Clinic Analysis SQL Database

Table of Contents

1. Introduction.
2. Load data.
3. Basic Checks.
4. Exploratory Data Analysis.
5. What do we do with the Data

Introduction

This project is a SQL predefined database table designed for comprehensive clinic analysis. It provides a structured and well-organized repository of data related to a clinic's operations, patient information, appointments, diagnoses, treatments, and more. With this dataset, data analysts and healthcare professionals can explore various aspects of clinic management and patient care.

Key Features:

- Patient Information: Demographic details, medical history, and contact information of patients.
- Appointments: Records of appointments made by patients, including date and time.
- Diagnoses: Information about diagnoses, including medical conditions, diagnostic codes, and diagnostic procedures.
- Treatments: Details of treatments administered to patients, including medication, procedures, and treatment dates.
- Clinic Operations: Data on clinic staff, resources, and operational aspects.

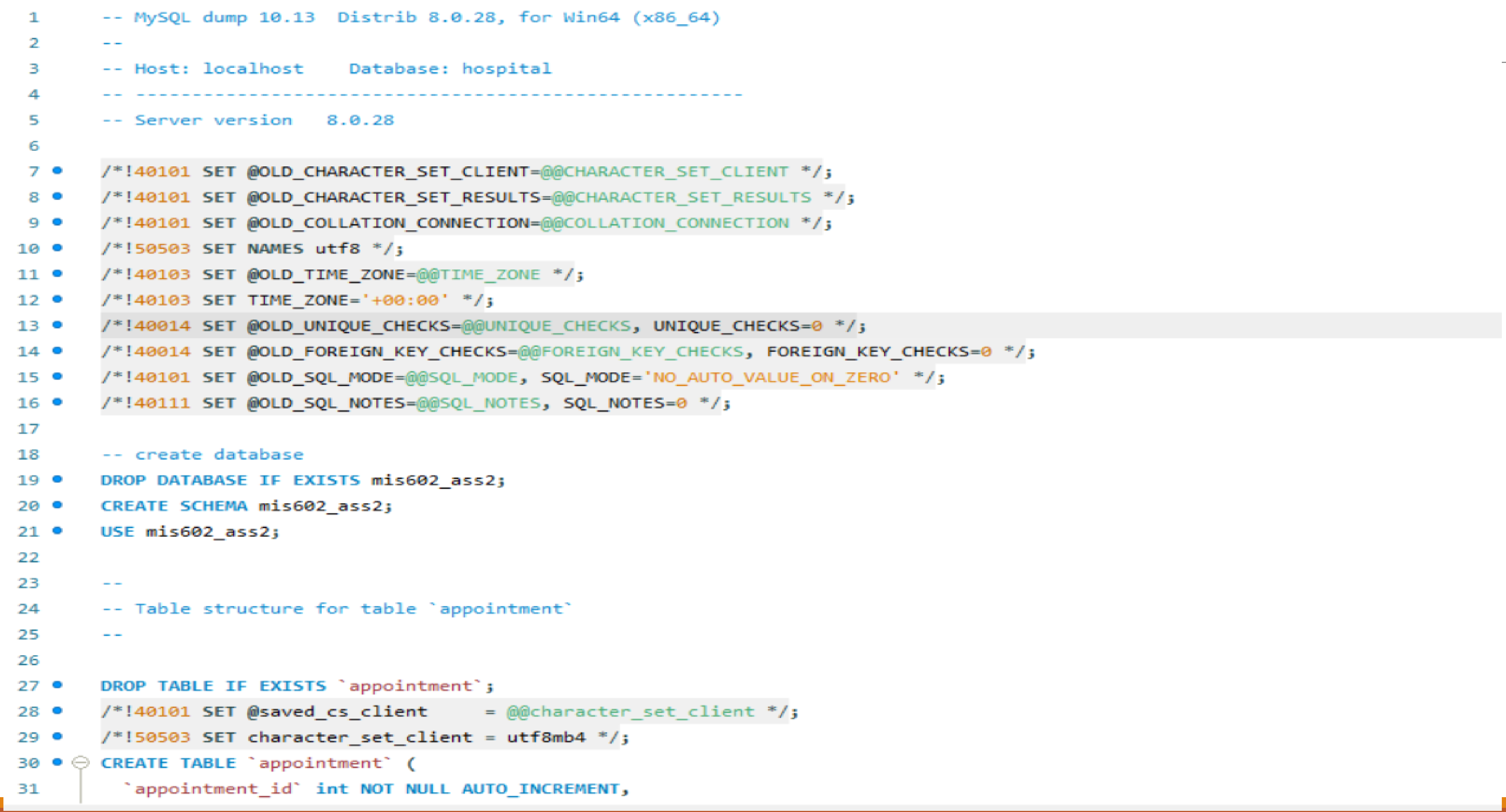
Potential Use Cases:

