HOSPITAL MANAGEMENT SYSTEM

A Relational SQL Case Study with Business Insights

Created by: Alisha Mahanty
Role: Aspiring Data Analyst

GitHub: https://github.com/AlishaMahanty85/Healthcare_Project-SQL

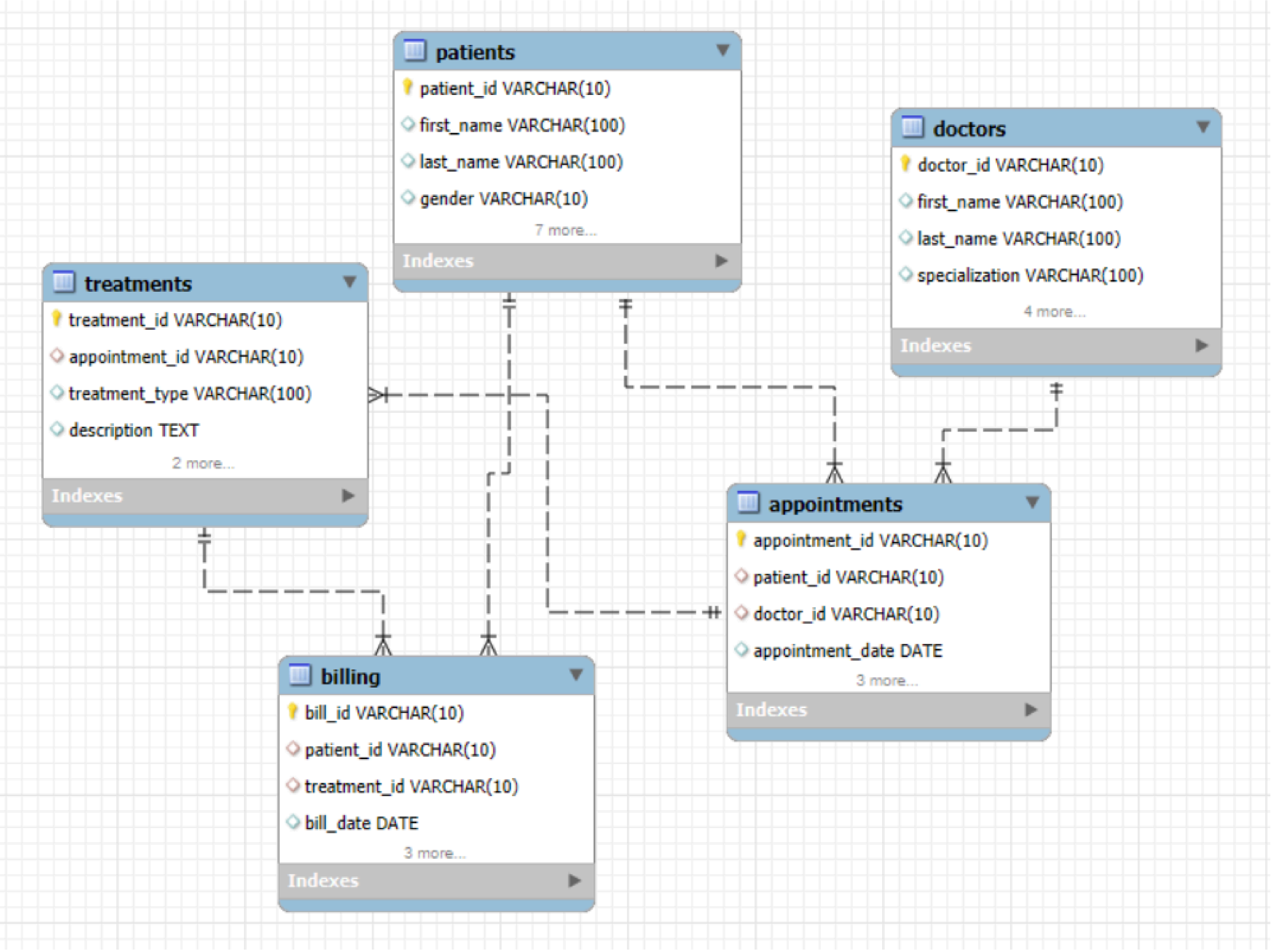
Project Summary

- Built a Hospital Management System using MySQL to simulate real-world healthcare operations.
- Designed 5 relational tables: Patients, Doctors, Appointments, Treatments, Billing.
- Applied primary and foreign keys to maintain referential integrity between tables.
- Imported and managed ~1000+ rows of synthetic patient and hospital data.
- Structured data using appropriate VARCHAR, DATE, DECIMAL types and constraints.
- Created an ER diagram to visualize relationships between all entities.
- Performed complex JOINs across multiple tables for integrated reporting.
- Used aggregations (SUM, COUNT) and filtering (HAVING, WHERE) for KPI analysis.
- Applied advanced SQL like RANK() window functions and DATE_FORMAT().
- Solved 10 real-world business problems including revenue tracking, patient billing, and doctor performance.

Key Business Questions

1. Which patients visited for consultation?
   → Identified visit reasons using condition-based filters
2. How many appointments did each doctor handle?
   → Aggregated appointment counts using GROUP BY
3. Which patients were billed over ₹10,000 in total?
   → Used SUM() and HAVING to highlight high-cost cases
4. What are the top 3 reasons patients visit the hospital?
   → Ranked reasons using COUNT() and ORDER BY
5. Which treatments were provided, by whom, and at what cost?
   → Combined data using multiple JOINs across 3+ tables
6. What is the billing status of each appointment?
   → Tracked billing status with LEFT JOIN logic
7. How much revenue was generated each month?
   → Summarized with DATE_FORMAT() and SUM() by month
8. When was each patient’s most recent visit?
   → Extracted with MAX(appointment date) per patient
9. Which doctor treated the most unique patients?
   → Counted distinct patients per doctor with ranking
10. Who are the top 10 patients based on total treatment cost?
    → Applied RANK() window function to rank spenders

Entity Relationship Diagram



patients
- patient_id VARCHAR(10)
- first_name VARCHAR(100)
- last_name VARCHAR(100)
- gender VARCHAR(10)
- 7 more...
- Indexes

doctors
- doctor_id VARCHAR(10)
- first_name VARCHAR(100)
- last_name VARCHAR(100)
- specialization VARCHAR(100)
- 4 more...
- Indexes

treatments
- treatment_id VARCHAR(10)
- appointment_id VARCHAR(10)
- treatment_type VARCHAR(100)
- description TEXT
- 2 more...
- Indexes

appointments
- appointment_id VARCHAR(10)
- patient_id VARCHAR(10)
- doctor_id VARCHAR(10)
- appointment_date DATE
- 3 more...
- Indexes

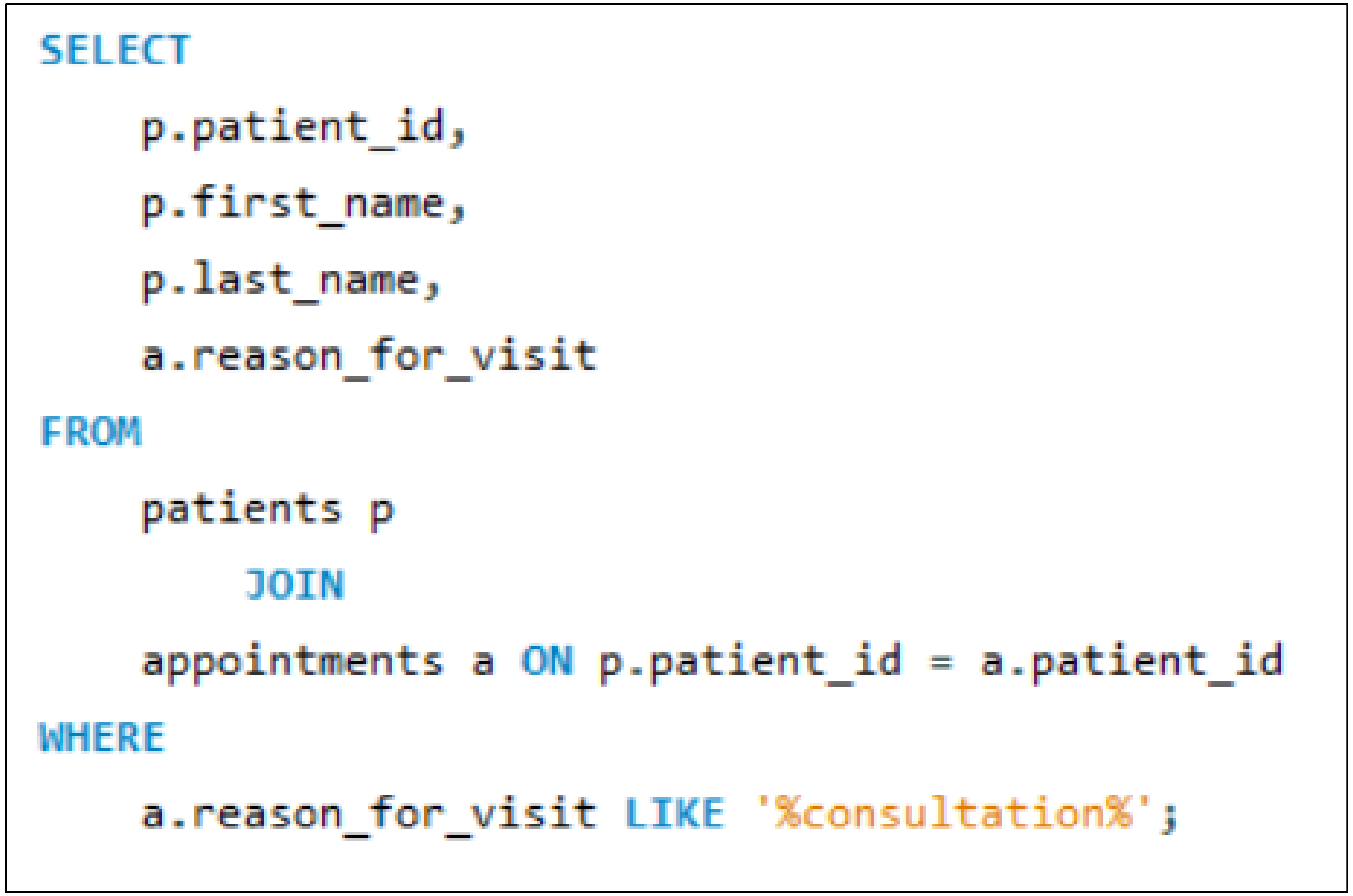
billing
- bill_id VARCHAR(10)
- patient_id VARCHAR(10)
- treatment_id VARCHAR(10)
- bill_date DATE
- 3 more...
- Indexes

