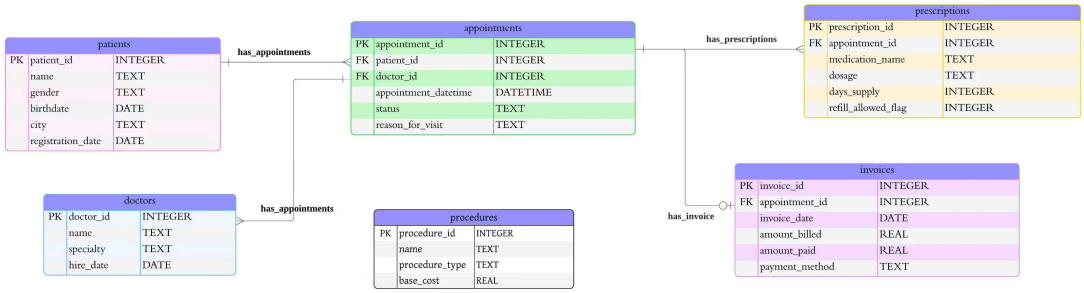
Task 1： ERD + DATA UNDERSTANDING

1.Reverse-Engineered ERD



2. Business Understanding Summary （One-Page Summary）

2.1 Domain Description – Healthcare Clinic Operations

The role of a clinic is to provide healthcare services to patients, a responsibility undertaken by medical professionals. These duties include scheduling appointments, making diagnoses, performing treatment procedures, issuing prescriptions, and maintaining accurate documentation.

To operate effectively, the clinic must adhere to standards of service quality, financial accountability, regulatory compliance, and efficient resource management.

Therefore, patients must register at the medical center and schedule appointments with relevant healthcare professionals. All scheduled appointments generate medical records. During visits, physicians may diagnose conditions, issue prescriptions, and order tests. Following treatment, the clinic issues detailed invoices listing medical records, consultation fees, and other related charges. These processes are supported by an integrated electronic system that stores all relevant information, generates reports, facilitates analysis, and supports operational management.

These activities rely on an integrated electronic information system that stores operational and patient data, generates reports, and supports managerial oversight. By maintaining a continuous digital record of patient interactions, provider activities, and financial outcomes, the system enables comprehensive visibility into operational history, performance metrics, staffing needs, and treatment results. This same data foundation also supports analytical use cases such as understanding no-show behavior, monitoring visit frequency, evaluating provider productivity, and analyzing revenue patterns across specialties. Domain-specific terminology used in this model includes: no-show (missed appointment with status='no_show'), revenue (sum of amount_paid), balance (amount_billed minus amount_paid), specialty (doctor’s field of practice), and procedure type (category of clinical service).

2.2 Key Business Processes Represented in the Data

（1）Patient Registration & Demographics

Patients enroll in the clinic by providing demographic information like name, gender, birth date and city. The registration date creates when a patient first established care.

This process supports patient population analysis, segmentation, and city demand level tracking.

（2）Provider Management

Doctors are recorded with their names, specialties, and hire dates.

Specialties enable analysis of service mix, capacity allocation, and provider productivity.

（3）Appointment Scheduling & Outcomes

The core operational process represented in the data.

Appointments link track by visit date and time, status (completed or canceled), reason for visit.

Appointment record help the clinic understand demand, no - show behavior, and scheduling efficiency.

（4）Prescriptions & Treatment Management

Doctors may issue prescriptions during an appointment. Prescription data record by medication name, dosage and duration, ad refill permissions.

This facilitates research on chronic disease management analysis and providers' prescribing practices.

（5）Billing & Payments

Each appointment generate an invoice, the billing data includes amount billed, amount paid and payment method.

This helps revenue analysis, outstanding balance tracking, and payment modality insights.