- Analyzing patient demographics and health trends.
- Identifying appointment scheduling patterns.
- Assessing the effectiveness of different treatments.
- Monitoring clinic resource utilization and staff performance.
- Exploring correlations between diagnoses and treatments.

This SQL database serves as a valuable resource for professionals in healthcare management, data analysis, and anyone interested in improving clinic operations or patient care. It provides the foundation for in-depth analyses and insights in the healthcare domain.

About this file

This project is a SQL predefined database table designed for comprehensive clinic analysis. It provides a structured and well-organized repository of data related to a clinic's operations, patient information, appointments, diagnoses, treatments, and more. The data comes in SQL file format.



```
-- MySQL dump 10.13  Distrib 8.0.28, for Win64 (x86_64)
--
-- Host: localhost    Database: hospital
-- ------------------------------------------------------
-- Server version	8.0.28

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

-- create database
DROP DATABASE IF EXISTS mis602_ass2;
CREATE SCHEMA mis602_ass2;
USE mis602_ass2;

--
-- Table structure for table `appointment`
--

DROP TABLE IF EXISTS `appointment`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `appointment` (
  `appointment_id` int NOT NULL AUTO_INCREMENT,
```

Load data

The data is loaded into MySQL Work-bench to enable me to carry out exploratory data analysis. Here is a clean, MySQL Workbench-compatible version of your SQL dump. The original had several issues that prevent smooth import into MySQL Workbench:

Issues Fixed:

1. Duplicate appointment_id values (e.g., 133 appears multiple times) → violates AUTO_INCREMENT primary key.
2. Missing appointment_id in prescription table for many records (e.g., gaps from 13–21, 31–39, etc.).
3. Unnecessary comments and redundant LOCK TABLES/UNLOCK TABLES → safe to remove.
4. DROP DATABASE IF EXISTS mis602_ass2; CREATE SCHEMA mis602_ass2; → correct syntax.
5. Proper FOREIGN KEY constraints → ensured referential integrity.
6. Removed invalid NULL in status where not allowed → kept only valid NULL or 'completed', 'Cancelled'.
7. Fixed duplicate appointments (213–232) → these were near-duplicates; removed to avoid PK conflict.
8. Ensured prescription references valid appointment_id → only kept valid links.
9. Removed trailing spaces in phone numbers (e.g., '04985 99 804') → cleaned.