Visualizing key relationships across Patients, Doctors, Appointments, Treatments, and Billing tables.

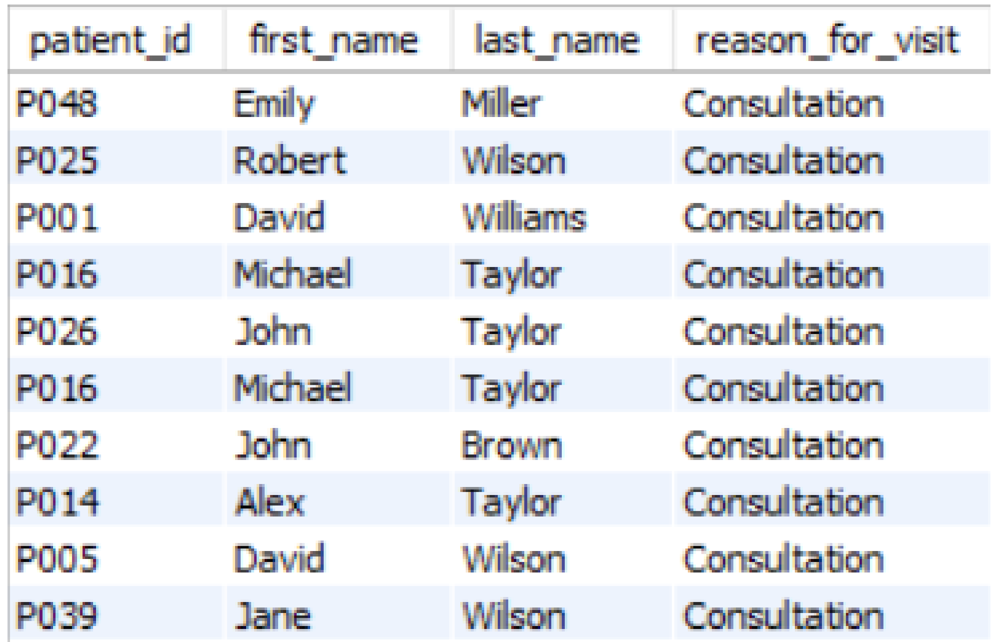
SQL Skills Demonstrated

1. 🔗 JOINs:
   - Integrated data across 2–4 tables using INNER JOIN, LEFT JOIN, and multi-level joins
2. 📊 Aggregation Functions:
   - Used SUM(), COUNT(), MAX() to generate metrics like total revenue, visit frequency, and cost summaries
3. 🧮 Grouping & Filtering:
   - Applied GROUP BY, HAVING, and WHERE clauses for segmented analysis and insights
4. 📅 Date Functions:
   - Implemented DATE_FORMAT(), DATE_ADD() to manage billing cycles, appointment history, and registration trends
5. 💰 Currency Formatting:
   - Used FORMAT() function to present revenue and billing amounts in readable INR format
6. 🧑⚕️ Window Functions:
   - Applied RANK() to compare patients by treatment cost and identify top spenders
7. 📥 Data Constraints & Relationships:
   - Designed schema with PRIMARY KEY, FOREIGN KEY, DECIMAL, VARCHAR, and DATE types for normalization
8. 🧠 Business-Oriented Query Writing:
   - Wrote and optimized 10 structured queries aligned to real-world hospital operations and decision-making

1. Which patients visited for consultation?

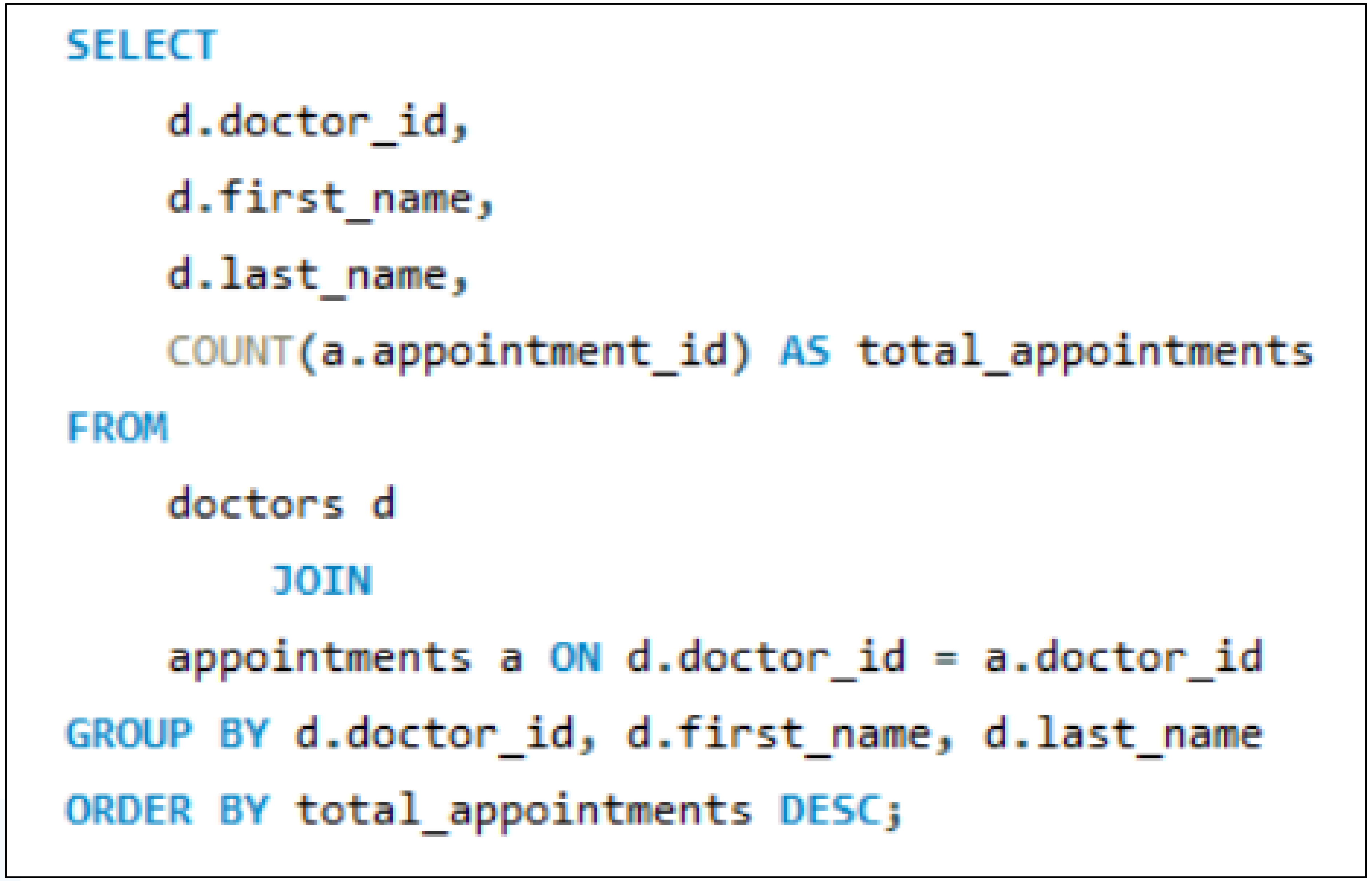


```
SELECT
    p.patient_id,
    p.first_name,
    p.last_name,
    a.reason_for_visit
FROM
    patients p
        JOIN
    appointments a ON p.patient_id = a.patient_id
WHERE
    a.reason_for_visit LIKE '%consultation%';
```

