Project Report

Object Oriented Database:

Extended Healthcare Management System

Problem Statement:

"Developing a unified and efficient database system to manage patient records, appointments, staff information, and medical resources, addressing current challenges in data fragmentation, scheduling inefficiencies, and limited access to comprehensive patient information in healthcare settings."

Database Description:

In the realm of healthcare management systems, the application of advanced database concepts plays a pivotal role in addressing complex data management needs. These concepts include types, arrays, multisets, inheritance, and object-based database design, each contributing uniquely to the system's efficiency and robustness.

Firstly, the concept of 'types' in object-relational databases is fundamental. These user-defined types act as templates for creating structured data objects, encapsulating both data and behaviors akin to classes in object-oriented programming. They are versatile in defining the structure of various entities, such as a `patient_type`, which might encompass attributes like name, age, and medical history. This encapsulation facilitates the management of complex data structures within the healthcare system.

Complementing this, arrays and multisets address the storage and handling of multiple data items within a single field. Arrays are ordered collections of elements of the same type, ideal for storing lists like phone numbers or medication names, where the order is significant. In contrast, multisets, akin to arrays but lacking ordering and uniqueness constraints, are suitable for scenarios where duplication is permissible and order is non-essential, such as logging various symptoms reported by patients.

Moreover, the concept of inheritance in database systems, borrowed from object-oriented programming, allows for the creation of new types based on existing ones, thereby enabling hierarchical data structures. This feature is utilized in creating generalized types like `person_type`, which can then be specialized into types such as `patient_type` or `doctor_type`, each inheriting common attributes while adding specific ones.

Lastly, object-based database design is instrumental in mirroring real-world scenarios within the database, creating structured entities like patients, doctors, and appointments as objects with their own properties and methods. This design approach not only enables the representation of complex relationships but also supports advanced database operations like encapsulation, polymorphism, and inheritance. This results in a system that can handle complex queries, such as retrieving a patient’s complete medical history or managing a doctor's schedule, thereby enhancing the healthcare system’s functionality.

User Defined Types:

1. ***Patient Type***

```
CREATE TYPE new_patient_type AS OBJECT (
    patient_id VARCHAR2(20),
    name VARCHAR2(100),
    age NUMBER,
    gender VARCHAR2(10),
    contact_numbers phone_numbers_type,
    address address_type
);
```

2. ***Doctor Type***

```
CREATE TYPE doctor_type AS OBJECT (
    doctor_id VARCHAR2(20),
    name VARCHAR2(100),
    specialization VARCHAR2(50),
    contact_numbers phone_numbers_type,
    address address_type
);
```

3. ***Phone No Type***

```
CREATE OR REPLACE TYPE
phone_numbers_type AS VARRAY(3) OF VARCHAR2(15);
```

4. ***Address Type***

```
CREATE OR REPLACE TYPE address_type AS OBJECT (
    street VARCHAR2(100),
    city VARCHAR2(50),
    state VARCHAR2(50),
    zip VARCHAR2(10)
);
```

5. ***Staff Type***

```
CREATE TYPE staff_type AS OBJECT (
   staff_id VARCHAR2(20),
   name VARCHAR2(100),
   position VARCHAR2(50),
   contact_numbers phone_numbers_type,
   address address_type
 );
```

6. *Ward Type*

```
CREATE TYPE ward_type AS OBJECT (
    ward_id VARCHAR2(20),
    ward_name VARCHAR2(100),
    capacity NUMBER,
    department_id VARCHAR2(20)
);
```

7. *Insurance Type*

```
CREATE TYPE insurance_type AS OBJECT (
    insurance_id VARCHAR2(20),
    patient_id VARCHAR2(20),
    provider_name VARCHAR2(100),
    coverage_details VARCHAR2(255)
);
```

8. *Treatment record Type*

```
CREATE OR REPLACE TYPE treatment_record_type AS OBJECT (
    treatment_date DATE,
    treatment_description VARCHAR2(255)
);
```

9. *Medical record Type(Nested Table)*

```
CREATE TYPE medical_record_type AS TABLE OF treatment_record_type;
```

10. *Medicine List Type*

```
CREATE TYPE medicine_list_type AS VARRAY(10) OF VARCHAR2(100);
```

11. Equipment Type

```
CREATE TYPE equipment_type AS OBJECT (
    equipment_id VARCHAR2(20),
    name VARCHAR2(100),
    type VARCHAR2(50),
    department_id VARCHAR2(20)
);
```

Tables:

a. Core Tabeles

1. Patients Table

```
CREATE TABLE Patients (
    patient_id VARCHAR2(20) PRIMARY KEY,
    patient_info patient_type,
    medical_records medical_record_type
) NESTED TABLE medical_records STORE AS medical_records_nt;
```

