RELATIONAL DATABASE PROJECT

Hospital Management System

Introduction:

The hospital management system is a database designed to help hospital staff manage information related to patients, employees, wards, appointments, and medicines. The system aims to provide a centralized location for storing and managing data, allowing for easy access and retrieval of information. By using this system, hospital staff can efficiently manage and monitor patient care, streamline administrative tasks, and improve overall efficiency.

Mission Statement:

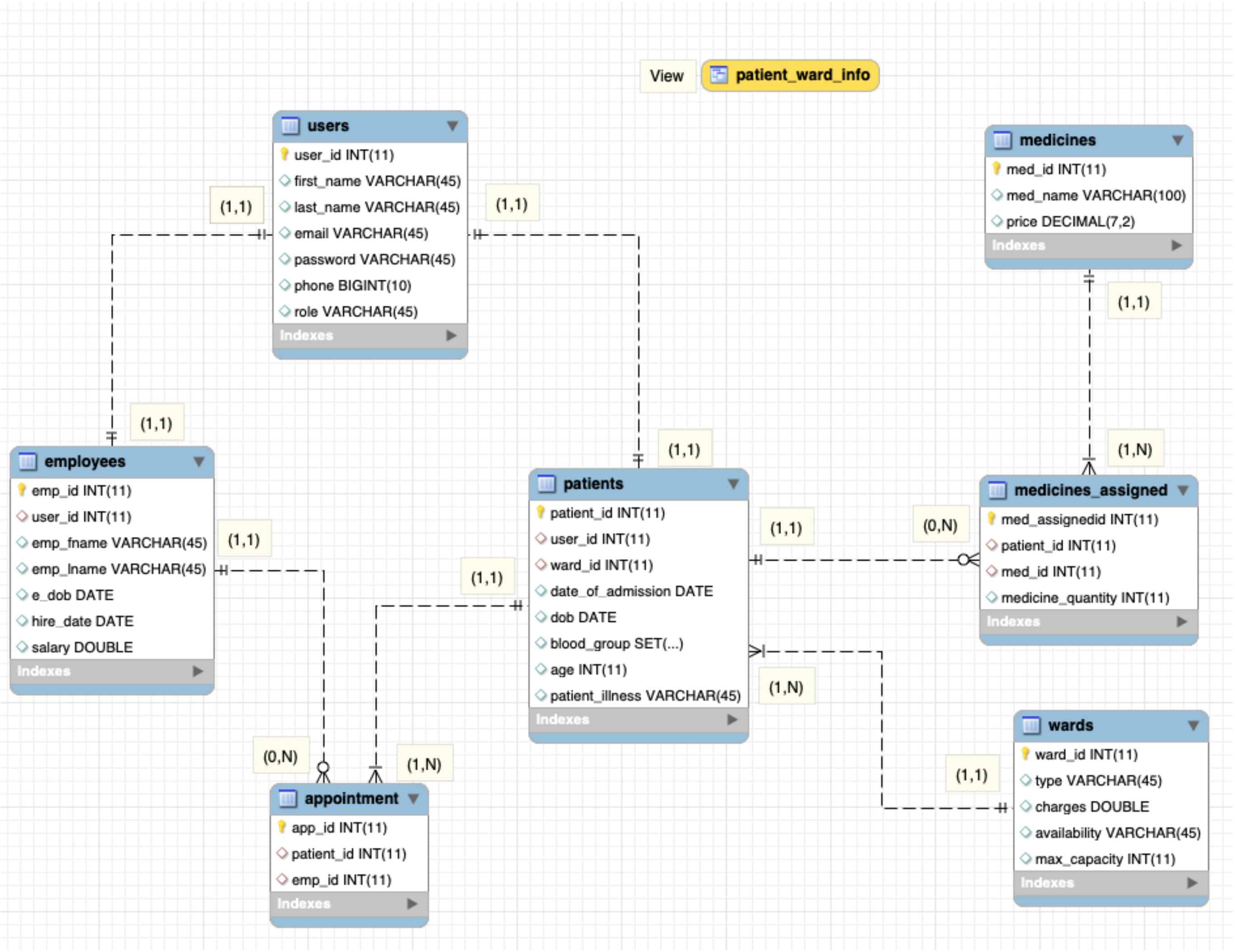
Our mission is to provide a comprehensive and user-friendly hospital management system that streamlines processes, optimizes patient care, and improves staff productivity.