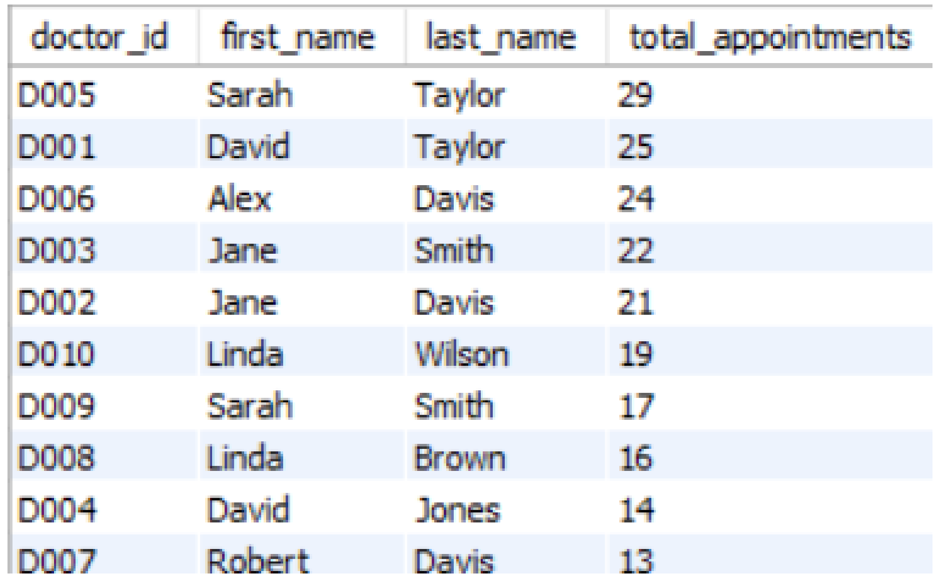


patient_id	first_name	last_name	reason_for_visit
P048	Emily	Miller	Consultation
P025	Robert	Wilson	Consultation
P001	David	Williams	Consultation
P016	Michael	Taylor	Consultation
P026	John	Taylor	Consultation
P016	Michael	Taylor	Consultation
P022	John	Brown	Consultation
P014	Alex	Taylor	Consultation
P005	David	Wilson	Consultation
P039	Jane	Wilson	Consultation

2. How many appointments did each doctor handle?

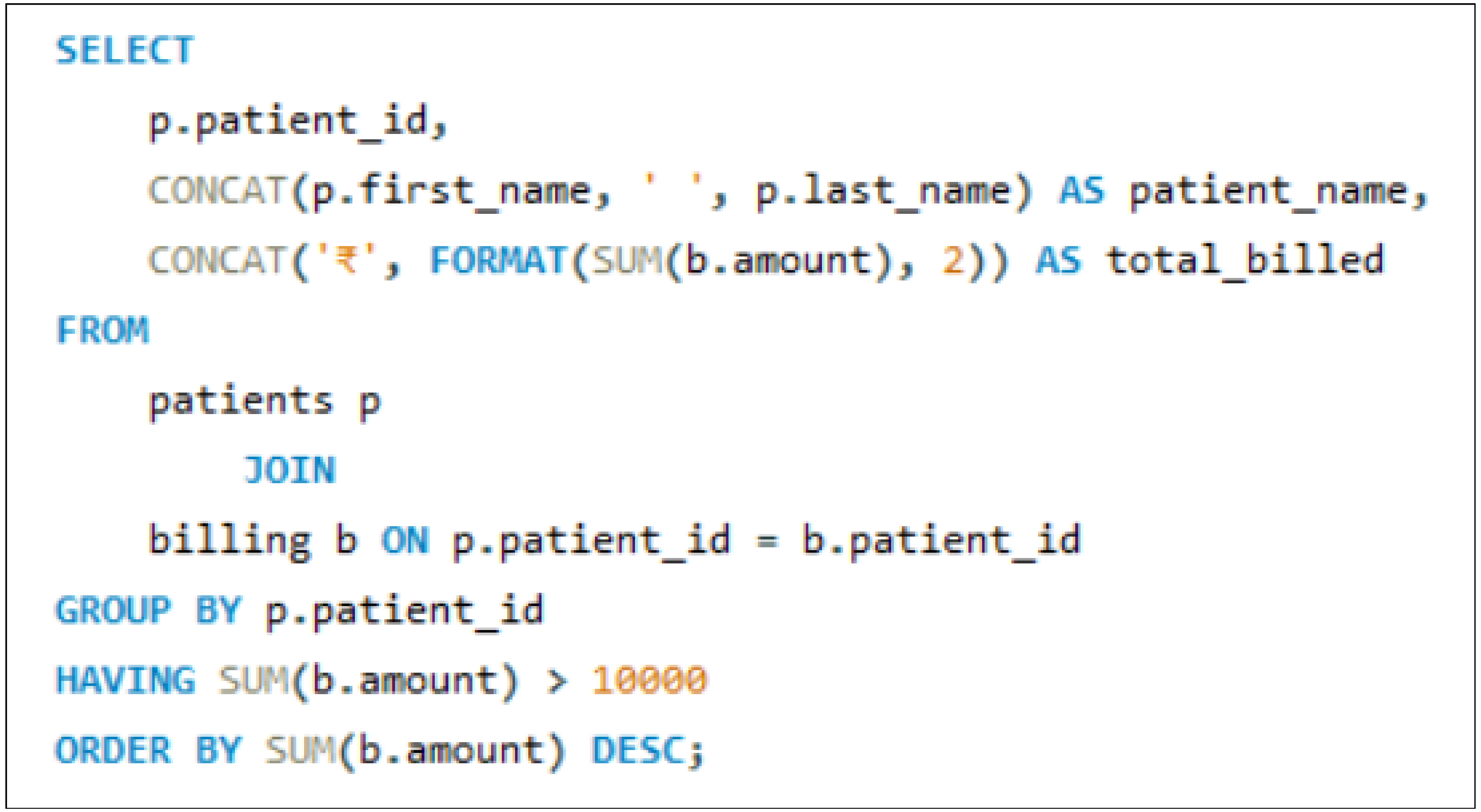


```
SELECT
    d.doctor_id,
    d.first_name,
    d.last_name,
    COUNT(a.appointment_id) AS total_appointments
FROM
    doctors d
        JOIN
    appointments a ON d.doctor_id = a.doctor_id
GROUP BY d.doctor_id, d.first_name, d.last_name
ORDER BY total_appointments DESC;
```

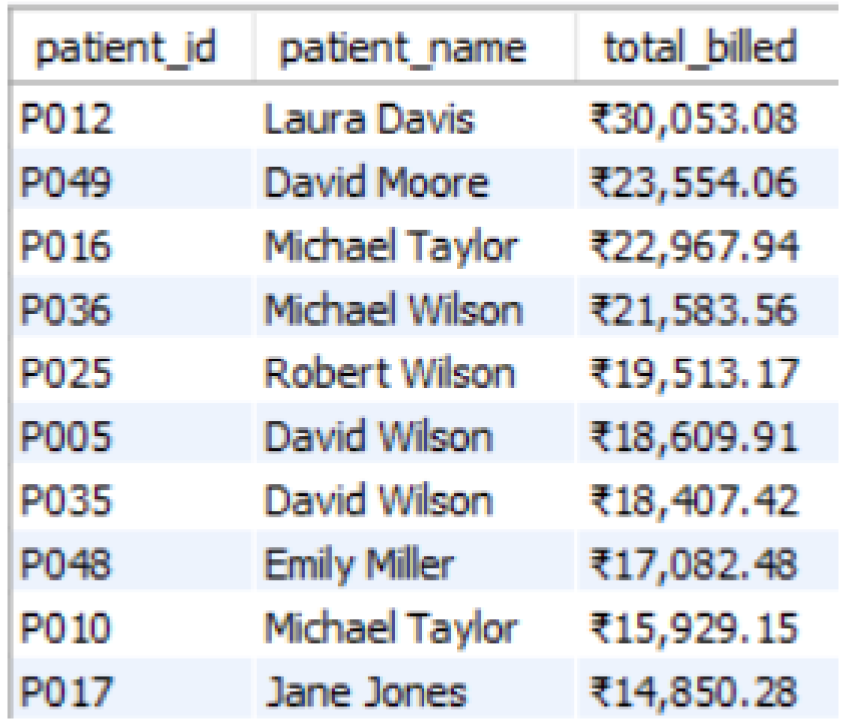


doctor_id	first_name	last_name	total_appointments
D005	Sarah	Taylor	29
D001	David	Taylor	25
D006	Alex	Davis	24
D003	Jane	Smith	22
D002	Jane	Davis	21
D010	Linda	Wilson	19
D009	Sarah	Smith	17
D008	Linda	Brown	16
D004	David	Jones	14
D007	Robert	Davis	13