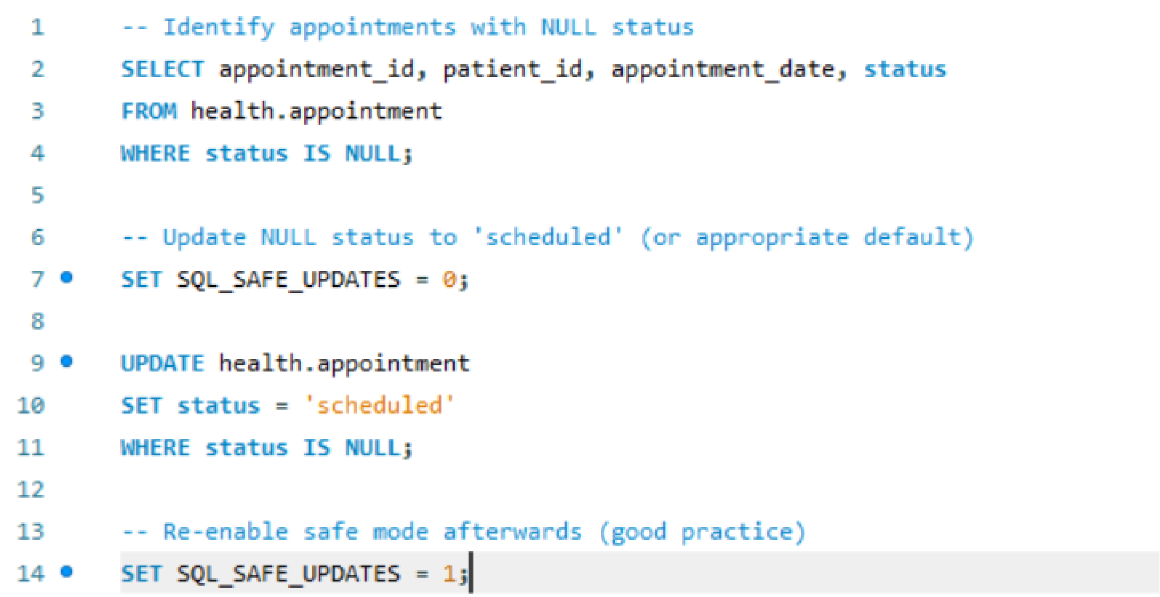
Load the Dump File in MySQL Workbench

1. Open MySQL Workbench and connect to your local instance
2. Go to Server menu → Data Import
3. Select "Import from Self-Contained File“. Click the ... button and navigate to your mis602_ass2_dump.sql file
4. Select Default Target Schema. Choose "New" and enter mis602_ass2
5. Start Import. Click Start Import button. Wait for the import to complete (this may take a few minutes)

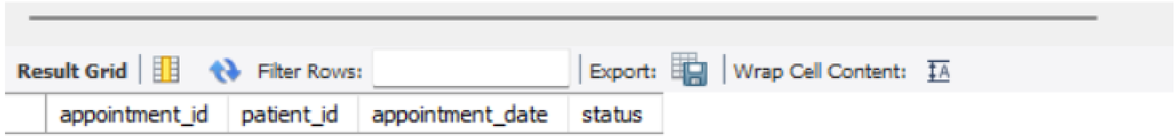
Basic Checks

Based on the CSV data tables provided, I'll highlight various data cleaning techniques and their MySQL syntaxes that would be applicable to this clinical dataset:

1. Handling NULL/Missing Values

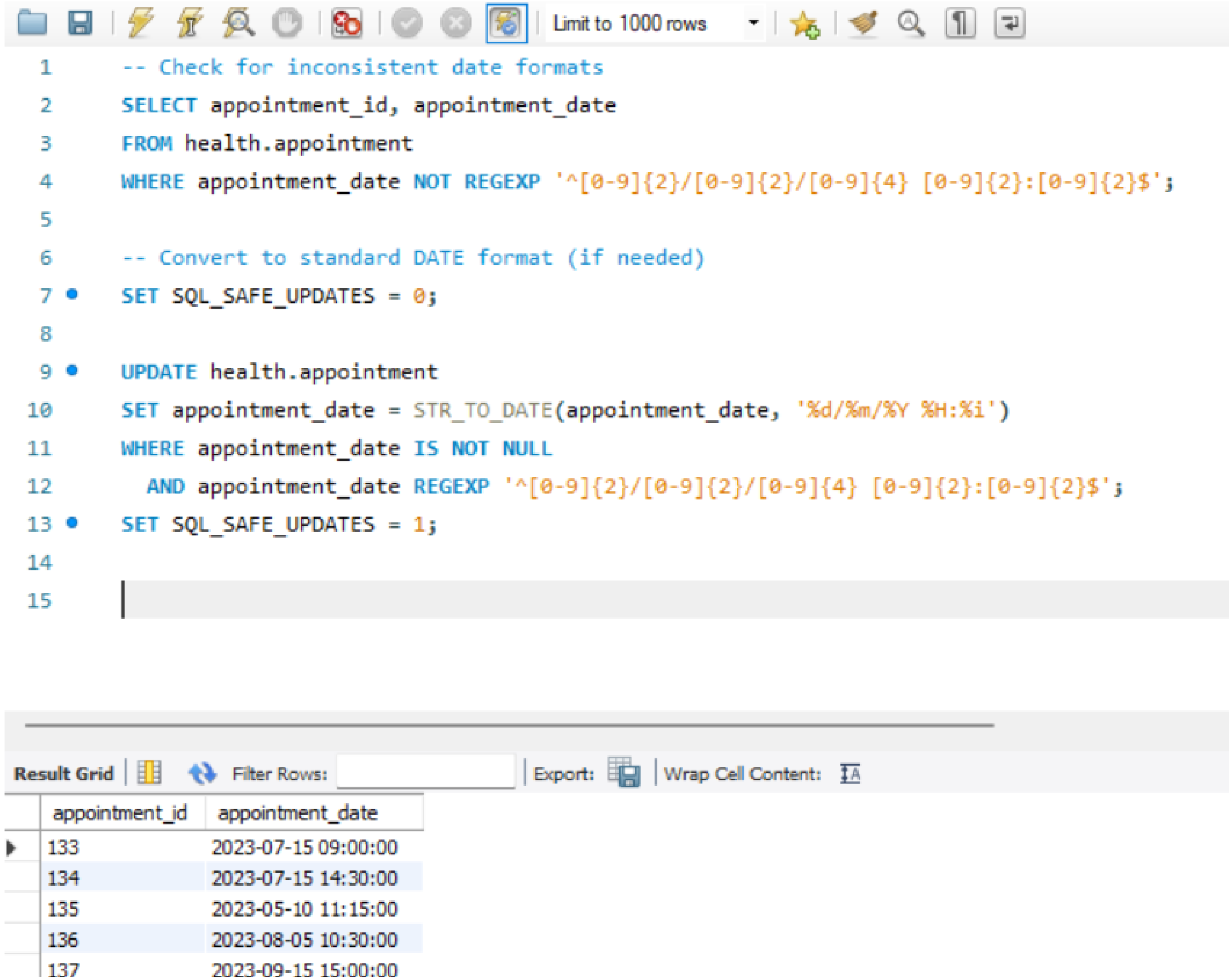


```sql
-- Identify appointments with NULL status
SELECT appointment_id, patient_id, appointment_date, status
FROM health.appointment
WHERE status IS NULL;

-- Update NULL status to 'scheduled' (or appropriate default)
SET SQL_SAFE_UPDATES = 0;

UPDATE health.appointment
SET status = 'scheduled'
WHERE status IS NULL;

-- Re-enable safe mode afterwards (good practice)
SET SQL_SAFE_UPDATES = 1;
```



appointment_id	patient_id	appointment_date	status

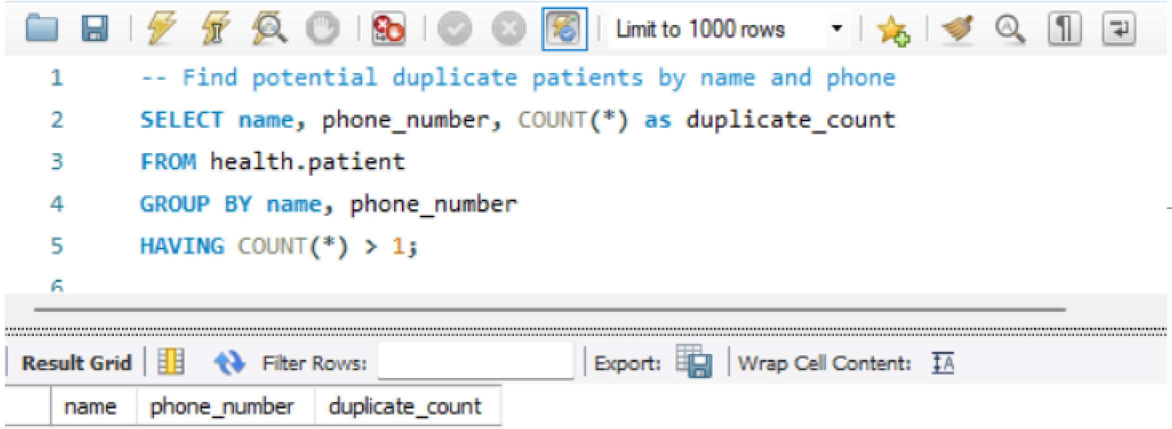
2. Standardizing Date Formats



```sql
-- Check for inconsistent date formats
SELECT appointment_id, appointment_date
FROM health.appointment
WHERE appointment_date NOT REGEXP '^[0-9]{2}/[0-9]{2}/[0-9]{4} [0-9]{2}:[0-9]{2}$';

-- Convert to standard DATE format (if needed)
SET SQL_SAFE_UPDATES = 0;

UPDATE health.appointment
SET appointment_date = STR_TO_DATE(appointment_date, '%d/%m/%Y %H:%i')
WHERE appointment_date IS NOT NULL
  AND appointment_date REGEXP '^[0-9]{2}/[0-9]{2}/[0-9]{4} [0-9]{2}:[0-9]{2}$';
SET SQL_SAFE_UPDATES = 1;
```