2. Doctors Table

```
CREATE TABLE Doctors (
    doctor_id VARCHAR2(20) PRIMARY KEY,
    doctor_info doctor_type
);
```

3. Staff Table

```
CREATE TABLE Staff (
    staff_id VARCHAR2(20) PRIMARY KEY,
    staff_info staff_type
);
```

b. Additional Tabeles

1. Medical Equipment Table

```
CREATE TABLE Medical_Equipment (
    equipment_id VARCHAR2(20) PRIMARY KEY,
    equipment_info equipment_type,
    CONSTRAINT fk_equipment_department FOREIGN KEY
(equipment_info.department_id) REFERENCES Departments(department_id)
);
```

2. Wards Table

```
CREATE TABLE Wards (
    ward_id VARCHAR2(20) PRIMARY KEY,
    ward_info ward_type,
    CONSTRAINT fk_ward_department FOREIGN KEY (ward_info.department_id)
REFERENCES Departments(department_id)
);
```

3. Pharmacy Table

```
CREATE TABLE Pharmacy (
    medicine_id VARCHAR2(20) PRIMARY KEY,
    medicine_list medicine_list_type
);
```

4. Insurance Table

```
CREATE TABLE Insurance (
    insurance_id VARCHAR2(20) PRIMARY KEY,
    insurance_info insurance_type,
    CONSTRAINT fk_insurance_patient FOREIGN KEY (insurance_info.patient_id)
REFERENCES Patients(patient_id)
);
```

c. Relationship Tabeles

1. Appointments Table

```
CREATE TABLE Appointments (
    appointment_id VARCHAR2(20) PRIMARY KEY,
    patient_id VARCHAR2(20),
    doctor_id VARCHAR2(20),
    appointment_date DATE,
    reason VARCHAR2(255),
    CONSTRAINT fk_patient_appointment FOREIGN KEY (patient_id) REFERENCES
Patients(patient_id),
    CONSTRAINT fk_doctor_appointment FOREIGN KEY (doctor_id) REFERENCES
Doctors(doctor_id)
);
```

2. Departments Table

```
CREATE TABLE Departments (
    department_id VARCHAR2(20) PRIMARY KEY,
    department_name VARCHAR2(50),
    head_doctor_id VARCHAR2(20),
    CONSTRAINT fk_head_doctor FOREIGN KEY (head_doctor_id) REFERENCES
Doctors(doctor_id)
);
```