Mission Objectives:

- To store and manage patient information, including their medical history, blood group, and current illness.
- To maintain a database of hospital employees, including their personal and employment information.
- To manage and monitor hospital wards, including their availability and maximum capacity.
- To schedule and manage patient appointments with doctors and other healthcare professionals.
- To keep track of the medicines available at the hospital and the medicines assigned to patients.
- To generate reports and analytics related to patient care, employee productivity, and hospital performance.
- To ensure data security and privacy by restricting access to authorized personnel only.

CONCEPTUAL DESIGN:

- ERD Diagram



View patient_ward_info

users
user_id INT(11)
first_name VARCHAR(45)
last_name VARCHAR(45)
email VARCHAR(45)
password VARCHAR(45)
phone BIGINT(10)
role VARCHAR(45)
Indexes

medicines
med_id INT(11)
med_name VARCHAR(100)
price DECIMAL(7,2)
Indexes

employees
emp_id INT(11)
user_id INT(11)
emp_fname VARCHAR(45)
emp_lname VARCHAR(45)
e_dob DATE
hire_date DATE
salary DOUBLE
Indexes

patients
patient_id INT(11)
user_id INT(11)
ward_id INT(11)
date_of_admission DATE
dob DATE
blood_group SET(...)
age INT(11)
patient_illness VARCHAR(45)
Indexes

medicines_assigned
med_assignedid INT(11)
patient_id INT(11)
med_id INT(11)
medicine_quantity INT(11)
Indexes

wards
ward_id INT(11)
type VARCHAR(45)
charges DOUBLE
availability VARCHAR(45)
max_capacity INT(11)
Indexes

appointment
app_id INT(11)
patient_id INT(11)
emp_id INT(11)
Indexes

- Mandatory/Optional Relationship Table

TABLES	RELATIONSHIP
users to employees	Mandatory
employees to users	Mandatory
users to patients	Mandatory
patients to users	Mandatory
employees to appointment	Optional
appointment to employees	Mandatory
patient to appointment	Mandatory
appointment to patient	Mandatory
patients to medicines_assigned	Optional
medicines_assigned to patients	Mandatory
medicines to medicines_assigned	Mandatory
medicines_assigned to medicines	Mandatory
patients to wards	Mandatory
wards to patients	Mandatory

LOGICAL DESIGN:

- Data Description Tables for each Entity:

1. USERS

Field Name	Data Type	Data Length	Null	Index	Key Type	Default Value	Range of Values	Description
user_id	INT	11	NO	PRIMARY	PRIMARY KEY	NOT NULL	NONE	A unique identifier for each user
first_name	VARCHAR	45	YES	None	None	NULL	NONE	The user's first name
last_name	VARCHAR	45	YES	None	None	NULL	NONE	The user's last name
email	VARCHAR	45	YES	None	None	NULL	NONE	The user's email address
password	VARCHAR	45	YES	None	None	NULL	NONE	The user's password
phone	BIGINT	10	YES	None	None	NULL	NONE	The user's phone number
role	VARCHAR	45	YES	None	None	NULL	NONE	The user's role or position within the system