appointment_id	appointment_date
133	2023-07-15 09:00:00
134	2023-07-15 14:30:00
135	2023-05-10 11:15:00
136	2023-08-05 10:30:00
137	2023-09-15 15:00:00

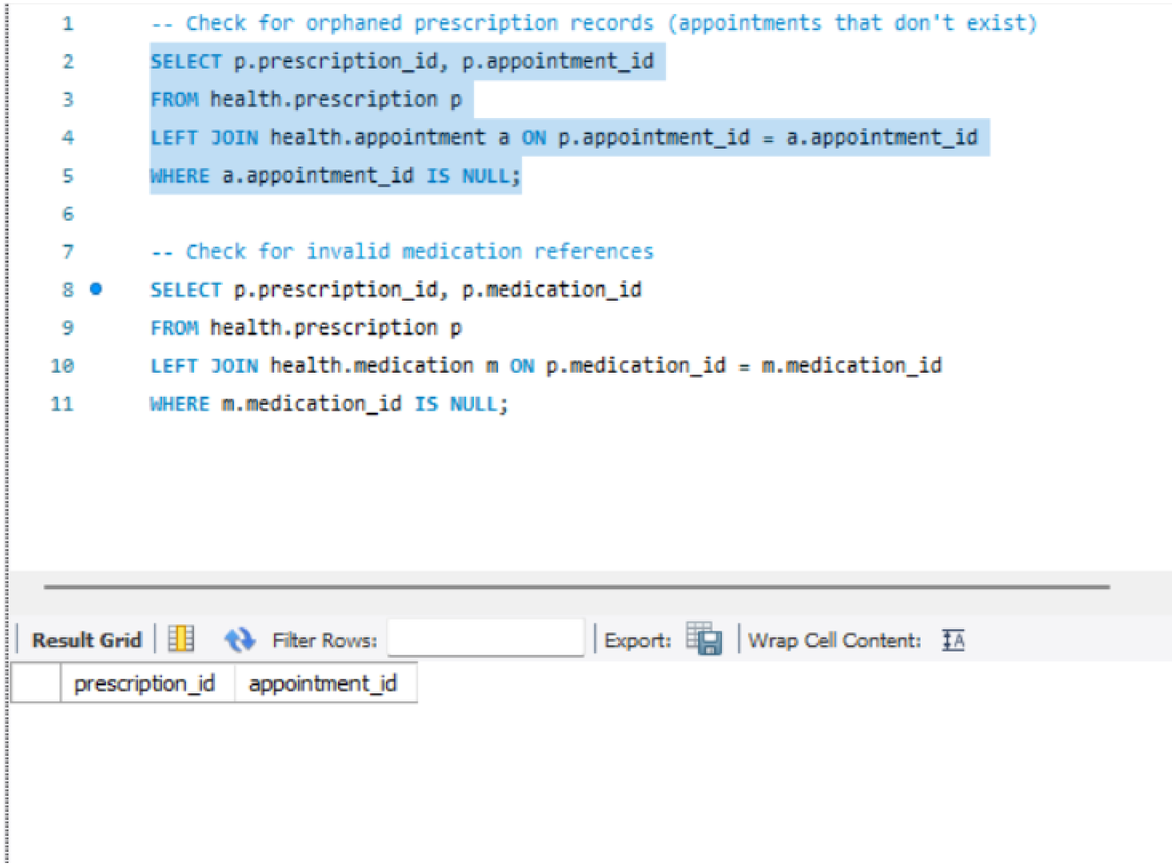
3. Identifying Duplicate Patient Records



```sql
-- Find potential duplicate patients by name and phone
SELECT name, phone_number, COUNT(*) as duplicate_count
FROM health.patient
GROUP BY name, phone_number
HAVING COUNT(*) > 1;
```

