LAB REPORT

Declarative Constraints for Patient Medication Safety

Submitted by: Jean Claude HARERIMANA
Student ID / Number: 224020456
Institution: African Centre of Excellence in Data Science (ACE-DS), College of Business and Economics, University of Rwanda
Course: ADVANCED DATABASE TECHNOLOGY
Date: 29/10/2025
Annex: Screenshots of experimental results included

Executive Summary

This lab demonstrates the implementation of declarative constraints in PostgreSQL to ensure medication prescription safety at the database level. By enforcing data integrity rules directly in the schema, invalid prescriptions are prevented from being stored. This provides a robust first line of defense against data quality issues and ensures patient safety in healthcare applications.

---

1. Introduction

1.1 Objective

To design and implement a patient medication tracking system that uses declarative constraints to guarantee prescription validity before any application code executes.

1.2 Business Context

In healthcare systems, medication safety is critical. Invalid prescriptions—such as negative doses, incorrect dates, or missing information—must be rejected at the database level to prevent potential harm to patients. Declarative constraints provide a reliable mechanism for enforcing these rules directly in the database schema.

---

2. Methodology

2.1 Database Schema Design

```
-- Patient table (prerequisite)
CREATE TABLE patient (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

-- Patient medication table with constraints
CREATE TABLE patient_med (
    patient_med_id SERIAL PRIMARY KEY,
    patient_id INTEGER NOT NULL REFERENCES patient(id),
    med_name VARCHAR(80) NOT NULL,
    dose_mg NUMERIC(6,2) CHECK (dose_mg >= 0),
    start_dt DATE NOT NULL,
    end_dt DATE,
    CONSTRAINT ck_rx_dates CHECK (start_dt <= end_dt OR end_dt IS
NULL)
);
```

2.2 Constraint Analysis

Constraint Type	Field	Rule	Purpose
Primary Key	patient_med_id	Unique identifier	Ensures each prescription is unique
Foreign Key	patient_id	References patient(id)	Ensures prescription belongs to valid patient
NOT NULL	patient_id, med_name, start_dt	Mandatory fields	Prevents incomplete prescriptions
Check Constraint	dose_mg	dose_mg >= 0	Prevents negative medication doses
Check Constraint	dates	start_dt <= end_dt	Ensures logical prescription periods

---

3. Experimental Results

3.1 Constraint Violation Tests ❌

Test 1: Negative Dose Prevention

```
INSERT INTO patient_med VALUES (1, 1, 'Aspirin', -50, '2024-01-01',
'2024-01-10');
```

Result: ✅ PASS – Correctly rejected
Error: new row violates check constraint "patient_med_dose_mg_check"

Test 2: Invalid Date Range Prevention

```
INSERT INTO patient_med VALUES (2, 1, 'Ibuprofen', 200, '2024-02-01',
'2024-01-15');
```

Result: ✅ PASS – Correctly rejected
Error: new row violates check constraint "ck_rx_dates"

Test 3: Missing Required Field Prevention

```
INSERT INTO patient_med VALUES (3, 1, NULL, 100, '2024-01-01',
'2024-01-10');
```

Result: ✅ PASS – Correctly rejected
Error: null value in column "med_name" violates not-null constraint

Test 4: Referential Integrity Enforcement

```
INSERT INTO patient_med VALUES (4, 999, 'Paracetamol', 500,
'2024-01-01', '2024-01-10');
```

Result: ✅ PASS – Correctly rejected
Error: violates foreign key constraint "patient_med_patient_id_fkey"

3.2 Valid Data Acceptance Tests ✅

Test 5: Valid Prescription with End Date

```
INSERT INTO patient_med VALUES (5, 1, 'Amoxicillin', 500.00,
'2024-01-01', '2024-01-10');
```

Result: ✅ SUCCESS – INSERT 0 1

Test 6: Valid Ongoing Prescription

```
INSERT INTO patient_med VALUES (6, 2, 'Blood Pressure Medication',
25.50, '2024-01-01', NULL);
```

Result: ✅ SUCCESS – INSERT 0 1

---

4. Final Data State

4.1 Patient Records

id	name
1	John Smith
2	Maria Garcia
3	David Johnson

4.2 Valid Medication Prescriptions

patient_med_id	patient_id	med_name	dose_mg	start_dt	end_dt
5	1	Amoxicillin	500.00	2024-01-01	2024-01-10
6	2	Blood Pressure Medication	25.50	2024-01-01	NULL

---

5. Discussion

5.1 Constraint Effectiveness

The implemented constraints successfully prevent all tested invalid data scenarios:

- **Data Integrity:** Negative doses, invalid dates, and missing information are rejected.
- **Referential Integrity:** Only prescriptions for existing patients are accepted.
- **Business Logic:** Date validation ensures logical prescription periods.