2. EMPLOYEES

Field Name	Data Type	Data Length	Null	Index	Key Type	Default Value	Range of Values	Description
emp_id	INT	11	NO	PRIMARY	PRIMARY KEY	NOT NULL	NONE	A unique identifier for each employee
user_id	INT	11	YES	user_id_fkk_idx	FOREIGN KEY	NULL	NONE	The ID of the user associated with the employee
emp_fname	VARCHAR	45	YES	None	None	NULL	NONE	The employee's first name
emp_lname	VARCHAR	45	YES	None	None	NULL	NONE	The employee's last name
e_dob	DATE	None	YES	None	None	NULL	NONE	The employee's date of birth
hire_date	DATE	None	YES	None	None	NULL	NONE	The date the employee was hired
salary	DOUBLE	None	YES	None	None	NULL	10000 to 200000	The employee's salary

3. PATIENTS

Field Name	Data Type	Data Length	Null	Index	Key Type	Default	Range of Values	Description
patient_id	INT	11	NO	PRIMARY	PRIMARY	NOT NULL	NONE	Unique ID for the patient
user_id	INT	11	YES	u_id_fk_idx	FOREIGN	NULL	NONE	Foreign key referencing users table
ward_id	INT	11	YES	ward_id_idx	FOREIGN	NULL	NONE	Foreign key referencing wards table
date_of_admission	DATE	None	YES	None	None	NULL	NONE	Date of patient admission to hospital
dob	DATE	None	YES	None	None	NULL	NONE	Date of birth of the patient
blood_group	SET	None	YES	None	None	NULL	A+, A-, B+, B-, O+, O-, AB+, AB-	Blood group of the patient
age	INT	11	YES	None	None	NULL	1 to 125	Age of the patient
patient_illness	VARCHAR	45	YES	None	None	NULL	NONE	Illness or medical condition of the patient

4. WARDS

Field Name	Data Type	Data Length	Null	Index	Key Type	Default Value	Range of Values	Description
ward_id	INT	11	NO	PRIMARY	PRIMARY KEY	NOT NULL	NONE	A unique identifier for each ward
type	VARCHAR	45	YES	None	None	NULL	None	The type of ward (e.g., private, general, etc.)
charges	DOUBLE	None	YES	patient_ward_charges_idx	None	NULL	1500 to 20000	The charges associated with the ward
availability	VARCHAR	45	YES	avail_idx	None	NULL	"Available" or "Unavailable"	The availability status of the ward
max_capacity	INT	11	YES	None	None	NULL	1 to 12	The maximum capacity of the ward

5. MEDICINES

Field Name	Data Type	Data Length	Null Determination	Index	Key Type	Default Value	Range of Value	Description
med_id	INT	11	NOT NULL	PRIMARY	PRIMARY KEY	NONE	NONE	Unique identifier for each medicine
med_name	VARCHAR	100	NULL	None	None	NULL	NONE	Name of the medicine
price	DECIMAL	(7,2)	NULL	None	None	NULL	1 to 1000	Price of the medicine

6. APPOINTMENT

Field Name	Data Type	Data Length	Null Determination	Index	Key Type	Default Value	Range of Value	Description
app_id	INT	11	Not Null	PRIMARY	PRIMARY	None	NONE	Unique ID for an appointment
patient_id	INT	11	Nullable	pat_id_fk_idx	FOREIGN	None	NONE	ID of the patient who has the appointment
emp_id	INT	11	Nullable	emp_id fk_idx	FOREIGN	None	NONE	ID of the employee who has the appointment

7. MEDICINES_ASSIGNED

Field Name	Data Type	Data Length	Null Determination	Index	Key Type	Default Value	Range of Value	Description
med_assigned id	INT	11	NOT NULL	PRIMARY	PRIMARY KEY	NULL	NONE	Unique identifier for the medicine assignment
patient_id	INT	11	NULL	pat_id_fk_idx	FOREIGN KEY (FK)	NULL	NONE	The ID of the patient for whom the medicine is assigned
med_id	INT	11	NULL	med_id_fk_idx	FOREIGN KEY (FK)	NULL	NONE	The ID of the medicine being assigned
medicine_quantity	INT	11	NULL	NONE	None	NULL	1 to 10	The quantity of the medicine assigned to the patient

- Business Rules for each entity

1. USERS:

i. Attribute-oriented business rule:

- The email attribute should be unique for each user. This ensures that no two users have the same email address in the system.

ii. Relationship-oriented business rule:

- Each user can have only one role assigned to them. This ensures that there is a clear and unambiguous understanding of each user's role in the system.

iii. Attribute-oriented business rule:

- The phone attribute should only contain valid phone numbers with 10 digits. This ensures that the phone numbers stored in the system are valid and can be used for communication purposes.