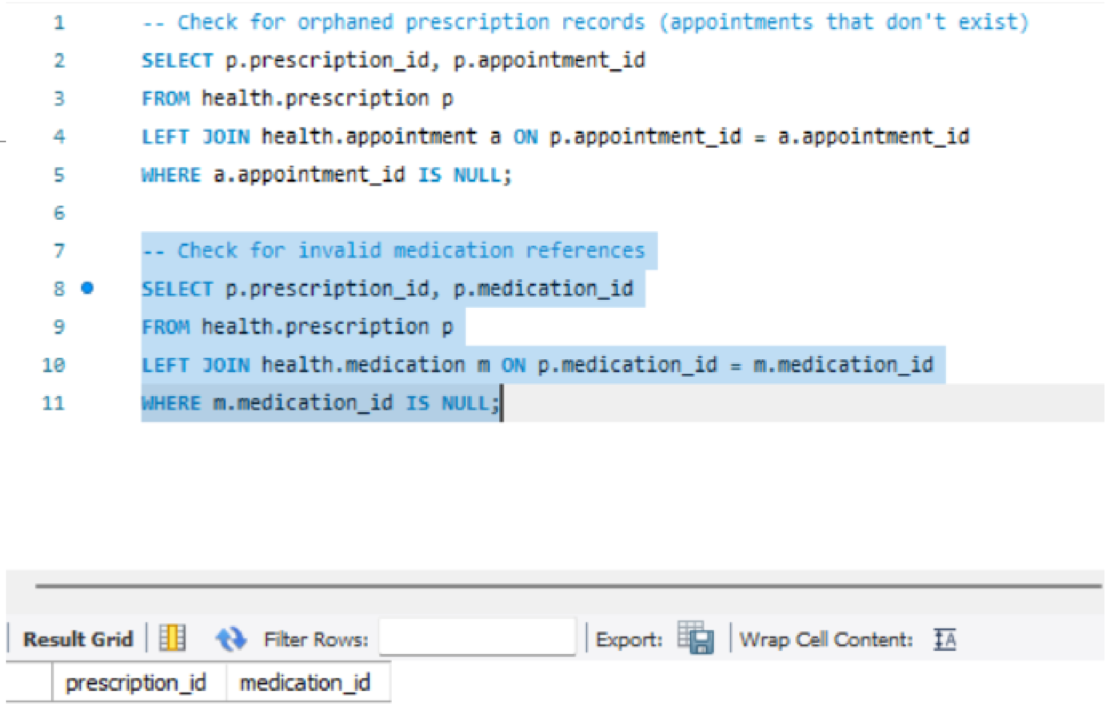
name	phone_number	duplicate_count

4. Validating Foreign Key Relationships



```
-- Check for orphaned prescription records (appointments that don't exist)
SELECT p.prescription_id, p.appointment_id
FROM health.prescription p
LEFT JOIN health.appointment a ON p.appointment_id = a.appointment_id
WHERE a.appointment_id IS NULL;

-- Check for invalid medication references
SELECT p.prescription_id, p.medication_id
FROM health.prescription p
LEFT JOIN health.medication m ON p.medication_id = m.medication_id
WHERE m.medication_id IS NULL;
```