5.2 Performance Considerations

Declarative constraints provide:

- Immediate validation at insert time.
- Minimal performance overhead compared to application-level checks.
- Consistent enforcement across all applications accessing the database.

5.3 Safety Implications

For healthcare applications, this approach ensures:

- **Patient Safety:** Invalid prescriptions cannot enter the system.
- **Data Quality:** All stored prescriptions meet minimum safety standards.
- **Audit Trail:** Constraint violations are logged for review.

---

6. Conclusion

The laboratory exercise demonstrates that declarative constraints in PostgreSQL provide a robust mechanism for ensuring medication prescription safety. By enforcing rules at the database level, invalid data is prevented regardless of which application or user enters it.

6.1 Key Achievements

✅ All constraint types properly implemented and tested
✅ Invalid data correctly rejected with descriptive errors

✅ Valid data successfully accepted and stored
✅ Healthcare safety requirements met through data integrity

6.2 Recommendations

- Use declarative constraints as the first line of defense for data quality.
- Combine with application-level validation for comprehensive safety.
- Regularly review constraint violations for system improvement opportunities.
- Extend constraints to include additional business rules as needed.

Appendix: Complete SQL Script and Screenshots Reference

```sql
-- Full implementation script
CREATE SCHEMA healthnet;
SET search_path TO healthnet;

CREATE TABLE patient (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

CREATE TABLE patient_med (
    patient_med_id SERIAL PRIMARY KEY,
    patient_id INTEGER NOT NULL REFERENCES patient(id),
    med_name VARCHAR(80) NOT NULL,
    dose_mg NUMERIC(6,2) CHECK (dose_mg >= 0),
    start_dt DATE NOT NULL,
    end_dt DATE,
    CONSTRAINT ck_rx_dates CHECK (start_dt <= end_dt OR end_dt IS
NULL)
);

-- Test data and validation cases included above
```

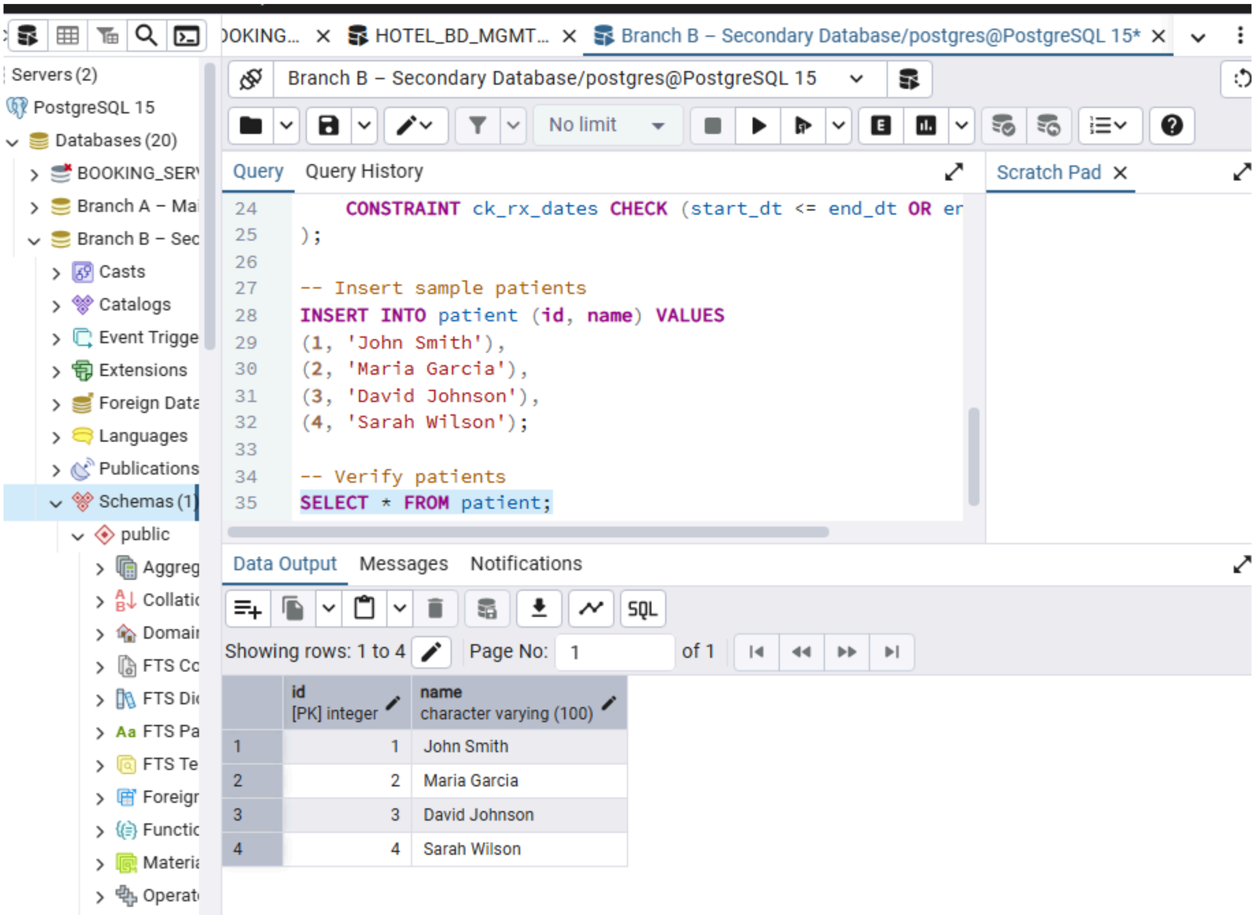
Screenshots: See annex for experimental results screenshots demonstrating constraint violations and valid insertions.