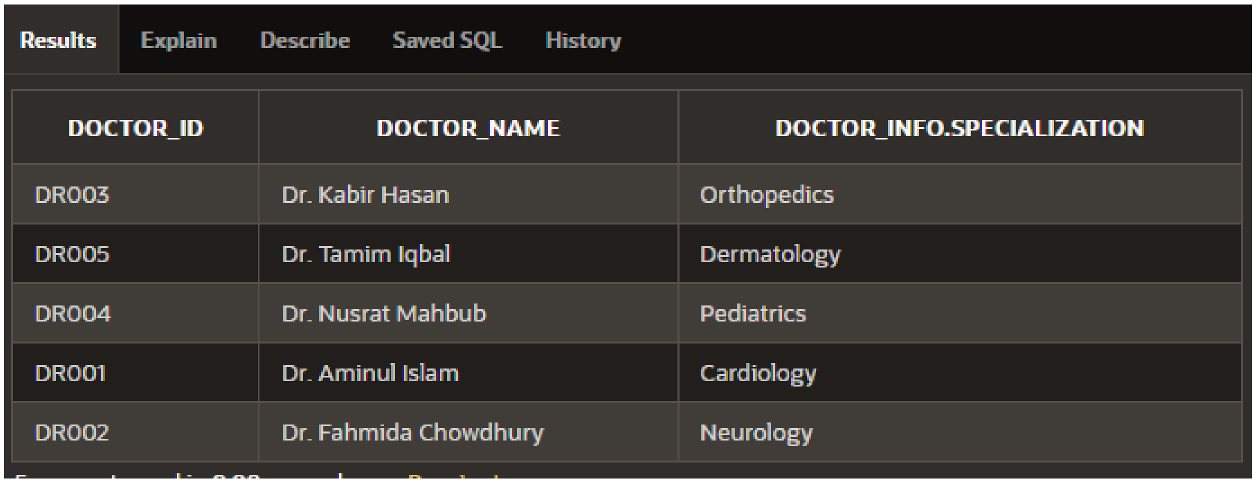
d. Insertion

```
INSERT INTO Doctors (doctor_id, doctor_info) VALUES
('DR001', doctor_type('DR001', 'Dr. Aminul Islam', 'Cardiology',
phone_numbers_type('880-11111'), address_type('123 Gulshan Ave', 'Dhaka',
'Dhaka', '1212'))),
('DR002', doctor_type('DR002', 'Dr. Fahmida Chowdhury', 'Neurology',
phone_numbers_type('880-22222'), address_type('47 Dhanmondi Rd', 'Dhaka',
'Dhaka', '1205'))),
('DR003', doctor_type('DR003', 'Dr. Kabir Hasan', 'Orthopedics',
phone_numbers_type('880-33333'), address_type('32 Uttara Sector', 'Dhaka',
'Dhaka', '1230'))),
('DR004', doctor_type('DR004', 'Dr. Nusrat Mahbub', 'Pediatrics',
phone_numbers_type('880-44444'), address_type('5 Mirpur St', 'Dhaka', 'Dhaka',
'1216'))),
```

```sql
('DR005', doctor_type('DR005', 'Dr. Tamim Iqbal', 'Dermatology',
phone_numbers_type('880-55555'), address_type('90 Banani Rd', 'Dhaka', 'Dhaka',
'1213')));

INSERT INTO Patients (patient_id, patient_info) VALUES
('P001', patient_type('P001', 'Mohammad Rahman', 45, 'Male',
phone_numbers_type('880-66666'), address_type('10 Chittagong Rd', 'Chittagong',
'Chittagong', '4000'))),
('P002', patient_type('P002', 'Fatema Begum', 38, 'Female',
phone_numbers_type('880-77777'), address_type('22 Rajshahi Ave', 'Rajshahi',
'Rajshahi', '6000'))),
('P003', patient_type('P003', 'Abul Kalam', 50, 'Male',
phone_numbers_type('880-88888'), address_type('18 Sylhet Ln', 'Sylhet',
'Sylhet', '3100'))),
('P004', patient_type('P004', 'Rashida Akter', 30, 'Female',
phone_numbers_type('880-99999'), address_type('34 Khulna Blvd', 'Khulna',
'Khulna', '9100'))),
('P005', patient_type('P005', 'Imran Hossain', 27, 'Male',
phone_numbers_type('880-101010'), address_type('56 Barisal St', 'Barisal',
'Barisal', '8200')));

INSERT INTO Departments (department_id, department_name, head_doctor_id) VALUES
('D001', 'Cardiology', 'DR001'),
('D002', 'Neurology', 'DR002'),
('D003', 'Orthopedics', 'DR003'),
('D004', 'Pediatrics', 'DR004'),
('D005', 'Dermatology', 'DR005');

INSERT INTO Appointments (appointment_id, patient_id, doctor_id,
appointment_date, reason) VALUES
('A001', 'P001', 'DR001', TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Routine
Checkup'),
('A002', 'P002', 'DR002', TO_DATE('2023-03-18', 'YYYY-MM-DD'), 'Consultation'),
('A003', 'P003', 'DR003', TO_DATE('2023-03-20', 'YYYY-MM-DD'), 'Orthopedic
Evaluation'),
('A004', 'P004', 'DR004', TO_DATE('2023-03-22', 'YYYY-MM-DD'), 'Pediatric
Exam'),
('A005', 'P005', 'DR005', TO_DATE('2023-03-25', 'YYYY-MM-DD'), 'Skin Checkup');

INSERT INTO Staff (staff_id, staff_info) VALUES
('S001', staff_type('S001', 'Nurse Ayesha', 'Nurse', phone_numbers_type('880-
121212'), address_type('11 Mirpur Rd', 'Dhaka', 'Dhaka', '1216'))),
('S002', staff_type('S002', 'Lab Tech Mahmud', 'Lab Technician',
phone_numbers_type('880-131313'), address_type('5 Banani', 'Dhaka', 'Dhaka',
'1213'))),
('S003', staff_type('S003', 'Receptionist Karim', 'Receptionist',
phone_numbers_type('880-141414'), address_type('90 Uttara', 'Dhaka', 'Dhaka',
'1230'))),
('S004', staff_type('S004', 'Nurse Rahim', 'Nurse', phone_numbers_type('880-
151515'), address_type('32 Dhanmondi', 'Dhaka', 'Dhaka', '1205'))),
('S005', staff_type('S005', 'Pharmacist Lily', 'Pharmacist',
phone_numbers_type('880-161616'), address_type('18 Gulshan', 'Dhaka', 'Dhaka',
'1212')));

INSERT INTO Medical_Equipment (equipment_id, equipment_info) VALUES
('E001', equipment_type('E001', 'EKG Machine', 'Diagnostic', 'D001')),
('E002', equipment_type('E002', 'EEG Machine', 'Diagnostic', 'D002')),
('E003', equipment_type('E003', 'X-Ray Machine', 'Imaging', 'D003')),
('E004', equipment_type('E004', 'Ultrasound Machine', 'Imaging', 'D004')),
('E005', equipment_type('E005', 'Dermascope', 'Diagnostic', 'D005'));

INSERT INTO Wards (ward_id, ward_info) VALUES
('W001', ward_type('W001', 'Cardiac Care Unit', 15, 'D001')),
('W002', ward_type('W002', 'Neurology Ward', 12, 'D002')),
('W003', ward_type('W003', 'Orthopedic Ward', 10, 'D003')),
('W004', ward_type('W004', 'Pediatric Ward', 20, 'D004')),
('W005', ward_type('W005', 'Dermatology Ward', 8, 'D005'));

INSERT INTO Pharmacy (medicine_id, medicine_list) VALUES
('M001', medicine_list_type('Paracetamol', 'Amlodipine', 'Cetirizine')),
('M002', medicine_list_type('Atorvastatin', 'Metformin', 'Salbutamol')),
('M003', medicine_list_type('Ranitidine', 'Amoxicillin', 'Azithromycin')),
('M004', medicine_list_type('Loratadine', 'Prednisolone', 'Hydrocortisone')),
('M005', medicine_list_type('Insulin', 'Ibuprofen', 'Diclofenac'));

INSERT INTO Insurance (insurance_id, insurance_info) VALUES
('I001', insurance_type('I001', 'P001', 'Delta Life Insurance', 'Comprehensive
Plan')),
('I002', insurance_type('I002', 'P002', 'MetLife Insurance', 'Basic Health
Coverage')),
('I003', insurance_type('I003', 'P003', 'Guardian Life Insurance', 'Family
Plan')),
('I004', insurance_type('I004', 'P004', 'Pragati Insurance', 'Standard Plan')),
('I005', insurance_type('I005', 'P005', 'Green Delta Insurance', 'Premium
Health Plan'));
```