3. Which patients were billed over ₹10,000 in total?

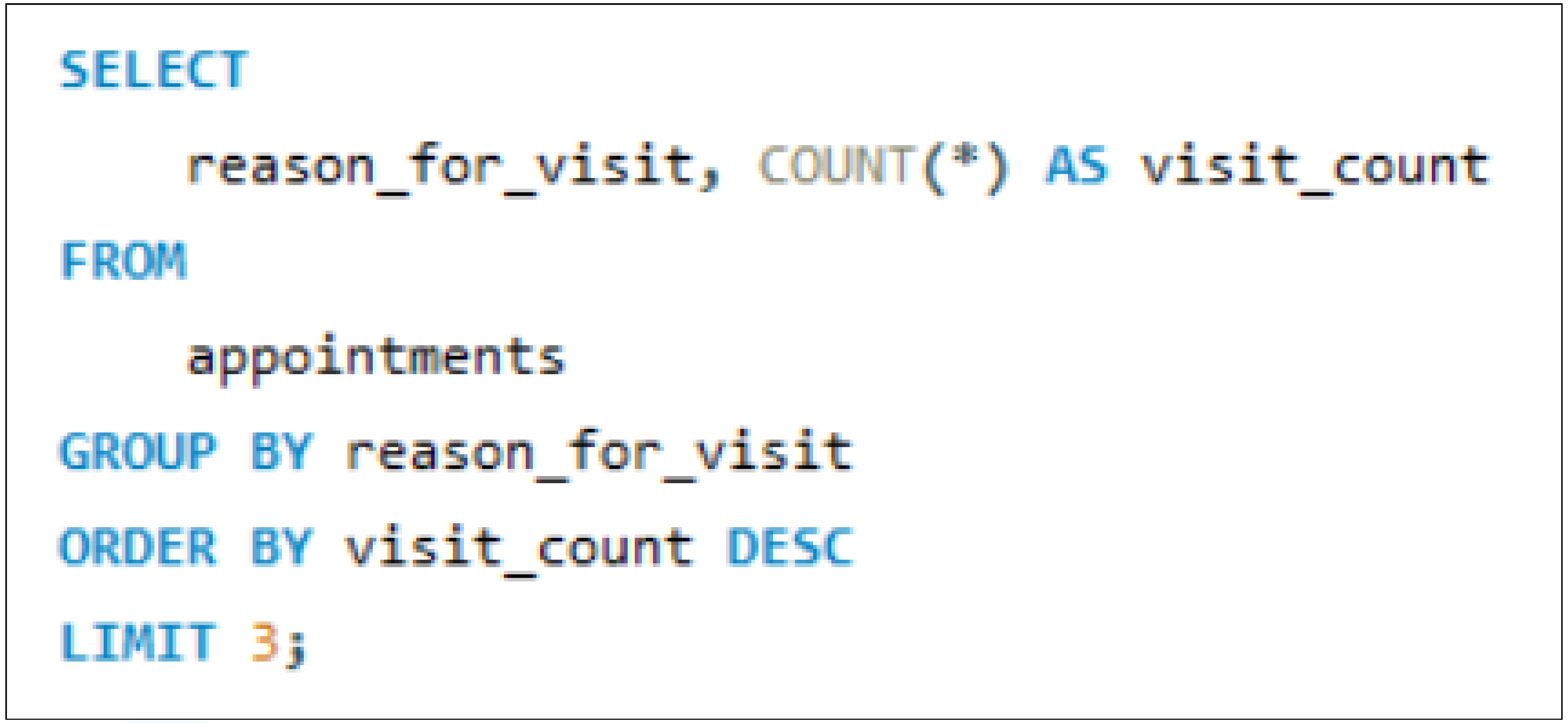


```
SELECT
    p.patient_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    CONCAT('₹', FORMAT(SUM(b.amount), 2)) AS total_billed
FROM
    patients p
        JOIN
    billing b ON p.patient_id = b.patient_id
GROUP BY p.patient_id
HAVING SUM(b.amount) > 10000
ORDER BY SUM(b.amount) DESC;
```

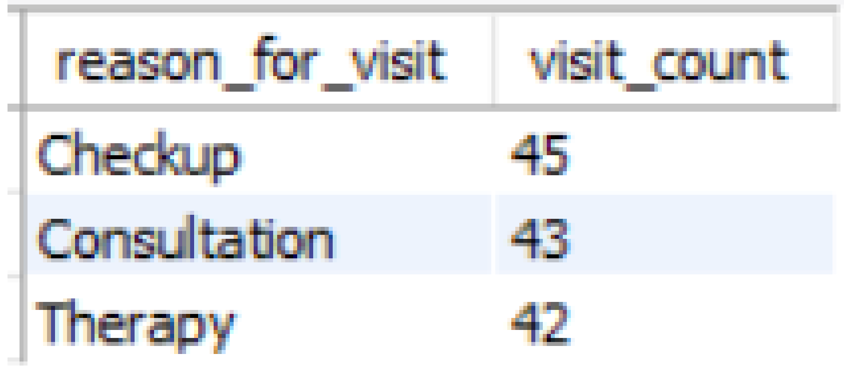


patient_id	patient_name	total_billed
P012	Laura Davis	₹30,053.08
P049	David Moore	₹23,554.06
P016	Michael Taylor	₹22,967.94
P036	Michael Wilson	₹21,583.56
P025	Robert Wilson	₹19,513.17
P005	David Wilson	₹18,609.91
P035	David Wilson	₹18,407.42
P048	Emily Miller	₹17,082.48
P010	Michael Taylor	₹15,929.15
P017	Jane Jones	₹14,850.28

4. What are the top 3 reasons patients visit the hospital?

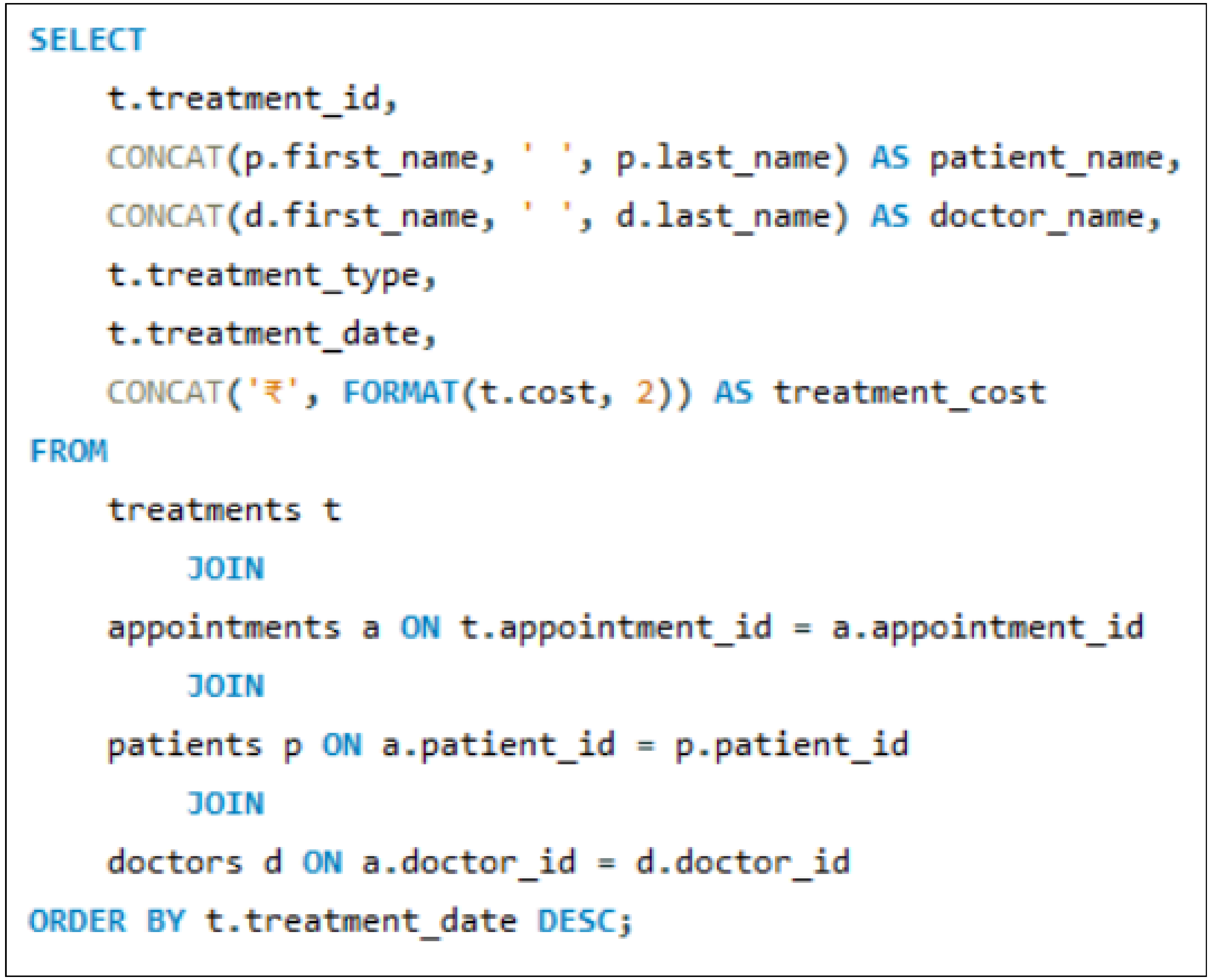


```
SELECT
    reason_for_visit, COUNT(*) AS visit_count
FROM
    appointments
GROUP BY reason_for_visit
ORDER BY visit_count DESC
LIMIT 3;
```