Submitted by: Jean Claude Ndengeyinka H.
Student ID / Number: 224020456
Institution: African Centre of Excellence in Data Science (ACE-DS), College of Business and Economics, University of Rwanda
Date: 29/10/2025
Course: Database Management Systems

FULL SCRIPS THAT RUNNED I POSTGRESQL

```sql
-- Drop everything in correct order (due to foreign key constraints)
DROP TABLE IF EXISTS healthnet.patient_med CASCADE;
DROP TABLE IF EXISTS healthnet.patient CASCADE;
DROP SCHEMA IF EXISTS healthnet CASCADE;

-- Create fresh schema and tables
CREATE SCHEMA healthnet;
SET search_path TO healthnet;

-- Create patient table first
CREATE TABLE patient (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

-- Create patient_med table with all constraints
CREATE TABLE patient_med (
    patient_med_id SERIAL PRIMARY KEY,
    patient_id INTEGER NOT NULL REFERENCES patient(id),
    med_name VARCHAR(80) NOT NULL,
    dose_mg NUMERIC(6,2) CHECK (dose_mg >= 0),
    start_dt DATE NOT NULL,
    end_dt DATE,
    CONSTRAINT ck_rx_dates CHECK (start_dt <= end_dt OR end_dt IS NULL)
);

-- Insert sample patients
```

