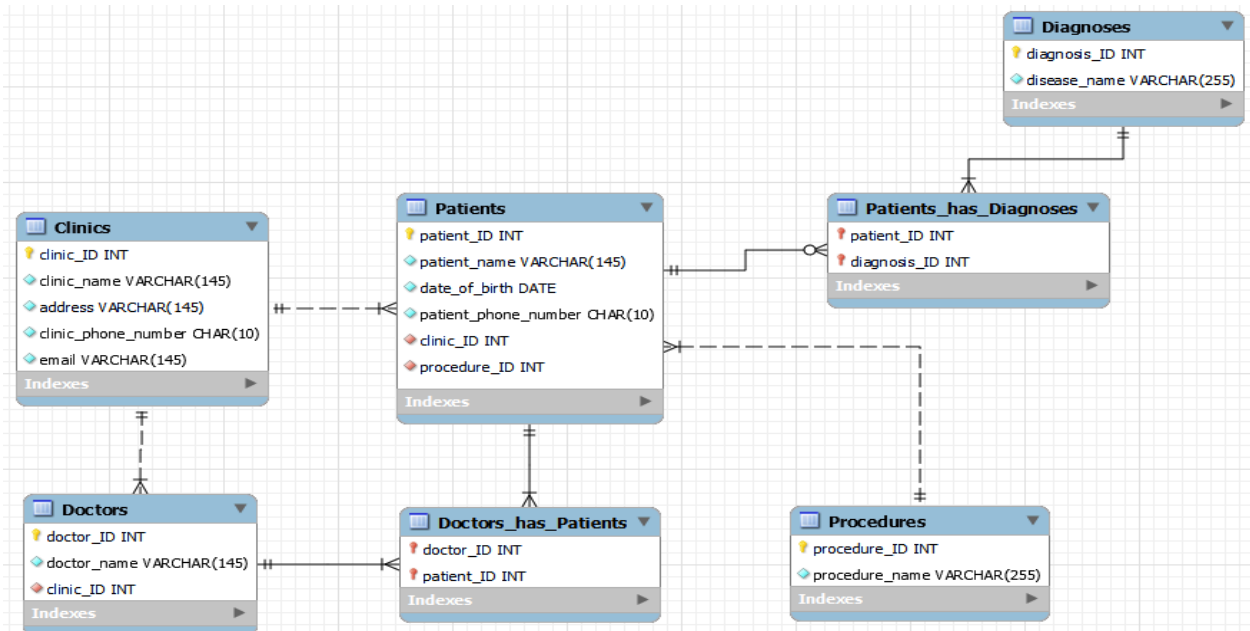
- Clinics: Keeps track of all the clinics which the patients and/or doctors are associated with.
 - clinic_ID: int, auto_increment, not NULL, PK
 - clinic_name: varchar(145), not NULL
 - address: varchar(145), not NULL
 - clinic_phone_number: char(10), not NULL
 - email: varchar(145), not NULL
 - Relationship:
 - a 1:M relationship between Clinics table and Patients table is implemented with clinic_ID as a FK inside of Patients table.
 - a 1:M relationship between Clinics table and Doctors table is implemented with clinic_ID as a FK inside of Doctors table.
- Patients: Records individuals who are patients of the clinic(s) or doctors.
 - patient_ID: int, auto_increment, not NULL, PK
 - patient_name: varchar(145), not NULL
 - date_of_birth: date, not NULL
 - patient_phone_number: CHAR(10), not NULL
 - clinic_ID: FK
 - diagnosis_ID: FK
 - procedure_ID: FK
 - Relationship:
 - a M:M relationship between Patients table and Doctors table is implemented with patient_ID as a FK in intersection table.
- Doctors: Records doctors to keep track of which patients are associated with which doctor.
 - doctor_ID: int, auto_increment, not NULL, PK
 - doctor_name: varchar(145), NOT NULL
 - clinic_ID: FK
 - Relationship:
 - a M:M relationship between Doctors table and Patients table is implemented with doctor_ID as a FK in intersection table.
- Diagnoses: Records the nature of illness that a patient is associated with.
 - diagnosis_ID: int, auto_increment, not NULL, PK
 - disease_name: VARCHAR(255), not NULL
 - Relationship:
 - a M:M relationship between Diagnoses table and Patients table is implemented with diagnosis_ID as a FK inside of Patients table.
 - Note: Crow's foot notation was implemented between Patient and Diagnoses.