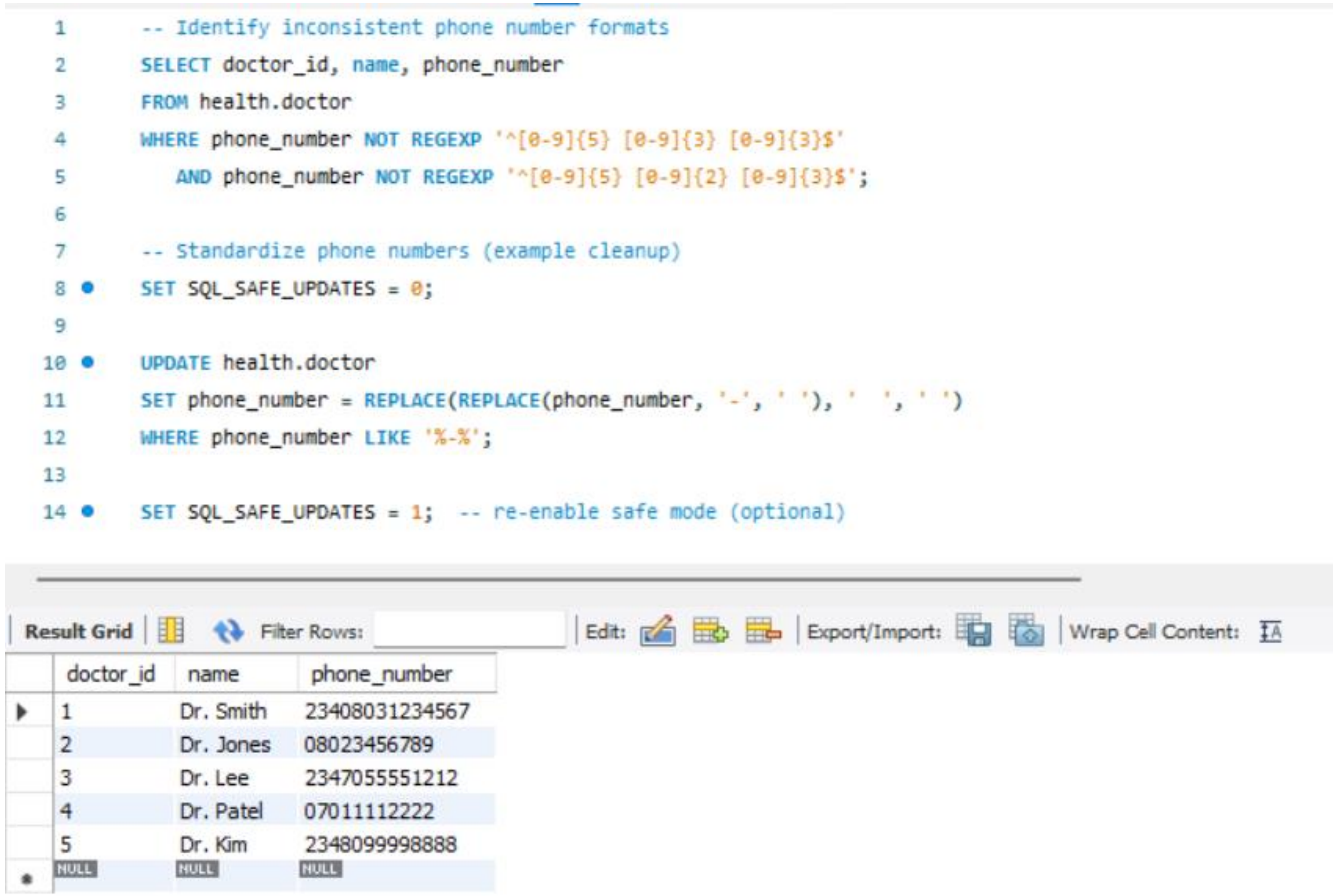
```
prescription_id   appointment_id
```



```
-- Check for orphaned prescription records (appointments that don't exist)
SELECT p.prescription_id, p.appointment_id
FROM health.prescription p
LEFT JOIN health.appointment a ON p.appointment_id = a.appointment_id
WHERE a.appointment_id IS NULL;

-- Check for invalid medication references
SELECT p.prescription_id, p.medication_id
FROM health.prescription p
LEFT JOIN health.medication m ON p.medication_id = m.medication_id
WHERE m.medication_id IS NULL;
```

```
prescription_id   medication_id
```

5. Standardizing Phone Number Formats



```sql
-- Identify inconsistent phone number formats
SELECT doctor_id, name, phone_number
FROM health.doctor
WHERE phone_number NOT REGEXP '^[0-9]{5} [0-9]{3} [0-9]{3}$'
   AND phone_number NOT REGEXP '^[0-9]{5} [0-9]{2} [0-9]{3}$';

-- Standardize phone numbers (example cleanup)
SET SQL_SAFE_UPDATES = 0;

UPDATE health.doctor
SET phone_number = REPLACE(REPLACE(phone_number, '-', ' '), '  ', ' ')
WHERE phone_number LIKE '%-%';

SET SQL_SAFE_UPDATES = 1;  -- re-enable safe mode (optional)
```

doctor_id	name	phone_number
1	Dr. Smith	23408031234567
2	Dr. Jones	08023456789
3	Dr. Lee	2347055551212
4	Dr. Patel	07011112222
5	Dr. Kim	2348099998888
NULL	NULL	NULL