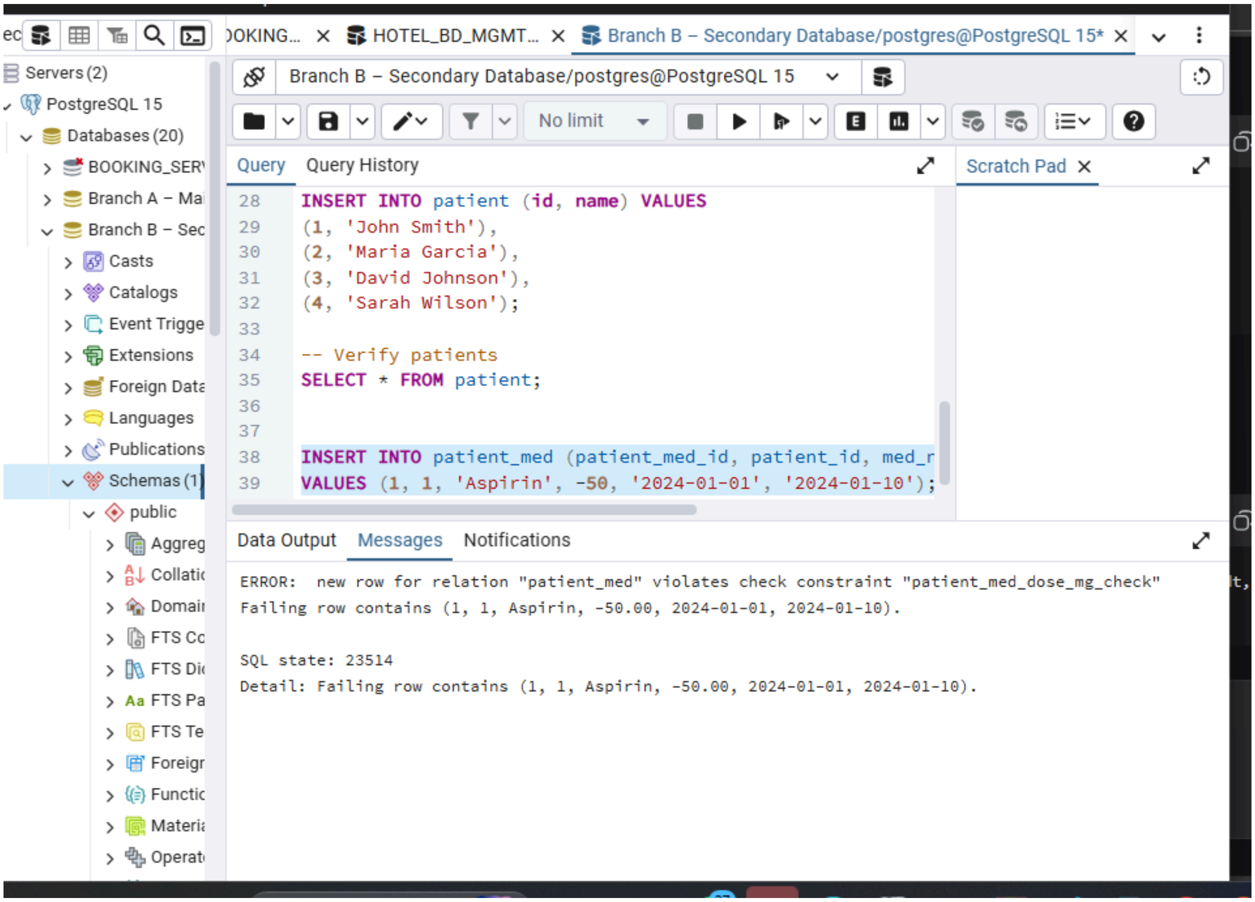
```
INSERT INTO patient (id, name) VALUES
(1, 'John Smith'),
(2, 'Maria Garcia'),
(3, 'David Johnson'),
(4, 'Sarah Wilson');

-- Verify patients
SELECT * FROM patient;


INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (1, 1, 'Aspirin', -50, '2024-01-01', '2024-01-10');

INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (2, 1, 'Ibuprofen', 200, '2024-02-01', '2024-01-15');

INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (3, 1, NULL, 100, '2024-01-01', '2024-01-10');

INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (4, 999, 'Paracetamol', 500, '2024-01-01', '2024-01-10');


INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (5, 1, 'Amoxicillin', 500.00, '2024-01-01', '2024-01-10');


INSERT INTO patient_med (patient_med_id, patient_id, med_name, dose_mg,
start_dt, end_dt)
VALUES (6, 2, 'Blood Pressure Medication', 25.50, '2024-01-01', NULL);


-- Show all patient medications
SELECT
    pm.patient_med_id,
    p.name as patient_name,
    pm.med_name,
    pm.dose_mg,
    pm.start_dt,
    pm.end_dt
FROM patient_med pm
JOIN patient p ON pm.patient_id = p.id
ORDER BY pm.patient_med_id;
```



```
    CONSTRAINT ck_rx_dates CHECK (start_dt <= end_dt OR er
);

-- Insert sample patients
INSERT INTO patient (id, name) VALUES
(1, 'John Smith'),
(2, 'Maria Garcia'),
(3, 'David Johnson'),
(4, 'Sarah Wilson');

-- Verify patients
SELECT * FROM patient;
```