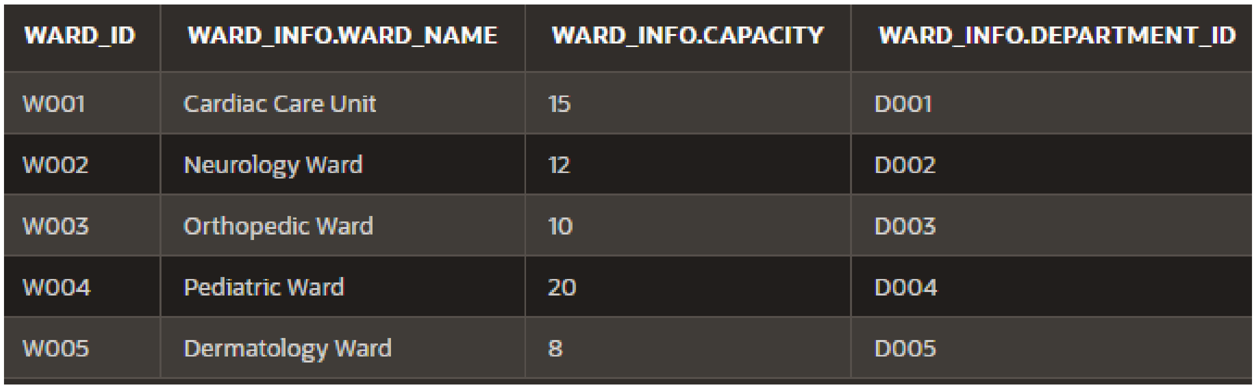
e. Query

```
Query 1: List of All Doctors with Their Specializations
This query retrieves all doctors along with their specializations.
SELECT
    d.doctor_id,
    d.doctor_info.name AS doctor_name,
    d.doctor_info.specialization
FROM
    Doctors d;
```



DOCTOR_ID	DOCTOR_NAME	DOCTOR_INFO.SPECIALIZATION
DR003	Dr. Kabir Hasan	Orthopedics
DR005	Dr. Tamim Iqbal	Dermatology
DR004	Dr. Nusrat Mahbub	Pediatrics
DR001	Dr. Aminul Islam	Cardiology
DR002	Dr. Fahmida Chowdhury	Neurology

```
Query 2: Details of All Wards and Their Respective Departments
This query shows details of wards along with the departments they are
associated with.
SELECT
    w.ward_id,
    w.ward_info.ward_name,
    w.ward_info.capacity,
    w.ward_info.department_id
FROM
    Wards w;
```