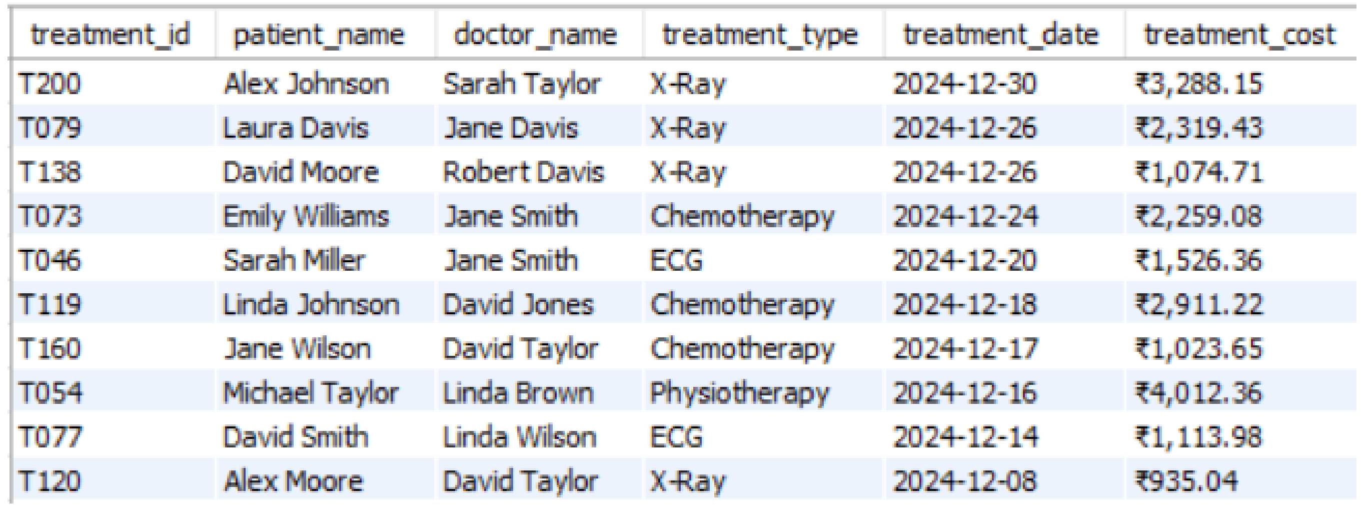


reason_for_visit	visit_count
Checkup	45
Consultation	43
Therapy	42

5. Which treatments were provided, by whom, and at what cost?

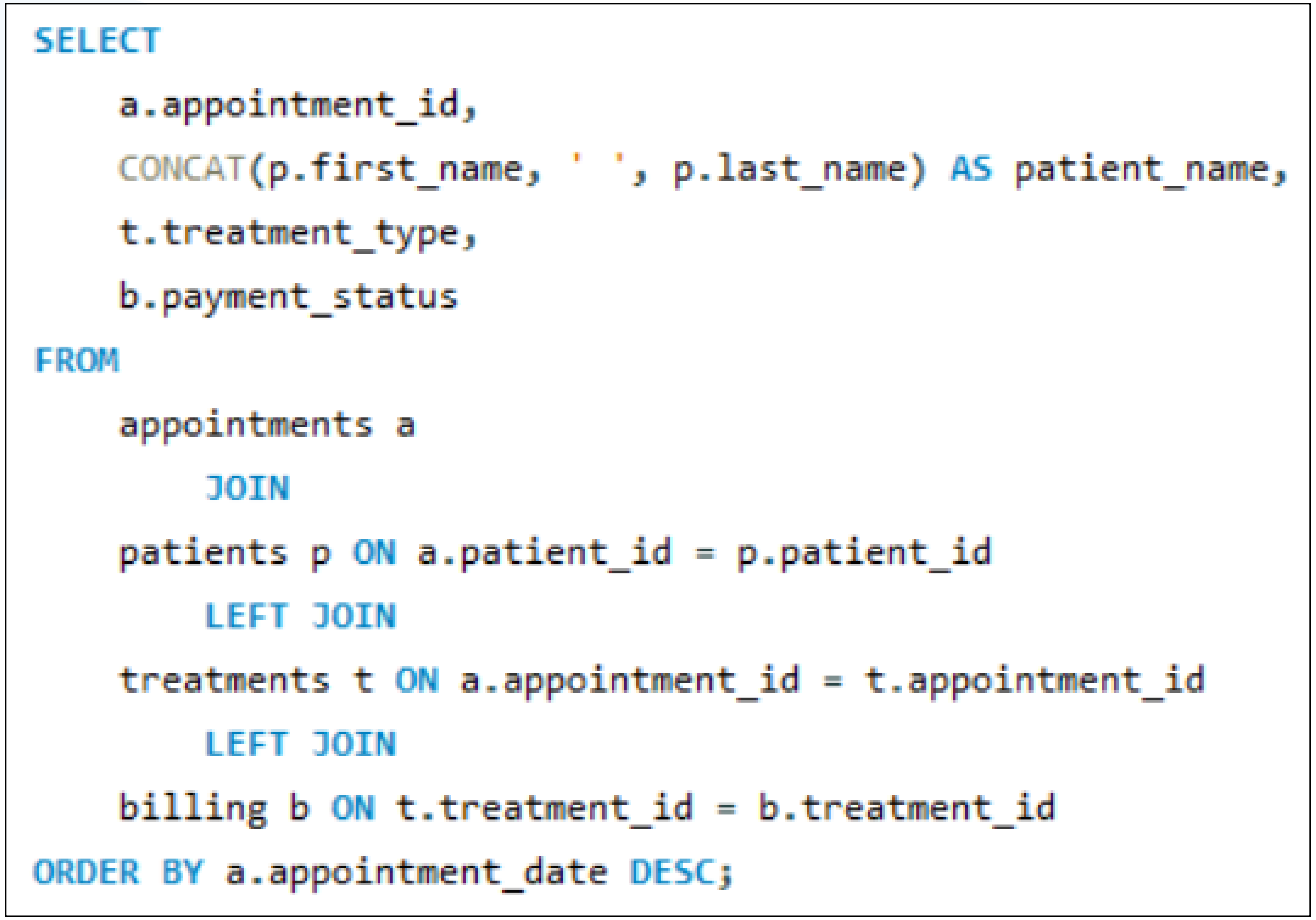


```
SELECT
    t.treatment_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    CONCAT(d.first_name, ' ', d.last_name) AS doctor_name,
    t.treatment_type,
    t.treatment_date,
    CONCAT('₹', FORMAT(t.cost, 2)) AS treatment_cost
FROM
    treatments t
        JOIN
    appointments a ON t.appointment_id = a.appointment_id
        JOIN
    patients p ON a.patient_id = p.patient_id
        JOIN
    doctors d ON a.doctor_id = d.doctor_id
ORDER BY t.treatment_date DESC;
```