3. Data Dictionary

Table1: patients

Column	Data Type	Description	Relationship Hints
patient_id	integer (PK)	Unique identifier for each patient	Referenced by appointments, patient_id
name	text	Patient full name	
gender	text	Male/Female/Other	
birth date	numeric	Date of birth	Useful for age-based analyses
city	text	Patient's city of residence	Enables geographic segmentation
registration_date	numeric	First registration date with clinic	

Table2: doctors

Column	Data Type	Description	Relationship Hints
doctor_id	integer (PK)	Unique doctor ID	Referenced by appointments, patient_id
name	text	Doctor name	
specialty	text	Medical specialty	Enables specialty level analysis

Column	Data Type	Description	Relationship Hints
hire_date	numeric	Date doctor joined the clinic	Useful for tenure analysis

Table3: procedures

Column	Data Type	Description	Relationship Hints
procedure_id	integer (PK)	Unique ID for medical procedure	
name	text	Procedure name	
procedure_type	text	Type/category of procedure	
base_cost	real	Standard cost of procedure	Enables cost center analysis

Note: No table links procedures to appointments. It is included for domain completeness but not relational connected.

Table4: appointments

Column	Data Type	Description	Relationship Hints
appointment_id	integer (PK)	Unique ID for appointment	Referenced by prescriptions and invoices
patient_id	integer (FK)	Patient receiving the service	Links to patients.patient_id
doctor_id	integer (FK)	Provider conducting the visit	Links to doctors.doctor_id
appointment_datetime	numeric	Date/time of appointment	Useful for time-based analytics
status	text	Visit outcome (completed/canceled/no_show)	Tracks patient attendance behavior
reason_for_visit	text	Chief complaint or visit reason	Enables clinical/operational analysis

Table5: prescriptions

Column	Data Type	Description	Relationship Hints
prescription_id	integer (PK)	Unique prescription ID	
appointment_id	integer (FK)	Visit that generated prescription	Links to appointments.appointment_id
medication_name	text	Drug prescribed	
dosage	text	Dose instructions	
days_supply	integer	Days the medication is intended to last	
refill_allowed_flag	integer	0/1 indicator: whether refills allowed	Supports chronic condition analysis

Table6: invoices

Column	Data Type	Description	Relationship Hints
invoice_id	integer (PK)	Unique invoice ID	
appointment_id	integer (FK)	Appointment billed	Links to appointments.appointment_id
invoice_date	numeric	Billing date	
amount_billed	real	Amount charged	For revenue analysis
amount_paid	real	Actual payment received	Used to compute outstanding balances
payment_method	text	Cash/credit/insurance	Supports financial analytics

Table7:Domain-Specific Terminology

Term	Meaning
Appointment	Scheduled patient visit to a doctor
No-show	Patient did not attend a scheduled appointment
Specialty	Medical area in which a doctor practices
Procedure	Clinical service or treatment provided by the clinic

Term	Meaning
Prescription	Medication ordered during a visit
Invoice	Billing record for a medical service
Amount Billed	Total amount charged for a visit
Amount Paid	Payment received for the visit
Outstanding Balance	Amount billed minus paid
Refill Allowed	Whether patient may refill prescription without new appointment

3. Proposed 10 Business Questions

1. What are the monthly trends in appointment volume?

2. Which doctors handle the most appointments, and how does workload vary by specialty?

3. What is the appointment status distribution(completed/cancelled/no-show), and how does it vary by doctor or patient demographics?

4. Which cities have the highest number of registered patients and appointments?

5. What are the top reasons for visit(reason_for_visit), and how do they change over time?

6. Which medications are prescribed most frequently, and by which specialties/doctors?

7. How often are refills allowed, and does refill policy vary by medication or doctor?

8. What is the average bulled amount and paid amount per appointment, and what factors drive higher charges?

9. What is the unpaid balance rate(amount_billed - amount_paid), and how does it vary by payment method or doctor specialty?

10. How many appointments lead to prescriptions, and what patient or visit characteristics predict receiving a prescription?