2. EMPLOYEES:

i. Attribute-oriented business rule:

- The salary attribute should not be negative. This ensures that the salary information stored in the system is accurate and consistent with the real-world scenario.

ii. Relationship-oriented business rule:

- Each employee must be associated with one user. This ensures that there is a clear understanding of the relationship between an employee and a user in the system.

iii. Attribute-oriented business rule:

- The hire_date attribute should not be in the future. This ensures that the hire date information stored in the system is accurate and consistent with the real-world scenario.

3. WARDS:

i. Attribute-oriented business rule:

- The max_capacity attribute should be a positive integer. This ensures that the maximum capacity of a ward is accurately represented and cannot be negative.

ii. Relationship-oriented business rule:

- A ward can have many patients, but each patient can only be assigned to one ward. This ensures that there is a clear understanding of the relationship between a ward and a patient in the system.

iii. Attribute-oriented business rule:

- The charges attribute should not be zero or negative. This ensures that the charges information stored in the system is accurate and consistent with the real-world scenario.

4. PATIENTS

i. Attribute-oriented business rule:

- The date_of_admission attribute should not be in the future. This ensures that the date of admission information stored in the system is accurate and consistent with the real-world scenario.

ii. Relationship-oriented business rule:

- Each patient must be associated with one user. This ensures that there is a clear understanding of the relationship between a patient and a user in the system.

iii. Attribute-oriented business rule:

- The age attribute should be greater than or equal to zero. This ensures that the age information stored in the system is accurate and consistent with the real-world scenario.

5. APPOINTMENT:

Attribute-oriented rules:

1. app_id must be unique and cannot be null.
2. patient_id and emp_id can be null, but they must be valid foreign keys referring to patients.patient_id and employees.emp_id respectively, if not null.

Relationship-oriented rules:

1. An appointment must be associated with exactly one patient and one employee.
2. An employee can have zero or more appointments.

6. MEDICINES:

i. Attribute-oriented: The med_name attribute must not be null. Every row in the table must have a value for med_name.

ii. Relationship-oriented: The medicines table may have a many-to-many relationship with the patients table through a Linking table ‘medicines_assigned’.