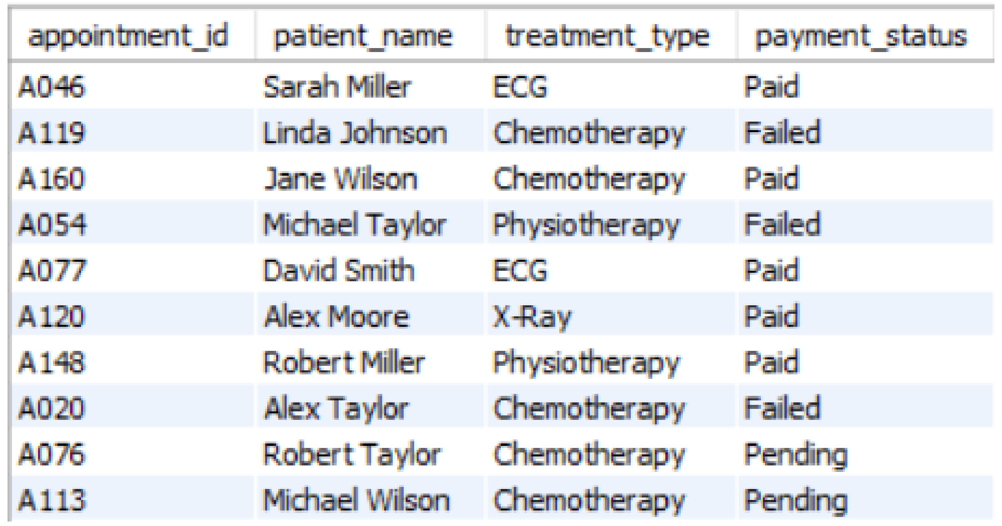


treatment_id	patient_name	doctor_name	treatment_type	treatment_date	treatment_cost
T200	Alex Johnson	Sarah Taylor	X-Ray	2024-12-30	₹3,288.15
T079	Laura Davis	Jane Davis	X-Ray	2024-12-26	₹2,319.43
T138	David Moore	Robert Davis	X-Ray	2024-12-26	₹1,074.71
T073	Emily Williams	Jane Smith	Chemotherapy	2024-12-24	₹2,259.08
T046	Sarah Miller	Jane Smith	ECG	2024-12-20	₹1,526.36
T119	Linda Johnson	David Jones	Chemotherapy	2024-12-18	₹2,911.22
T160	Jane Wilson	David Taylor	Chemotherapy	2024-12-17	₹1,023.65
T054	Michael Taylor	Linda Brown	Physiotherapy	2024-12-16	₹4,012.36
T077	David Smith	Linda Wilson	ECG	2024-12-14	₹1,113.98
T120	Alex Moore	David Taylor	X-Ray	2024-12-08	₹935.04

6. What is the billing status of each appointment?

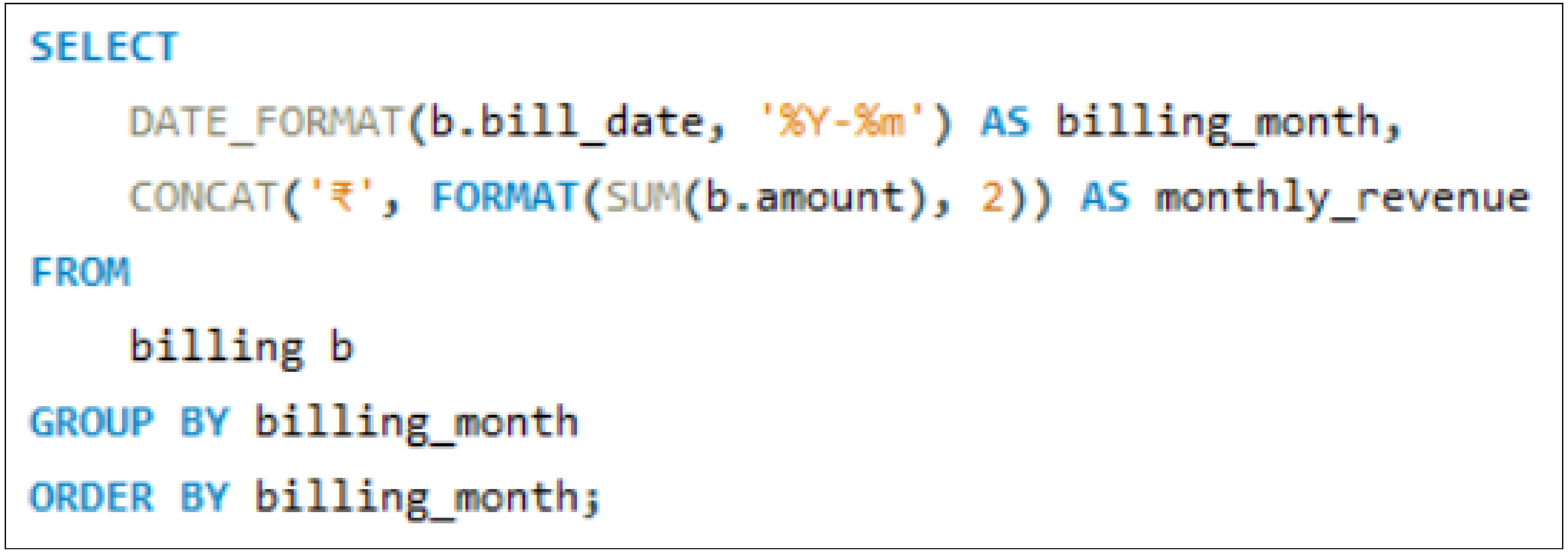


```
SELECT
    a.appointment_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    t.treatment_type,
    b.payment_status
FROM
    appointments a
        JOIN
    patients p ON a.patient_id = p.patient_id
        LEFT JOIN
    treatments t ON a.appointment_id = t.appointment_id
        LEFT JOIN
    billing b ON t.treatment_id = b.treatment_id
ORDER BY a.appointment_date DESC;
```

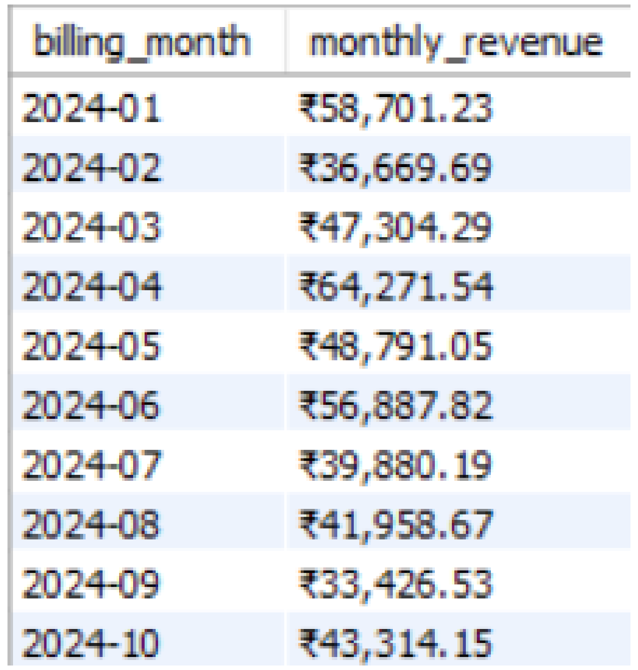


appointment_id	patient_name	treatment_type	payment_status
A046	Sarah Miller	ECG	Paid
A119	Linda Johnson	Chemotherapy	Failed
A160	Jane Wilson	Chemotherapy	Paid
A054	Michael Taylor	Physiotherapy	Failed
A077	David Smith	ECG	Paid
A120	Alex Moore	X-Ray	Paid
A148	Robert Miller	Physiotherapy	Paid
A020	Alex Taylor	Chemotherapy	Failed
A076	Robert Taylor	Chemotherapy	Pending
A113	Michael Wilson	Chemotherapy	Pending