Task 2 ANALYTICAL SQL QUERIES

1. Design 1 analytical question along with the solution in SQL.

Which payment methods bring in the most collected revenue?

Query:

```
3  ▶ ▼ SELECT
4          payment_method,
5          ROUND(SUM(amount_paid), 2) AS total_collected
6      FROM invoices
7      GROUP BY payment_method
8      ORDER BY total_collected DESC;
```

Screenshot of output:

3 rows • 255ms

payment_method	total_collected
card	7430551.88
cash	7398848.49
insurance	7385468.51

Interpretation of findings:
We found that card payments generated the highest total revenue (≈ $7.43M), slightly above cash (≈ $7.40M) and insurance (≈ $7.39M). The differences are small, but card payments consistently bring in the most collected revenue across all invoices.

Explanation of joins, filters, window logic:
The query only uses the invoices table, so no joins or window functions were needed. We simply grouped invoices by payment_method and summed amount_paid for each group.

Why the insight matters to the business:
This insight helps the clinic understand which payment method brings in revenue most consistently, allowing them to focus on streamlining that method to support faster and more reliable cash flow.

Analytical Questions

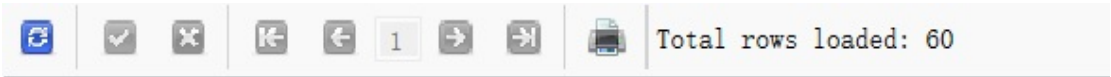
1. **Which doctors and time slots have the highest appointment no-show rates?**

(first we figure out which doctor have the highest appointment no-show rates)

Query:

```sql
SELECT
    d.name AS doctor_name,
    COUNT(*) AS total_appointments,
    SUM(CASE WHEN a.status = 'no_show' THEN 1 ELSE 0 END) AS no_show_count,
    ROUND(
        SUM(CASE WHEN a.status = 'no_show' THEN 1 ELSE 0 END) * 1.0
        / COUNT(*),
        3
    ) AS no_show_rate
FROM appointments a
JOIN doctors d
    ON a.doctor_id = d.doctor_id
GROUP BY d.name
ORDER BY no_show_rate DESC, total_appointments DESC;
```

Screenshot of output (first 10)



	doctor name	total appointments	no show coun	no show rate
1	Claudia Holland	159	14	0.088
2	Wayne Brown	172	13	0.076
3	Nicholas Dixon	189	14	0.074
4	Sara Rangel	241	17	0.071
5	Matthew Baker	301	21	0.07
6	Stuart Benson	228	16	0.07
7	Jose Thompson	186	13	0.07
8	Christina Schroeder	160	11	0.069
9	Samuel Williams Jr.	195	13	0.067
10	Allison Hunt	274	18	0.066

Interpretation of findings:

The results show that some doctors experience significantly higher patient no-show rates than others. The top 3 doctors are:

- Claudia Holland (8.8%)
- Wayne Brown (7.6%)
- Nicholas Dixon (7.4%)

Where Claudia Holland has the highest appointment no-show rate.

Explanation of joins, filters, window logic:

We joined appointments with doctors on doctor_id, and calculated:

- total appointments per doctor
- number of no-show appointments
- no-show rate = no_show_count / total_appointments

Doctors were ranked by highest no-show rate.

(then we figure out which time slots have the highest appointment no-show rates)

Query:

```sql
SELECT
    strftime('%H:00', a.appointment_datetime) AS time_slot,
    COUNT(*) AS total_appointments,
    SUM(CASE WHEN a.status = 'no_show' THEN 1 ELSE 0 END) AS no_show_count,
    ROUND(
        SUM(CASE WHEN a.status = 'no_show' THEN 1 ELSE 0 END) * 1.0
        / COUNT(*),
        3
    ) AS no_show_rate
FROM appointments a
GROUP BY time_slot
ORDER BY no_show_rate DESC, total_appointments DESC;
```