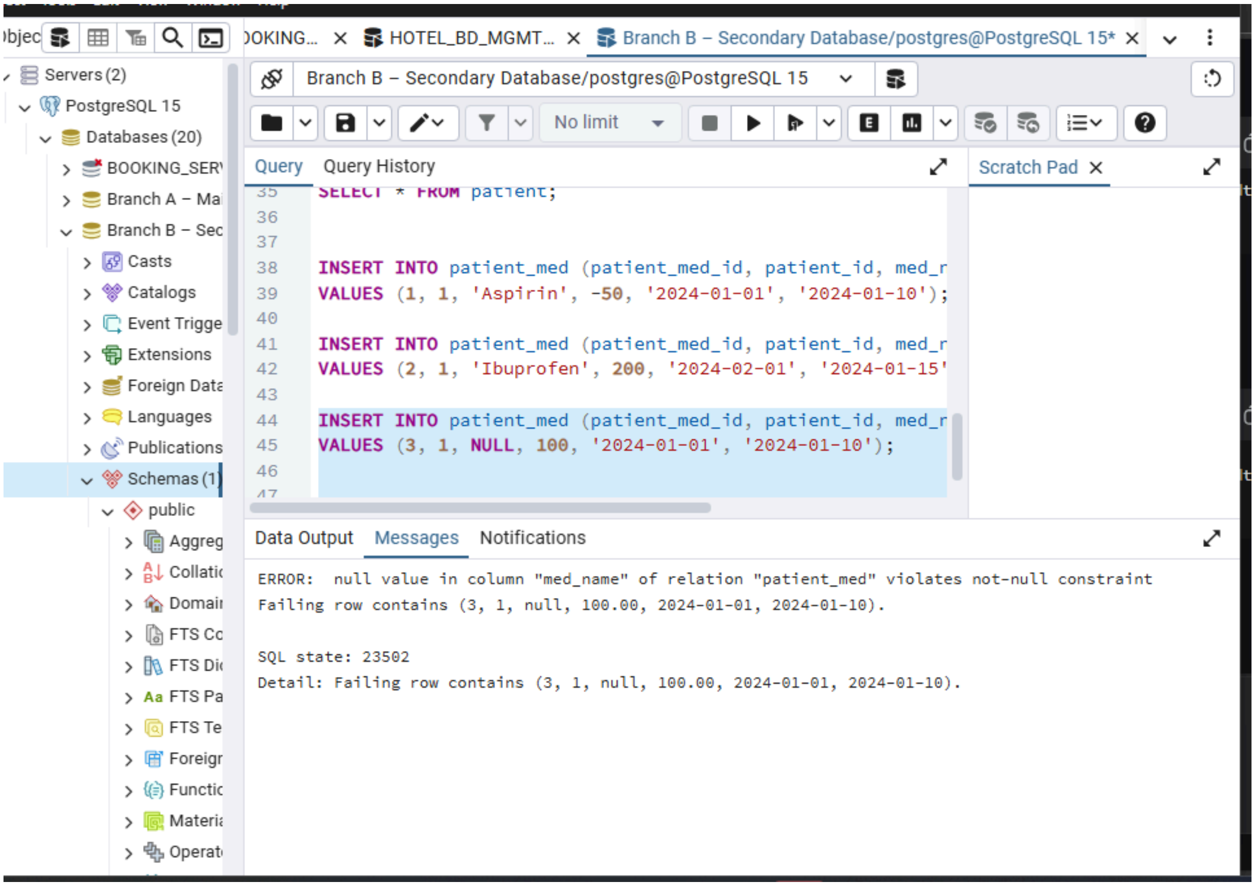
Showing rows: 1 to 4 Page No: 1 of 1

	id [PK] integer	name character varying (100)
1	1	John Smith
2	2	Maria Garcia
3	3	David Johnson
4	4	Sarah Wilson