WARD_ID	WARD_INFO.WARD_NAME	WARD_INFO.CAPACITY	WARD_INFO.DEPARTMENT_ID
W001	Cardiac Care Unit	15	D001
W002	Neurology Ward	12	D002
W003	Orthopedic Ward	10	D003
W004	Pediatric Ward	20	D004
W005	Dermatology Ward	8	D005

Query 3: List of Doctors in a Specific Department
This query lists all doctors working in a specific department, assuming that the department is linked to doctors.

```
SELECT
    d.doctor_id,
    d.doctor_info.name AS doctor_name
FROM
    Doctors d
WHERE
    d.department_id = 'DeptID';
```

Query 4: Overview of Medication Stock in the Pharmacy
Function

```
CREATE OR REPLACE FUNCTION ConvertMedicineListToString(medicine_list IN
medicine_list_type) RETURN VARCHAR2 IS
    result_str VARCHAR2(4000) := '';
BEGIN
    FOR i IN 1..medicine_list.COUNT LOOP
        result_str := result_str || medicine_list(i) || ', ';
    END LOOP;
    RETURN RTRIM(result_str, ', ');
END;
```

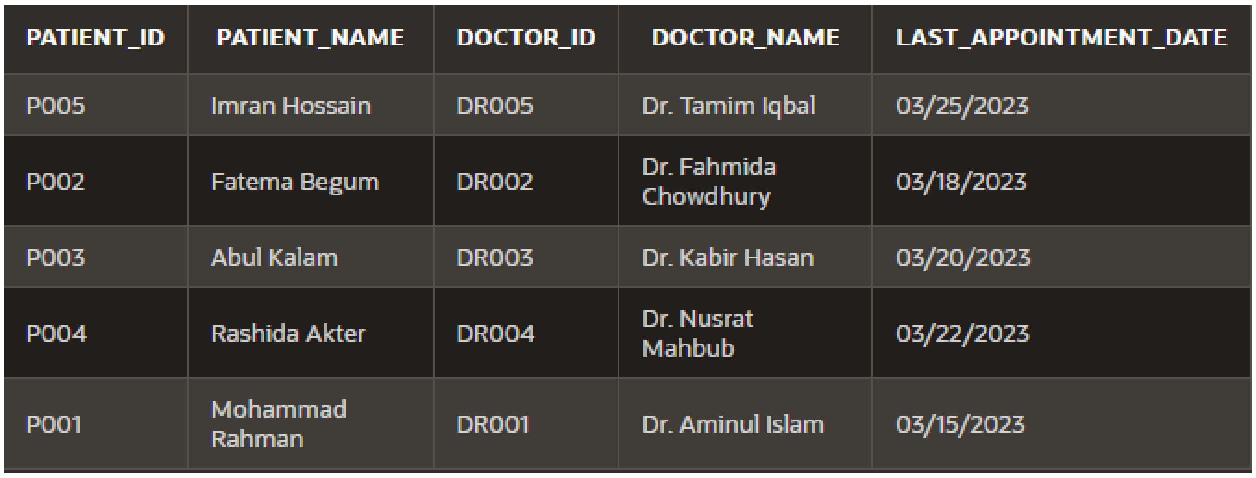
Query 5.

```
SELECT
    ph.medicine_id,
    ConvertMedicineListToString(ph.medicine_list) AS medicine_names
FROM
    Pharmacy ph;
```

Query 6: Details of All Patients Covered by a Specific Insurance Provider

```
SELECT
    p.patient_id,
    p.patient_info.name AS patient_name,
    i.insurance_info.provider_name
FROM
    Patients p
JOIN
    Insurance i ON p.patient_id = i.insurance_info.patient_id
WHERE
    i.insurance_info.provider_name = 'Delta Life Insurance';
```

```
Query 7: List of All Patients with Their Assigned Doctor and Last Appointment
Date
SELECT
    p.patient_id,
    p.patient_info.name AS patient_name,
    d.doctor_id,
    d.doctor_info.name AS doctor_name,
    MAX(a.appointment_date) AS last_appointment_date
FROM
    Patients p
JOIN
    Appointments a ON p.patient_id = a.patient_id
JOIN
    Doctors d ON a.doctor_id = d.doctor_id
GROUP BY
    p.patient_id, p.patient_info.name, d.doctor_id, d.doctor_info.name;
```