Screenshot of output:

	time slot	total appointments	no show coun	no show rate
1	08:00	1382	77	0.056
2	17:00	1335	73	0.055
3	16:00	1350	70	0.052
4	11:00	1297	66	0.051
5	12:00	1391	70	0.05
6	15:00	1395	69	0.049
7	13:00	1349	66	0.049
8	09:00	1329	64	0.048
9	14:00	1374	59	0.043
10	10:00	1378	58	0.042

Interpretation of findings:

No-show rates also vary by appointment hour. The highest no-show time periods include:

- 08:00 (5.6%)
- 17:00 (5.5%)
- 16:00 (5.2%)

Where 08:00 has the highest appointment no-show rate.

Explanation of joins, filters, window logic:
We extracted the hour of each appointment using: strftime('%H:00', appointment_datetime)

Then we computed:

- total appointments in each time slot
- number of no-shows
- no-show rate for each time slot

Time slots were ranked by highest no-show rate.

Why the insight matters to the business:
Understanding which doctors and time slots experience the highest no-show rates is essential for improving resource allocation. High no-show periods lead to idle staff time and reduced revenue for other patients. By identifying specific doctors and hours where no-shows are more frequent, the clinic can implement targeted interventions such as reminder messages or overbooking strategies. These insights help the clinic maximize schedule utilization and reduce operational waste.

2. **Which medical specialties generate the most revenue?**

Query:

```sql
SELECT
    d.specialty,
    SUM(i.amount_billed) AS total_revenue,
    COUNT(i.invoice_id) AS total_invoices
FROM invoices i
JOIN appointments a
    ON i.appointment_id = a.appointment_id
JOIN doctors d
    ON a.doctor_id = d.doctor_id
GROUP BY d.specialty
ORDER BY total_revenue DESC;
```

Screenshot of output:

	specialty	total_revenue	total_invoices
1	Cardiology	9400234.55	1796
2	Pediatrics	4847173.02	1818
3	Psychiatry	3840077.44	1819
4	Orthopedics	2947908.76	1884
5	Dermatology	2257121.14	1809
6	GP	1401248.09	1745

Interpretation of findings:
The top 3 specialty which generate the most revenue include:
Cardiology (9400234.55 revenue)
Pediatrics (4847173.02 revenue)
Psychiatry (3840077.44 revenue)

Where Cardiology generates the highest revenue (~$9.4M).

Explanation of joins, filters, window logic:
To determine which medical specialties generate the most revenue,we joined the invoices table with appointments and then joined appointments with doctors
.
Revenue was measured using amount_billed from the invoices table.

We grouped the results by each doctor's specialty and summed all billed amounts to obtain total revenue per specialty.

Why the insight matters to the business:
Identifying which medical specialties generate the most revenue enables the clinic to make more operational decisions. High-revenue specialties such as Cardiology and Pediatrics contribute disproportionately to the clinic's financial performance, meaning they may warrant additional resources or staffing availability. Conversely, lower-revenue specialties highlight potential areas for investment in efficiency improvements.
Understanding revenue contribution across specialties allows leadership to allocate budgets more effectively and focus growth initiatives on areas with the strongest financial impact.

3. **Which patients visit the clinic most frequently?**

Query:

```sql
SELECT
    p.name AS patient_name,
    COUNT(a.appointment_id) AS total_visits
FROM appointments a
JOIN patients p
    ON a.patient_id = p.patient_id
GROUP BY p.name
ORDER BY total_visits DESC;
```

Screenshot of output (first 10)

	patient_name	total_visits
1	Thomas Byrd	76
2	Jennifer Brown	52
3	Christina Lutz	52
4	Anita Abbott	50
5	Mark Rhodes	47
6	Gabrielle Pena	47
7	Victoria Vargas	46
8	Victor Hayes	46
9	Ronald Rogers	46
10	Meagan Conway	46

Interpretation of findings:
The top 3 most frequent patients are:
Thomas Byrd (76)
Jennifer Brown (52)
Christina Lutz (52)

Where Thomas Byrd is the most frequent visitor with 76 total appointments

Explanation of joins, filters, window logic:
We grouped all appointment records by patient_id to calculate the total number of visits for each patient:COUNT(*) AS total_visits

We then joined the patients table to retrieve the patient names, enabling us to report results in a meaningful and readable format.