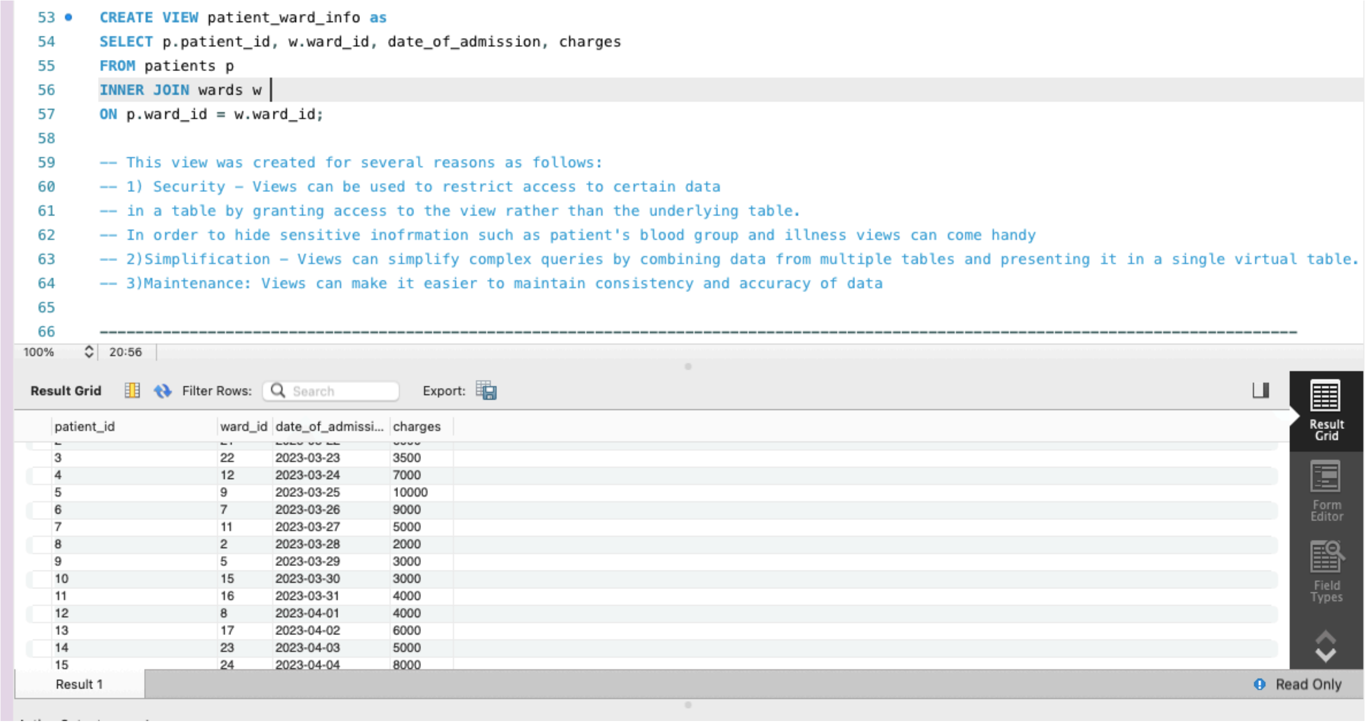
iii. Attribute-oriented: The price attribute should be a decimal with a maximum of 7 digits, and 2 decimal places. This ensures that the price of a medicine is stored accurately and consistently in the database.

7. MEDICINES_ASSIGNED:

Attribute-oriented:

1. The med_assignedid attribute must be unique for each record.
2. The medicine_quantity attribute must be a positive integer.
3. The med_id attribute must be a foreign key that references the medicines table.

○ VIEWS:



```sql
CREATE VIEW patient_ward_info as
SELECT p.patient_id, w.ward_id, date_of_admission, charges
FROM patients p
INNER JOIN wards w 
ON p.ward_id = w.ward_id;

-- This view was created for several reasons as follows:
-- 1) Security - Views can be used to restrict access to certain data
-- in a table by granting access to the view rather than the underlying table.
-- In order to hide sensitive inofrmation such as patient's blood group and illness views can come handy
-- 2)Simplification - Views can simplify complex queries by combining data from multiple tables and presenting it in a single virtual table.
-- 3)Maintenance: Views can make it easier to maintain consistency and accuracy of data

-----------------------------------------------------------------------------------------------------------------------------------
```

patient_id	ward_id	date_of_admissi...	charges
2	21	2023-03-22	[illegible]
3	22	2023-03-23	3500
4	12	2023-03-24	7000
5	9	2023-03-25	10000
6	7	2023-03-26	9000
7	11	2023-03-27	5000
8	2	2023-03-28	2000
9	5	2023-03-29	3000
10	15	2023-03-30	3000
11	16	2023-03-31	4000
12	8	2023-04-01	4000
13	17	2023-04-02	6000
14	23	2023-04-03	5000
15	24	2023-04-04	8000

This view was created for several reasons as follows:
1) Security - Views can be used to restrict access to certain data in a table by granting access to the view rather than the underlying table. In order to hide sensitive information such as patient's blood group and illness views can come handy.
2)Simplification - Views can simplify complex queries by combining data from multiple tables and presenting it in a single virtual table.
3)Maintenance: Views can make it easier to maintain consistency and accuracy of data