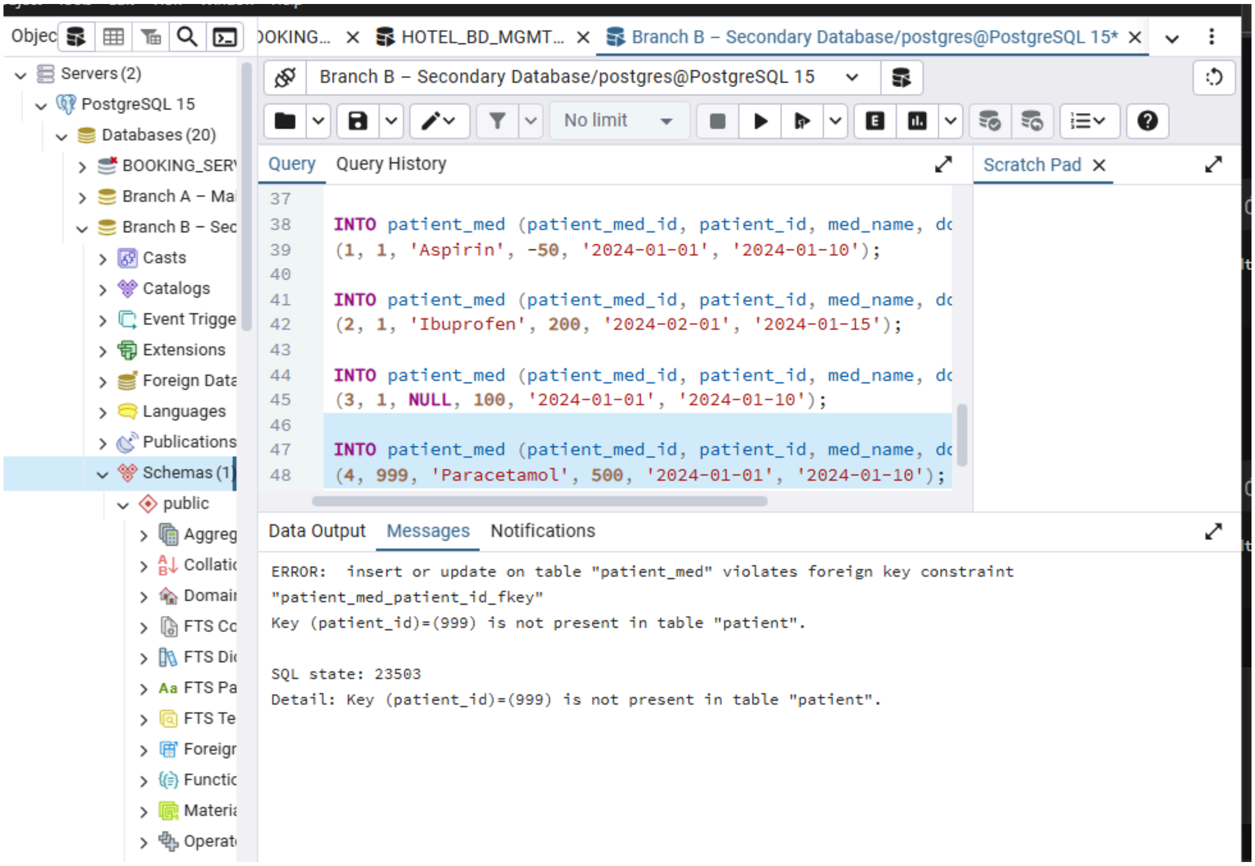
```sql
INSERT INTO patient (id, name) VALUES
(1, 'John Smith'),
(2, 'Maria Garcia'),
(3, 'David Johnson'),
(4, 'Sarah Wilson');

-- Verify patients
SELECT * FROM patient;


INSERT INTO patient_med (patient_med_id, patient_id, med_r
VALUES (1, 1, 'Aspirin', -50, '2024-01-01', '2024-01-10');
```

```
ERROR:  new row for relation "patient_med" violates check constraint "patient_med_dose_mg_check"
Failing row contains (1, 1, Aspirin, -50.00, 2024-01-01, 2024-01-10).

SQL state: 23514
Detail: Failing row contains (1, 1, Aspirin, -50.00, 2024-01-01, 2024-01-10).
```