Finally, we sorted the results in descending order of visit count to identify the clinic's most frequent visitors.

Why the insight matters to the business:
Identifying the patients who visit the clinic most frequently helps the clinic understand which individuals rely heavily on its services and may require more proactive care. High-frequency visitors often indicate chronic conditions. By recognizing these patients early, the clinic can provide more structured follow-up and improve resource allocation. This insight also allows the clinic to design personalized outreach programs that enhance patient satisfaction and long-term retention.

4. **What are the most common procedures performed across specialties?**

Query:

```
SELECT
    substr(p.name, 1, instr(p.name, ' procedure') - 1) AS specialty,
    p.procedure_type,
    COUNT(*) AS num_procedures
FROM procedures p
GROUP BY specialty, p.procedure_type
ORDER BY specialty, num_procedures DESC;
```

Screenshot of output (first 10)

	specialty	procedure type	num procedure
1	Cardiology	surgery	3
2	Dermatology	lab	3
3	Dermatology	consultation	2
4	Dermatology	surgery	1
5	GP	lab	3
6	GP	surgery	2
7	GP	consultation	2
8	Orthopedics	consultation	4
9	Orthopedics	surgery	2
10	Pediatrics	surgery	4
11	Pediatrics	lab	2
12	Pediatrics	consultation	1
13	Psychiatry	surgery	1
14	Psychiatry	lab	1
15	Psychiatry	consultation	1

Interpretation of findings:

The output shows patterns in the types of procedures each specialty most commonly performs:

- Cardiology most frequently performs surgery.
- Dermatology primarily performs lab tests and consultations.
- General Practice shows a balanced mix of lab tests, surgery, and consultations, with lab tests being the most common.
- Orthopedics mainly conducts consultations and surgeries.
- Pediatrics frequently performs surgeries, followed by lab tests.
- Psychiatry primarily conducts consultations, with occasional lab tests and

surgeries.

Explanation of joins, filters, window logic:
Here we use the procedures table as the source of procedure information and derive the specialty from the procedure name.

Then, group by specialty and procedure_type and count how many procedures each specialty offers in each category.

Finally, we sort by specialty and num_procedures in descending order to see which procedure types are most common within each specialty.