- This indicates there can exist a Patient with no diagnosis/es.
- Procedures: Records the treatments that were done for a patient.
 - procedure_ID: int, auto_increment, not NULL, PK
 - procedure_name: VARCHAR(255), not NULL
 - Relationship:
 - a 1:M relationship between Procedures table and Patients table is implemented, with procedure_id as a FK inside of Patients table.

b) Entity-Relationship (ER) Diagram



c) Schema:



Normalization:

- 1NF normalization:** Primary key attributes are included for each of the relevant tables
 - Clinics: **PK** clinic_ID
 - Doctors: **PK** doctor_ID
 - Patients: **PK** patient_ID
 - Procedures: **PK** procedure_ID
 - Diagnoses: **PK** diagnoses_ID
- 2NF normalization:** Tables in our schema hold no partial dependencies
 - Clinics: all attributes are dependent on clinic_ID
 - Patients: all attributes are dependent on patient_ID
 - Doctors: all attributes are dependent on doctor_ID
 - Diagnoses: all attributes are dependent on diagnoses_ID
 - Procedures: all attributes are dependent on procedure_ID
- 3NF:** Tables in our schema are clear of transitive dependencies
 - Clinics: all attributes are dependent on clinic_ID
 - Patients: Removed attribute containing doctor_ID
 - Doctors: Removed attribute containing patient_ID
 - Diagnoses: all attributes are dependent on diagnoses_ID
 - Procedures: all attributes are dependent on procedures_ID

d) Sample Data:

Clinics				
clinic_ID	clinic_name	address	clinic_phone_number	email
1	Evergreen	123 Fake St, Seattle, WA	2064445599	evergreen@aol.com
2	Rainier	467 Sike St, Burien, WA	4259998787	rainier@hotmail.com

Doctors		
doctor_ID	doctor_name	clinic_ID
256	Dr. Lawrence	1
678	Dr. Timothy	1
901	Dr. Larry	2

Doctors_has_Patients	
doctor_ID	patient_ID
256	999
256	1000
678	1004
901	1002

Procedures	
procedure_ID	procedure_name
10	Check Up
11	Vaccinations
12	Physical Therapy

Patients						
patient_ID	patient_name	date_of_birth	patient_phone_number	diagnosis_ID	clinic_ID	procedure_ID
999	Hasselhoff, Lam	1995-11-15	2062434484	40	1	10
1000	Steel, Rick	2003-05-28	4256619092	41	2	11
1002	Ross, Bobby	1965-01-10	3609910909	NULL (0)	2	10
1004	Packer, Michelle	1980-03-11	2067890943	42	1	12

Patients_has_Diagnoses	
patient_ID	diagnoses_ID
999	40
1000	41
1002	NULL (0)
1004	42

Diagnoses	
diagnoses_ID	disease_name
40	Rhinovirus
41	Allergic reaction
42	Diabetes
0	No Diagnosis

e) UI

READ/INSERT/UPDATE/DELETE Clinics Page

Clinics

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		clinic_ID	clinic_name	address	clinic_phone_number	email
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1	Evergreen	123 Fake St, Seattle, WA	2064445599	evergreen@aol.com
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	2	Rainier	467 Sike St, Burien, WA	4259998787	rainier@hotmail.com

Add Clinic

Clinic Name Address Phone Number Email

Update Clinic

ID: 1 Clinic Name Address Phone Number Email

Delete Clinic

Please confirm that you want to delete:

ID: 2 Clinic Name **Rainier**

- Lists all clinics with options to view, add, edit, and delete clinics.

READ/INSERT/UPDATE/DELETE Diagnoses Page

Diagnoses

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		diagnosis_ID	disease_name
Edit	Delete	40	Rhinovirus
Edit	Delete	41	Allergic Reaction
Edit	Delete	42	Diabetes
Edit	Delete	0	No Diagnosis

Add Diagnosis

Diagnosis Name

[Add Diagnosis](#) [Cancel](#)

Update Diagnosis

ID: 42 Diagnosis Name

[Update Diagnosis](#) [Cancel](#)

Delete Diagnosis

Please confirm that you want to delete:

ID: 42 Diagnosis Name **Diabetes**

[Delete Diagnosis](#) [Cancel](#)

- Lists all diagnoses with options to view, add, edit, and delete diagnoses.