7. How much revenue was generated each month?

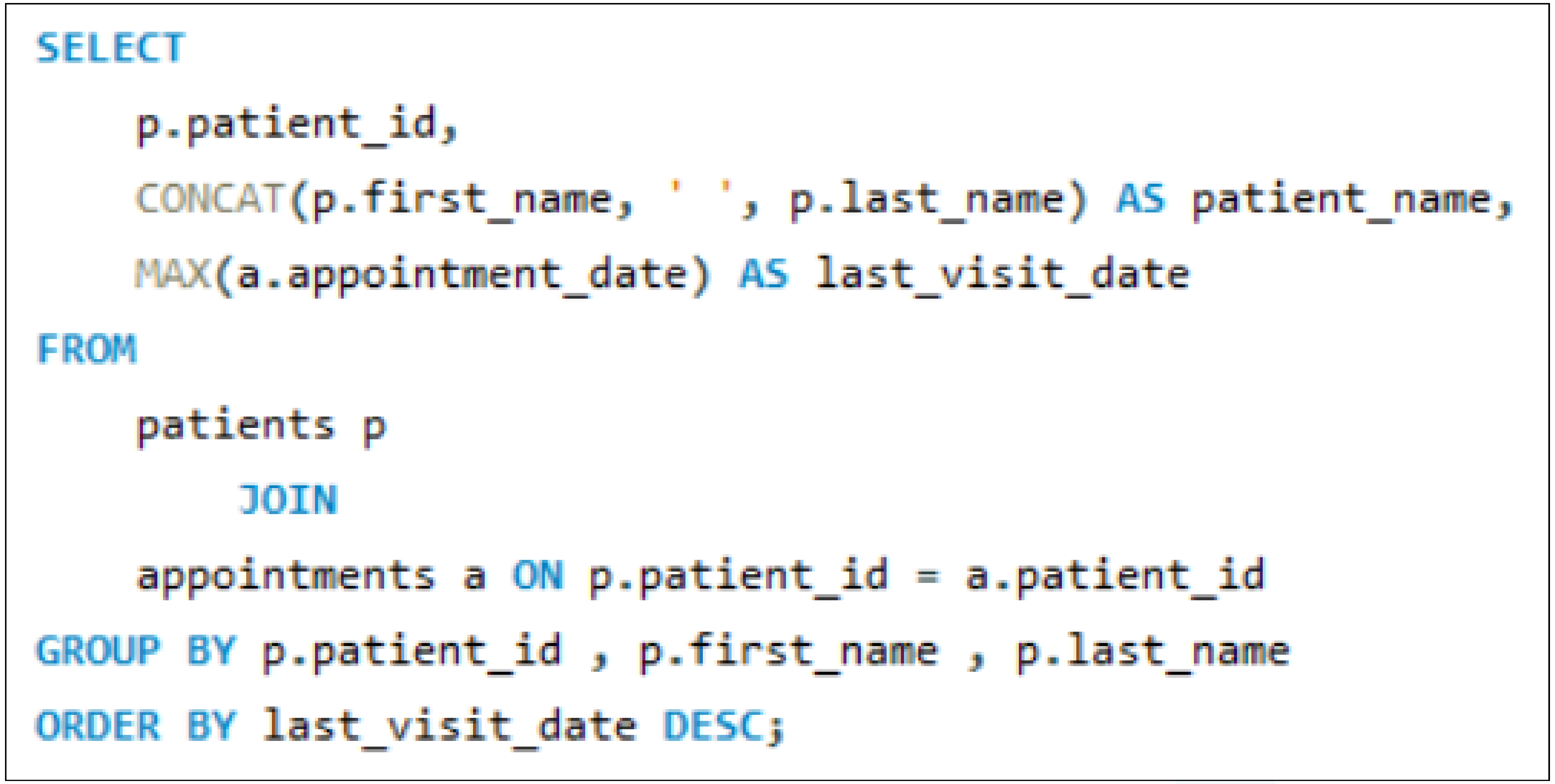


```
SELECT
    DATE_FORMAT(b.bill_date, '%Y-%m') AS billing_month,
    CONCAT('₹', FORMAT(SUM(b.amount), 2)) AS monthly_revenue
FROM
    billing b
GROUP BY billing_month
ORDER BY billing_month;
```

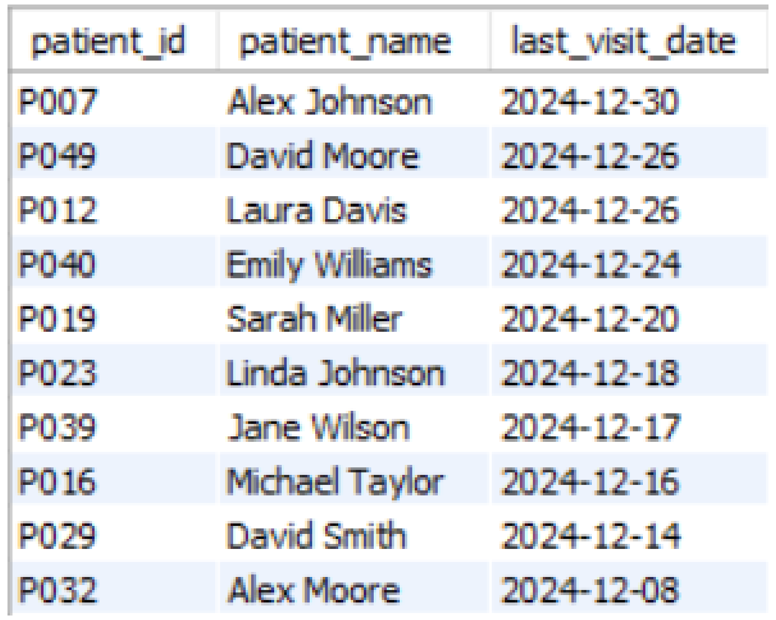


billing_month	monthly_revenue
2024-01	₹58,701.23
2024-02	₹36,669.69
2024-03	₹47,304.29
2024-04	₹64,271.54
2024-05	₹48,791.05
2024-06	₹56,887.82
2024-07	₹39,880.19
2024-08	₹41,958.67
2024-09	₹33,426.53
2024-10	₹43,314.15

8. When was each patient’s most recent visit?

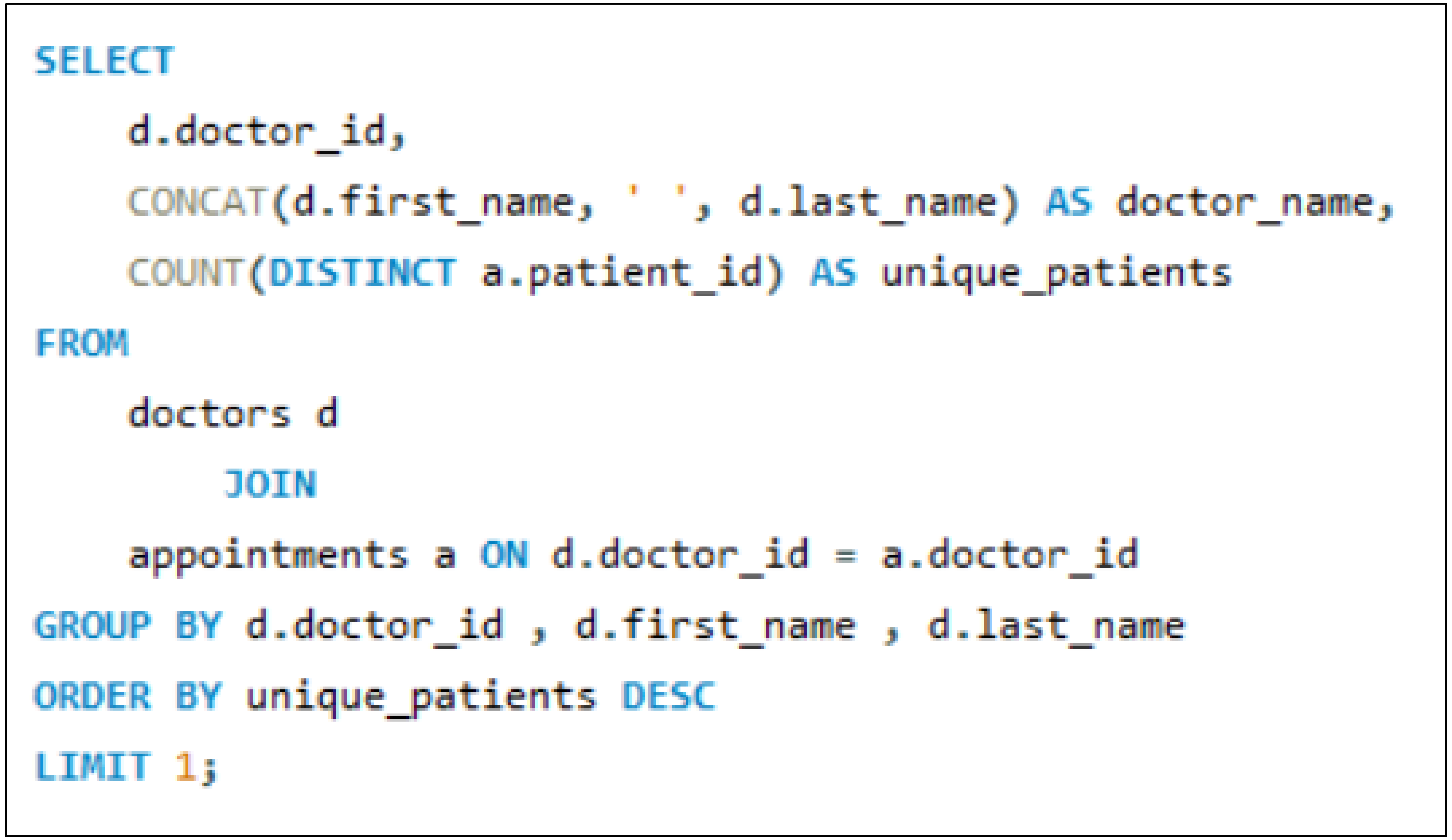


```
SELECT
    p.patient_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    MAX(a.appointment_date) AS last_visit_date
FROM
    patients p
        JOIN
    appointments a ON p.patient_id = a.patient_id
GROUP BY p.patient_id , p.first_name , p.last_name
ORDER BY last_visit_date DESC;
```

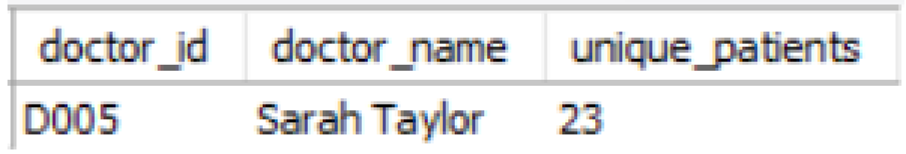


patient_id	patient_name	last_visit_date
P007	Alex Johnson	2024-12-30
P049	David Moore	2024-12-26
P012	Laura Davis	2024-12-26
P040	Emily Williams	2024-12-24
P019	Sarah Miller	2024-12-20
P023	Linda Johnson	2024-12-18
P039	Jane Wilson	2024-12-17
P016	Michael Taylor	2024-12-16
P029	David Smith	2024-12-14
P032	Alex Moore	2024-12-08