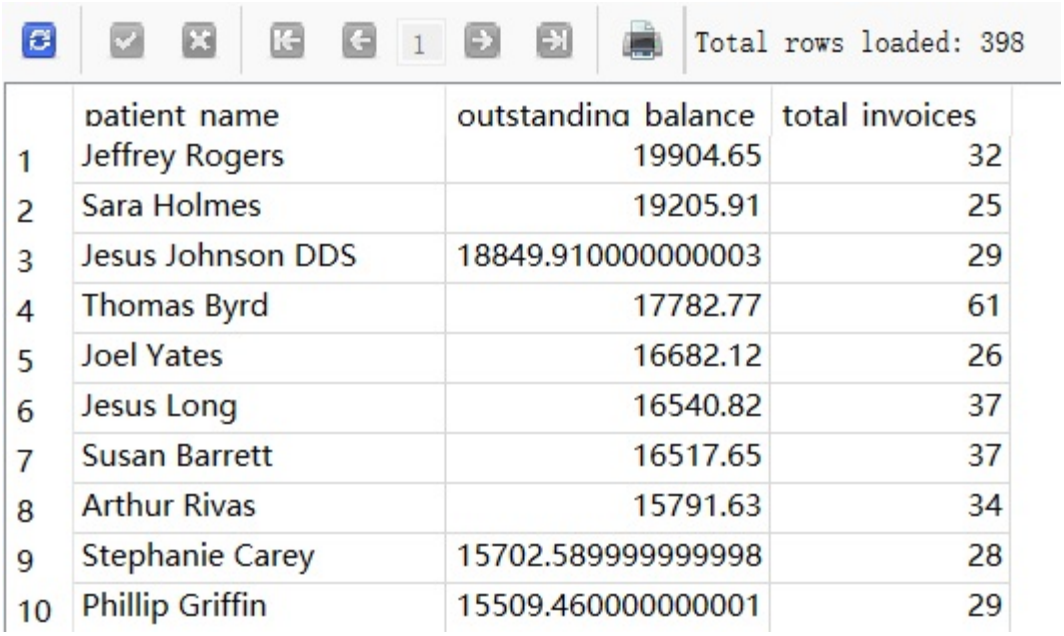
Why the insight matters to the business:
Understanding which procedures are most common within each specialty helps the clinic better plan staffing, and equipment needs. Specialties that frequently perform lab tests, consultations, or surgeries require different operational support, and recognizing these patterns allows the clinic to align capacity with actual demand. This insight also helps identify high-volume procedure types that may benefit from workflow optimization. Overall, knowing which procedures dominate each specialty enables more efficient service delivery, and smarter long-term planning for clinical operations.

5. **Which patients have outstanding balances and how large are they?**

Query:

```sql
SELECT
    p.name AS patient_name,
    SUM(i.amount_billed - i.amount_paid) AS outstanding_balance,
    COUNT(i.invoice_id) AS total_invoices
FROM invoices i
JOIN appointments a
    ON i.appointment_id = a.appointment_id
JOIN patients p
    ON a.patient_id = p.patient_id
GROUP BY p.name
HAVING outstanding_balance > 0
ORDER BY outstanding_balance DESC;
```

Screenshot of output (first 10)



Total rows loaded: 398

	patient_name	outstanding_balance	total_invoices
1	Jeffrey Rogers	19904.65	32
2	Sara Holmes	19205.91	25
3	Jesus Johnson DDS	18849.910000000003	29
4	Thomas Byrd	17782.77	61
5	Joel Yates	16682.12	26
6	Jesus Long	16540.82	37
7	Susan Barrett	16517.65	37
8	Arthur Rivas	15791.63	34
9	Stephanie Carey	15702.589999999998	28
10	Phillip Griffin	15509.460000000001	29

Interpretation of findings:
The top 3 patients with most outstanding balances are:
Jeffrey Rogers (19904.65)
Sara Holmes (19205.91)
Jesus Johnson DDS (18849.91)

The results show that patient Jeffrey Rogers (~19,904.65) has the largest unpaid balances.

Explanation of joins, filters, window logic:
To identify patients with outstanding balances, we joined the invoices table with appointments to obtain patient_id, and then joined patients to retrieve patient names.

The outstanding balance for each invoice was calculated as: amount_billed - amount_paid

We aggregated these balances at the patient level and filtered for patients whose total outstanding amount is greater than zero.

Why the insight matters to the business:
Identifying patients with outstanding balances is essential for improving the clinic's cash-flow management. By understanding which patients consistently carry unpaid amounts and how large those balances are, the clinic can prioritize outreach and reduce revenue leakage. Clear visibility into unpaid invoices also allows the clinic to design more effective billing policies and streamline administrative follow-up. Ultimately, this insight helps ensure that services delivered translate into collected revenue, supporting both operational sustainability and future investment in patient care.