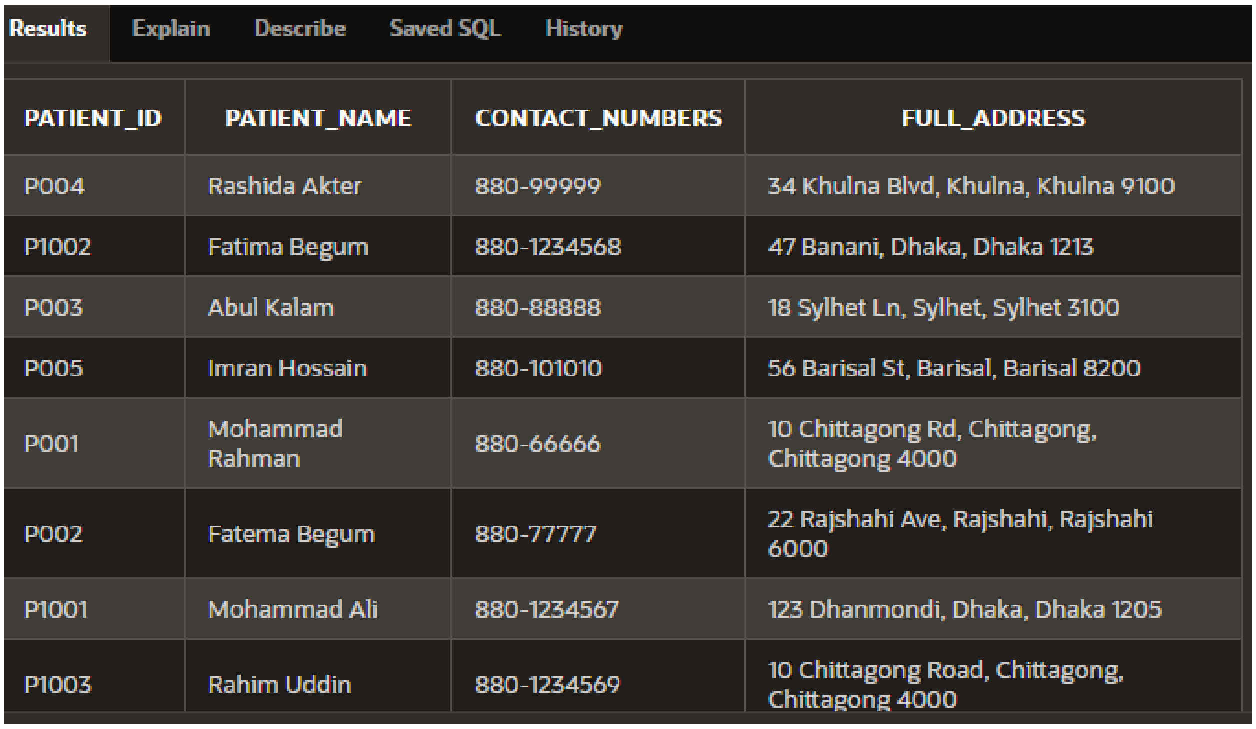


PATIENT_ID	PATIENT_NAME	DOCTOR_ID	DOCTOR_NAME	LAST_APPOINTMENT_DATE
P005	Imran Hossain	DR005	Dr. Tamim Iqbal	03/25/2023
P002	Fatema Begum	DR002	Dr. Fahmida Chowdhury	03/18/2023
P003	Abul Kalam	DR003	Dr. Kabir Hasan	03/20/2023
P004	Rashida Akter	DR004	Dr. Nusrat Mahbub	03/22/2023
P001	Mohammad Rahman	DR001	Dr. Aminul Islam	03/15/2023

```
Query 8: Equipment List in Each Department
SELECT
    d.department_name,
    e.equipment_id,
    e.equipment_info.name AS equipment_name
FROM
    Medical_Equipment e
JOIN
    Departments d ON e.equipment_info.department_id = d.department_id
ORDER BY
    d.department_name;

Query :Contact Information for All Staff in a Specific Position
fUNCTION
```

```
CREATE OR REPLACE FUNCTION ConvertArrayToString(arr IN phone_numbers_type)
RETURN VARCHAR2 IS
    result_str VARCHAR2(4000) := '';
BEGIN
    FOR i IN 1..arr.COUNT LOOP
        result_str := result_str || arr(i) || ', ';
    END LOOP;
    RETURN RTRIM(result_str, ', '); -- Remove the trailing comma
END;

Query 9.
SELECT
    s.staff_id,
    s.staff_info.name,
    s.staff_info.position,
    ConvertArrayToString(s.staff_info.contact_numbers) AS contact_numbers
FROM
    Staff s
WHERE
    s.staff_info.position = 'Nurse';

Query 10: Insurance Details for Each Patient
SELECT
    p.patient_id,
    p.patient_info.name AS patient_name,
    i.insurance_info.provider_name,
    i.insurance_info.coverage_details
FROM
    Patients p
JOIN
    Insurance i ON p.patient_id = i.insurance_info.patient_id;

Query 11: Listing all patient’s information

CREATE OR REPLACE FUNCTION ConvertContactNumbersToString(contacts IN
phone_numbers_type) RETURN VARCHAR2 IS
    contact_str VARCHAR2(4000) := '';
BEGIN
    FOR i IN 1..contacts.COUNT LOOP
        contact_str := contact_str || contacts(i) || ', ';
    END LOOP;
    RETURN RTRIM(contact_str, ', ');
END;

Query 12:
```

```
SELECT
    p.patient_id,
    p.patient_info.name AS patient_name,
    ConvertContactNumbersToString(p.patient_info.contact_numbers) AS
contact_numbers,
    p.patient_info.address.street || ', ' ||
    p.patient_info.address.city || ', ' ||
    p.patient_info.address.state || ' ' ||
    p.patient_info.address.zip AS full_address
FROM
    Patients p;
```