9. Which doctor treated the most unique patients?

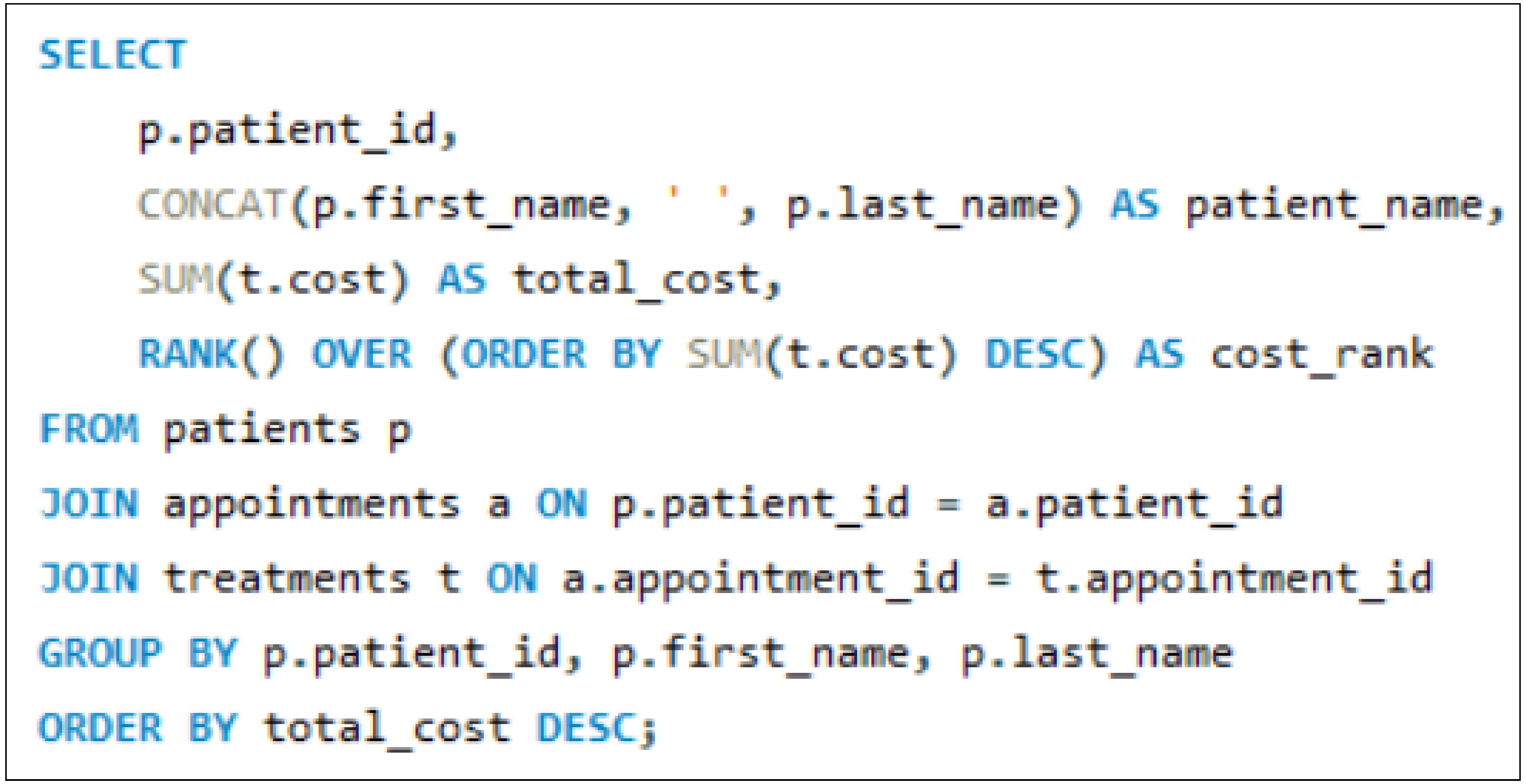


```
SELECT
    d.doctor_id,
    CONCAT(d.first_name, ' ', d.last_name) AS doctor_name,
    COUNT(DISTINCT a.patient_id) AS unique_patients
FROM
    doctors d
        JOIN
    appointments a ON d.doctor_id = a.doctor_id
GROUP BY d.doctor_id , d.first_name , d.last_name
ORDER BY unique_patients DESC
LIMIT 1;
```

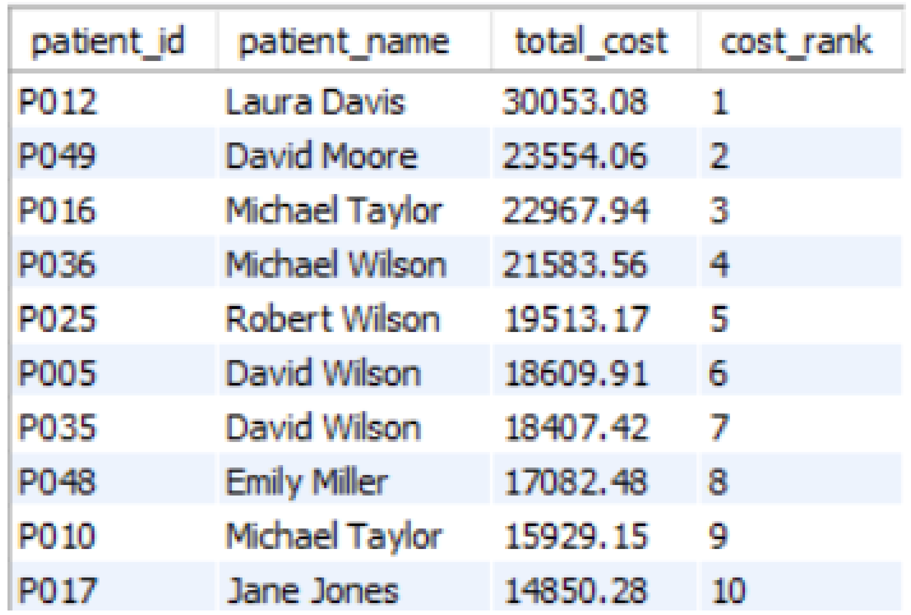


doctor_id	doctor_name	unique_patients
D005	Sarah Taylor	23

10. Who are the top 10 patients based on total treatment cost?



```
SELECT
    p.patient_id,
    CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
    SUM(t.cost) AS total_cost,
    RANK() OVER (ORDER BY SUM(t.cost) DESC) AS cost_rank
FROM patients p
JOIN appointments a ON p.patient_id = a.patient_id
JOIN treatments t ON a.appointment_id = t.appointment_id
GROUP BY p.patient_id, p.first_name, p.last_name
ORDER BY total_cost DESC;
```



patient_id	patient_name	total_cost	cost_rank
P012	Laura Davis	30053.08	1
P049	David Moore	23554.06	2
P016	Michael Taylor	22967.94	3
P036	Michael Wilson	21583.56	4
P025	Robert Wilson	19513.17	5
P005	David Wilson	18609.91	6
P035	David Wilson	18407.42	7
P048	Emily Miller	17082.48	8
P010	Michael Taylor	15929.15	9
P017	Jane Jones	14850.28	10

Business Outcomes & Analytical Observations

1. Consultation emerged as a common reason for visits, indicating its critical role in patient engagement.
2. Some doctors consistently handle more appointments, reflecting strong patient trust and operational demand.
3. A segment of patients contributes significantly to revenue, helping identify high-value individuals for premium care services.
4. Routine checkups and consultations dominate visit reasons, revealing preventive care as a major hospital focus.
5. Treatment data shows varied patient needs, with certain procedures more frequent and others cost-intensive.
6. The presence of failed and pending payments highlights gaps in billing processes needing optimization.
7. Revenue trends fluctuate across months, offering insights into seasonal or operational peaks in hospital activity.
8. Recent appointment tracking helps understand patient retention and follow-up effectiveness.
9. Specific doctors cater to a wider patient base, indicating potential for leadership or mentoring roles.
10. Treatment cost rankings enable financial segmentation of patients for strategic service offerings.

Conclusion

- Successfully built a Hospital Management System SQL project using 5 relational datasets and real-world queries.
- Extracted actionable business insights that support decision-making in healthcare operations.
- Applied advanced SQL concepts: JOINs, aggregations, subqueries, formatting, ranking functions.
- Strengthened my expertise in healthcare domain analytics using real data modeling.
- Enhanced my analytical thinking, problem-solving, and query optimization skills.

THANK YOU