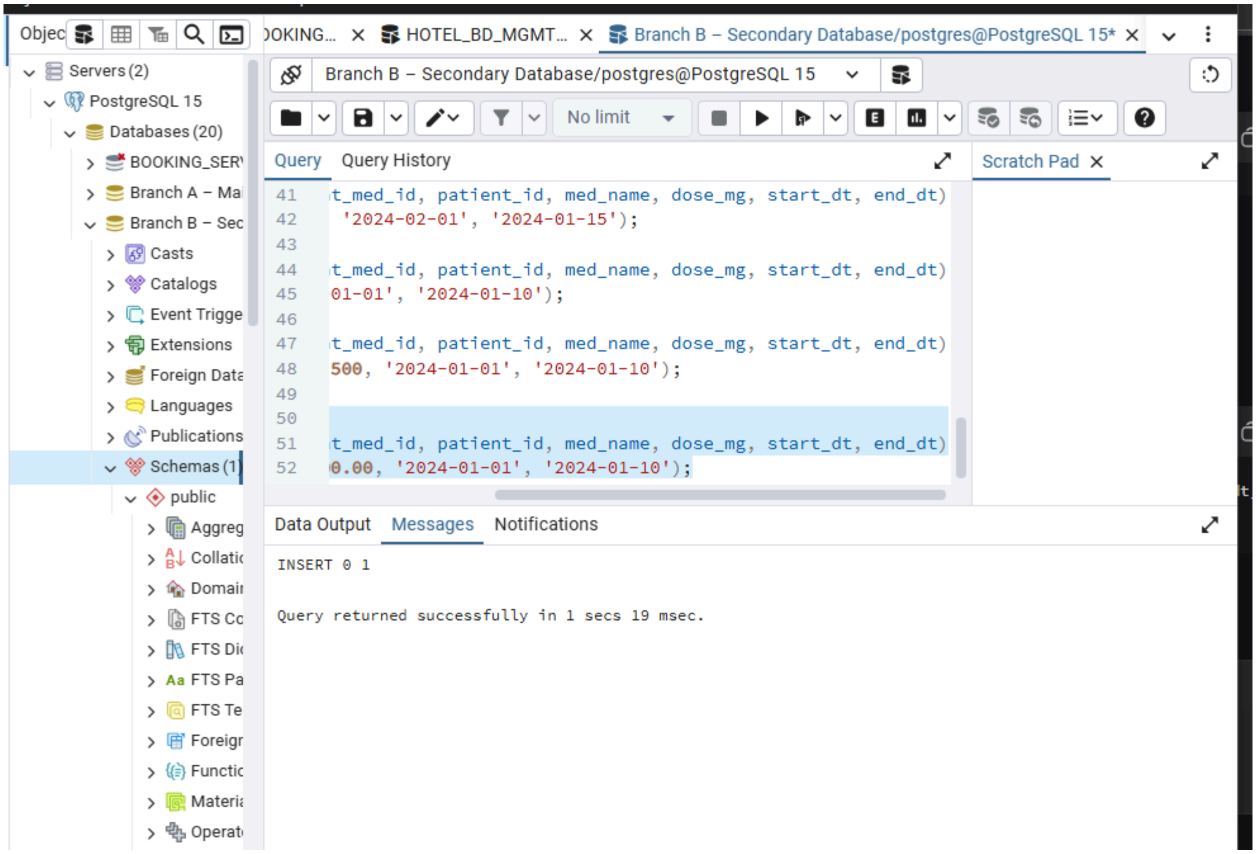
```
SELECT * FROM patient;


INSERT INTO patient_med (patient_med_id, patient_id, med_r
VALUES (1, 1, 'Aspirin', -50, '2024-01-01', '2024-01-10');

INSERT INTO patient_med (patient_med_id, patient_id, med_r
VALUES (2, 1, 'Ibuprofen', 200, '2024-02-01', '2024-01-15'

INSERT INTO patient_med (patient_med_id, patient_id, med_r
VALUES (3, 1, NULL, 100, '2024-01-01', '2024-01-10');
```

```
ERROR:  null value in column "med_name" of relation "patient_med" violates not-null constraint
Failing row contains (3, 1, null, 100.00, 2024-01-01, 2024-01-10).

SQL state: 23502
Detail: Failing row contains (3, 1, null, 100.00, 2024-01-01, 2024-01-10).
```