READ/INSERT/UPDATE/DELETE Doctors Page

Doctors

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		doctor_ID	doctor_name	clinic_ID
Edit	Delete	256	Dr. Lawrence	Evergreen
Edit	Delete	678	Dr. Timothy	Evergreen
Edit	Delete	901	Dr. Larry	Rainier

Add Doctor

Doctor Name Clinic

[Add Doctor](#) [Cancel](#)

Update Patient

ID: 256 Doctor Name Clinic

[Update Doctor](#) [Cancel](#)

Delete Doctor

Please confirm that you want to delete:

ID: 901 Doctor Name **Dr. Larry**

[Delete Doctor](#) [Cancel](#)

- Lists all doctors with options to view, add, edit, and delete doctors.

READ/INSERT/UPDATE/DELETE Patients Page

Patients

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

<input type="text"/> <input type="button" value="Search"/>								
Filter by Clinic: <input type="button" value="All Clinics"/> Filter by Diagnosis: <input type="button" value="All Diagnoses"/> <input type="button" value="Apply Filters"/>								
		patient_ID	patient_name	date_of_birth	patient_phone_number	diagnosis_ID	clinic_ID	procedure_ID
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	999	Hasselhoff, Lam	1995-11-15	2062434484	Rhinovirus	Evergreen	Check Up
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1000	Steel, Rick	2003-05-28	4256619092	Allergic Reaction	Rainier	Vaccinations
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1002	Ross, Bobby	1965-01-10	3609910909	No Diagnosis	Rainier	Check Up
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1004	Packer, Michelle	1980-03-11	2067890943	Diabetes	Evergreen	Physical Therapy

Add Patient

Patient Name Date of Birth Phone Number Diagnosis Clinic

Update Patient

ID: 1000 Patient Name Steel, Rick Date of Birth 2003-05-28 Phone Number 9496651111 Diagnosis Allergic reaction Clinic Rainier Procedure Vaccinations

Delete Patient

Please confirm that you want to delete:

ID: 1002 Patient Name **Ross, Bobby**

- Lists all patients with options to view, add, edit, and delete patients.
- Diagnosis field is nullable, showing that a patient can be added without specifying a diagnosis.

READ/INSERT/UPDATE/DELETE Patients Per Doctor Page

Patients Per Doctor

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		doctor_ID	patient_ID
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	256	999
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	256	1000
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	678	1004
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	901	1002

Add Doctor/Patient Relationship

Doctor ID Patient ID

Update Doctor/Patient Relationship

Doctor ID: 678

Patient ID

Delete Doctor/Patient Relationship

Please confirm that you want to delete the relationship between:

Doctor ID: 256

Patient ID: 999

- Lists all patients associated with each doctor with options to add new associations, edit existing ones, and delete them.
- Edit and Delete option next to each patient-doctor relationship (M:M), allowing update or deletion of a patient from a doctor's list and vice versa.

READ/INSERT/UPDATE/DELETE Procedures Page

Procedures

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		procedure_ID	procedure_name
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	10	Check Up
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	11	Vaccinations
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	12	Physical Therapy

Add Procedure

Procedure Name

Update Procedure

ID: 12 Procedure Name

Delete Procedure

Please confirm that you want to delete:

ID: 11 Procedure Name **Vaccinations**

- Lists all procedures with options to view, add, edit, and delete procedures.

READ/INSERT/UPDATE/DELETE Diagnoses Per Patient Page

Diagnoses Per Patient

[[Home](#) | [Clinics](#) | [Patients](#) | [Doctors](#) | [Patients Per Doctor](#) | [Diagnoses Per Patient](#) | [Procedures](#) | [Diagnoses](#)]

		patient_ID	diagnoses_ID
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	999	Rhinovirus
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1000	Allergic Reaction
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1002	No Diagnosis
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	1004	Diabetes

Add Patient/Diagnoses Relationship

Patient ID Diagnoses ID

Update Patient/Diagnoses Relationship

Patient ID: 1002

Diagnoses ID

Delete Patient/Diagnoses Relationship

Please confirm that you want to delete the relationship between:

Patient ID: 999

Diagnosis ID: Rhinovirus

- Lists all diagnoses associated with each patient with options to add new associations, edit existing ones, and delete them.
- Edit and Delete option next to each diagnosis-patient relationship (M:M), allowing the update or deletion of a diagnosis from a patient's list and vice versa.