Results Explain Describe Saved SQL History

PATIENT_ID	PATIENT_NAME	CONTACT_NUMBERS	FULL_ADDRESS
P004	Rashida Akter	880-99999	34 Khulna Blvd, Khulna, Khulna 9100
P1002	Fatima Begum	880-1234568	47 Banani, Dhaka, Dhaka 1213
P003	Abul Kalam	880-88888	18 Sylhet Ln, Sylhet, Sylhet 3100
P005	Imran Hossain	880-101010	56 Barisal St, Barisal, Barisal 8200
P001	Mohammad Rahman	880-66666	10 Chittagong Rd, Chittagong, Chittagong 4000
P002	Fatema Begum	880-77777	22 Rajshahi Ave, Rajshahi, Rajshahi 6000
P1001	Mohammad Ali	880-1234567	123 Dhanmondi, Dhaka, Dhaka 1205
P1003	Rahim Uddin	880-1234569	10 Chittagong Road, Chittagong, Chittagong 4000

```
Query 13: Count of Equipment per Department

SELECT e.equipment_info.department_id, COUNT(*) AS equipment_count
FROM Medical_Equipment e
GROUP BY e.equipment_info.department_id;
check Patients treatment Recoird
SELECT
    p.patient_id,
    mr.treatment_date,
    mr.treatment_description
FROM
    Patients p,
    TABLE(p.medical_records) mr;
```

Database Schema Overview:

Tables and Their Fields:

1. **Departments**
   - **department_id** VARCHAR2(20) **PRIMARY KEY**
   - **department_name** VARCHAR2(100)
   - **head_doctor_id** VARCHAR2(20) [**Foreign Key** referencing **Doctors**]
2. **Doctors**
   - **doctor_id** VARCHAR2(20) **PRIMARY KEY**
   - **doctor_info** **doctor_type** (Custom Object Type)
3. **Patients**
   - **patient_id** VARCHAR2(20) **PRIMARY KEY**
   - **patient_info** **patient_type** (Custom Object Type)
   - **medical_records** **medical_record_type** (Nested Table Type)
4. **Staff**
   - **staff_id** VARCHAR2(20) **PRIMARY KEY**
   - **staff_info** **staff_type** (Custom Object Type)
5. **Medical_Equipment**
   - **equipment_id** VARCHAR2(20) **PRIMARY KEY**
   - **equipment_info** **equipment_type** (Custom Object Type)
   - **Foreign Key**: **equipment_info.department_id** referencing **Departments.department_id**
6. **Wards**
   - **ward_id** VARCHAR2(20) **PRIMARY KEY**
   - **ward_info** **ward_type** (Custom Object Type)
   - Foreign Key: **ward_info.department_id** referencing **Departments.department_id**
7. **Pharmacy**
   - **medicine_id** VARCHAR2(20) **PRIMARY KEY**
   - **medicine_list** **medicine_list_type** (VARRAY or Nested Table)
8. **Insurance**

- **insurance_id** VARCHAR2(20) PRIMARY KEY
- **insurance_info** **insurance_type** (Custom Object Type)
- Foreign Key: **insurance_info.patient_id** referencing **Patients.patient_id**

9. **Appointments**
    - **appointment_id** VARCHAR2(20) **PRIMARY KEY**
    - **patient_id** VARCHAR2(20) [Foreign Key referencing **Patients**]
    - **doctor_id** VARCHAR2(20) [Foreign Key referencing **Doctors**]
    - **appointment_date** DATE
    - **reason** VARCHAR2(255)

Relationships:

- **Departments to Doctors**: One-to-Many (One department can have many doctors, but a doctor is associated with one department)
- **Doctors to Patients (via Appointments)**: Many-to-Many (A doctor can have many patients, and a patient can see many doctors)
- **Patients to Insurance**: One-to-One (One patient can have one insurance policy)
- **Departments to Wards**: One-to-Many (A department can have multiple wards)
- **Departments to Medical Equipment**: One-to-Many (A department can have multiple pieces of equipment)

Custom Types:

- doctor_type: Custom Object Type (Includes doctor's details)
- patient_type: Custom Object Type (Includes patient's details)
- staff_type: Custom Object Type (Includes staff member's details)
- equipment_type: Custom Object Type (Includes equipment details)
- ward_type: Custom Object Type (Includes ward details)
- insurance_type: Custom Object Type (Includes insurance policy details)
- medical_record_type: Nested Table Type (Includes medical records)
- medicine_list_type: VARRAY or Nested Table (Includes list of medicines)

Schema:

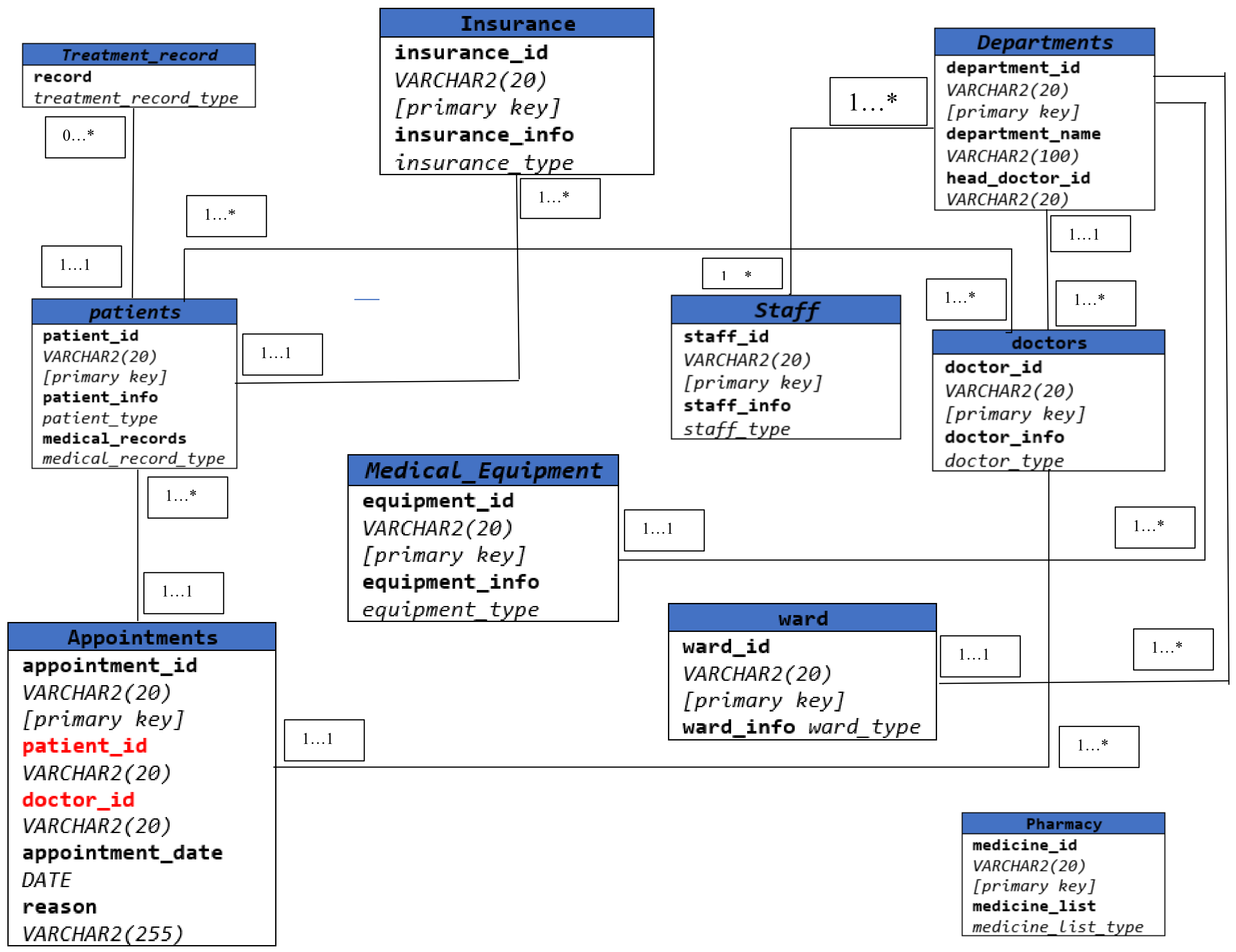


Fig: Database Schema