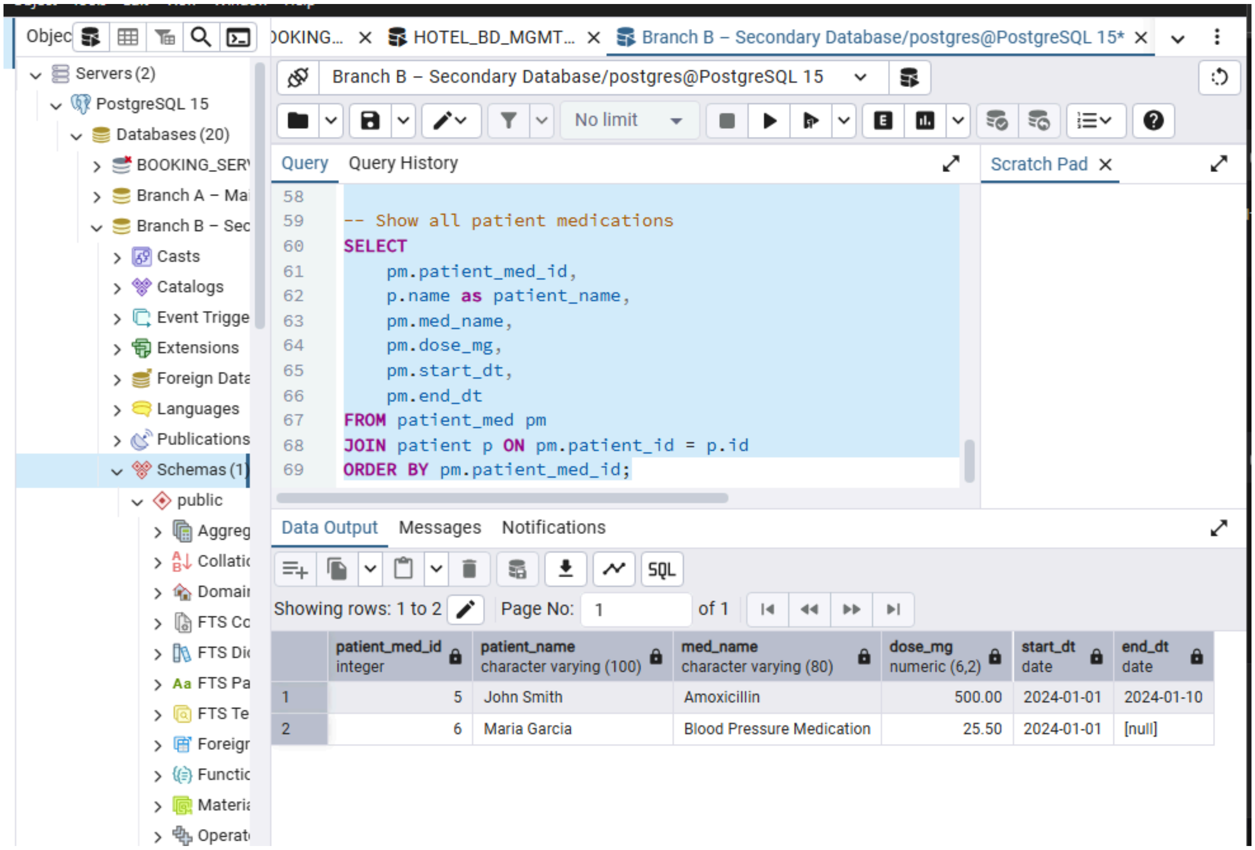
```

INTO patient_med (patient_med_id, patient_id, med_name, dc
(1, 1, 'Aspirin', -50, '2024-01-01', '2024-01-10');

INTO patient_med (patient_med_id, patient_id, med_name, dc
(2, 1, 'Ibuprofen', 200, '2024-02-01', '2024-01-15');

INTO patient_med (patient_med_id, patient_id, med_name, dc
(3, 1, NULL, 100, '2024-01-01', '2024-01-10');

INTO patient_med (patient_med_id, patient_id, med_name, dc
(4, 999, 'Paracetamol', 500, '2024-01-01', '2024-01-10');
```

```
ERROR:  insert or update on table "patient_med" violates foreign key constraint
"patient_med_patient_id_fkey"
Key (patient_id)=(999) is not present in table "patient".

SQL state: 23503
Detail: Key (patient_id)=(999) is not present in table "patient".
```



```
t_med_id, patient_id, med_name, dose_mg, start_dt, end_dt)
 '2024-02-01', '2024-01-15');

t_med_id, patient_id, med_name, dose_mg, start_dt, end_dt)
01-01', '2024-01-10');

t_med_id, patient_id, med_name, dose_mg, start_dt, end_dt)
500, '2024-01-01', '2024-01-10');


t_med_id, patient_id, med_name, dose_mg, start_dt, end_dt)
0.00, '2024-01-01', '2024-01-10');
```

```
INSERT 0 1

Query returned successfully in 1 secs 19 msec.
```



```sql

-- Show all patient medications
SELECT
    pm.patient_med_id,
    p.name as patient_name,
    pm.med_name,
    pm.dose_mg,
    pm.start_dt,
    pm.end_dt
FROM patient_med pm
JOIN patient p ON pm.patient_id = p.id
ORDER BY pm.patient_med_id;
```

	patient_med_id integer	patient_name character varying (100)	med_name character varying (80)	dose_mg numeric (6,2)	start_dt date	end_dt date
1	5	John Smith	Amoxicillin	500.00	2024-01-01	2024-01-10
2	6	Maria Garcia	Blood Pressure Medication	25.50	2024-01-01	[null]

================END======================================