Window Function Questions

6. **What is the number of days between each patient's consecutive visits?**

Query:

```sql
SELECT
    a.patient_id,
    p.name AS patient_name,
    a.appointment_datetime,
    LAG(a.appointment_datetime) OVER (
        PARTITION BY a.patient_id
        ORDER BY a.appointment_datetime
    ) AS previous_visit,
    ROUND(
        julianday(a.appointment_datetime) -
        julianday(
            LAG(a.appointment_datetime) OVER (
                PARTITION BY a.patient_id
                ORDER BY a.appointment_datetime
            )
        ),
        1
    ) AS days_since_last_visit
FROM appointments a
JOIN patients p
    ON a.patient_id = p.patient_id
ORDER BY a.patient_id, a.appointment_datetime;
```

Screenshot of output (first 10)

13580 rows • 2s

patient_id	patient_name	appointment_datetime	previous_visit	days_since_last_visit
1	Austin Nelson	2024-06-18 15:03:05	NULL	NULL
1	Austin Nelson	2024-07-20 09:56:22	2024-06-18 15:03:05	31.8
1	Austin Nelson	2024-08-11 08:17:02	2024-07-20 09:56:22	21.9
1	Austin Nelson	2024-08-18 10:24:48	2024-08-11 08:17:02	7.1
1	Austin Nelson	2024-08-26 08:25:47	2024-08-18 10:24:48	7.9
1	Austin Nelson	2024-09-03 12:12:15	2024-08-26 08:25:47	8.2
1	Austin Nelson	2024-09-21 08:21:42	2024-09-03 12:12:15	17.8
1	Austin Nelson	2024-11-27 11:03:12	2024-09-21 08:21:42	67.1
1	Austin Nelson	2025-01-07 14:14:23	2024-11-27 11:03:12	41.1
1	Austin Nelson	2025-01-18 16:59:18	2025-01-07 14:14:23	11.1

Interpretation of findings:
We found most patients return within 1–40 days between visits, with occasional long gaps over 50+ days, showing mixed follow-up consistency.

Explanation of joins, filters, window logic:
We joined appointments with patients to add patient names and used LAG () OVER (PARTITION BY patient_id ORDER BY appointment_datetime) to capture each patient's previous visit and calculate the day gap.

Why the insight matters to the business:
This insight helps us knowing visit gaps helps the clinic identify patients who delay follow-ups so staff can improve scheduling and follow-up reminders.

7. **What is the revenue rank of each doctor?**

Query:

```
205 ▶ ▼ WITH doctor_revenue AS (
206         SELECT
207             d.doctor_id,
208             d.name AS doctor_name,
209             ROUND(SUM(i.amount_billed), 2) AS total_revenue
210         FROM invoices i
211         JOIN appointments a
212             ON i.appointment_id = a.appointment_id
213         JOIN doctors d
214             ON a.doctor_id = d.doctor_id
215         GROUP BY d.doctor_id, doctor_name
216     )
217     SELECT
218         doctor_id,
219         doctor_name,
220         total_revenue,
221         RANK() OVER (ORDER BY total_revenue DESC) AS revenue_rank
222     FROM doctor_revenue
223     ORDER BY revenue_rank;
```

Screenshot of output (first 10)

60 rows • 1s

doctor_id	doctor_name	total_revenue	revenue_rank
23	Sarah Cobb	766193.21	1
37	Megan Smith	746200.96	2
15	Teresa Johnston	705854.69	3
11	Stephanie Andrews	689762.33	4
48	Kimberly Hardin	686646.13	5
21	Kevin Abbott	679852.1	6
31	John Spencer	678919.89	7
59	Julie Horn	673783.49	8
38	John Gonzales	663062.06	9
39	Christina Schroeder	650415.88	10

Interpretation of findings:
The results show each doctor's total revenue and rank, with the top earners (e.g., Sarah Cobb ≈ $766K, Megan Smith ≈ $747K) significantly ahead of others.

Explanation of joins, filters, window logic:
We joined invoices → appointments → doctors to link billed amounts to the correct doctor, then grouped totals per doctor. Finally, we applied RANK() to order doctors by total revenue.

Why the insight matters to the business:
This ranking helps the clinic identify high-impact doctors and allocate resources or support where it drives the most financial value.