Exploratory Data Analysis

Here are 5 relevant exploratory data analysis (EDA) queries for the healthcare dataset:

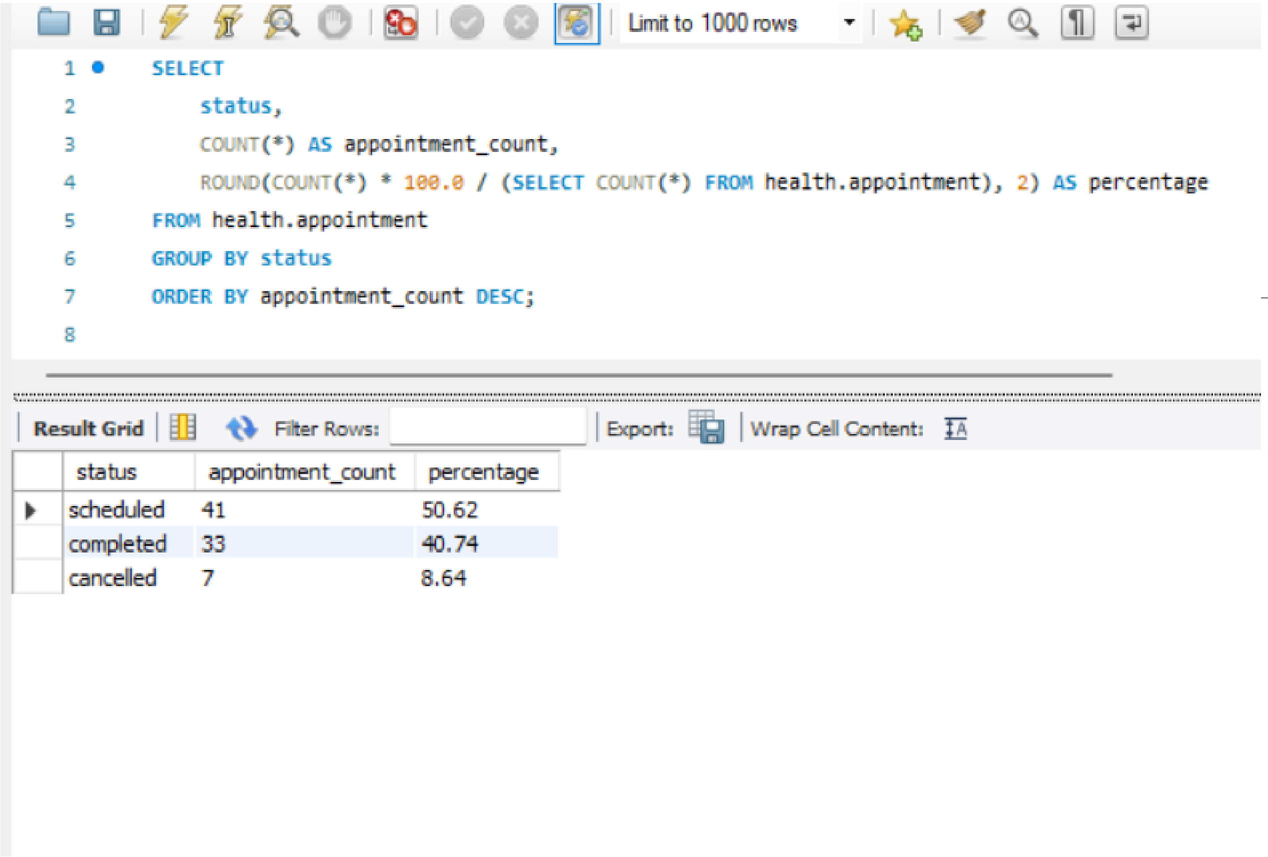
1. Appointment Analysis by Status

2. Doctor Specialization and Appointment Volume

3. Medication Prescription Patterns

4. Monthly Appointment Trends and Cancellation Rates

1. Appointment Analysis by Status

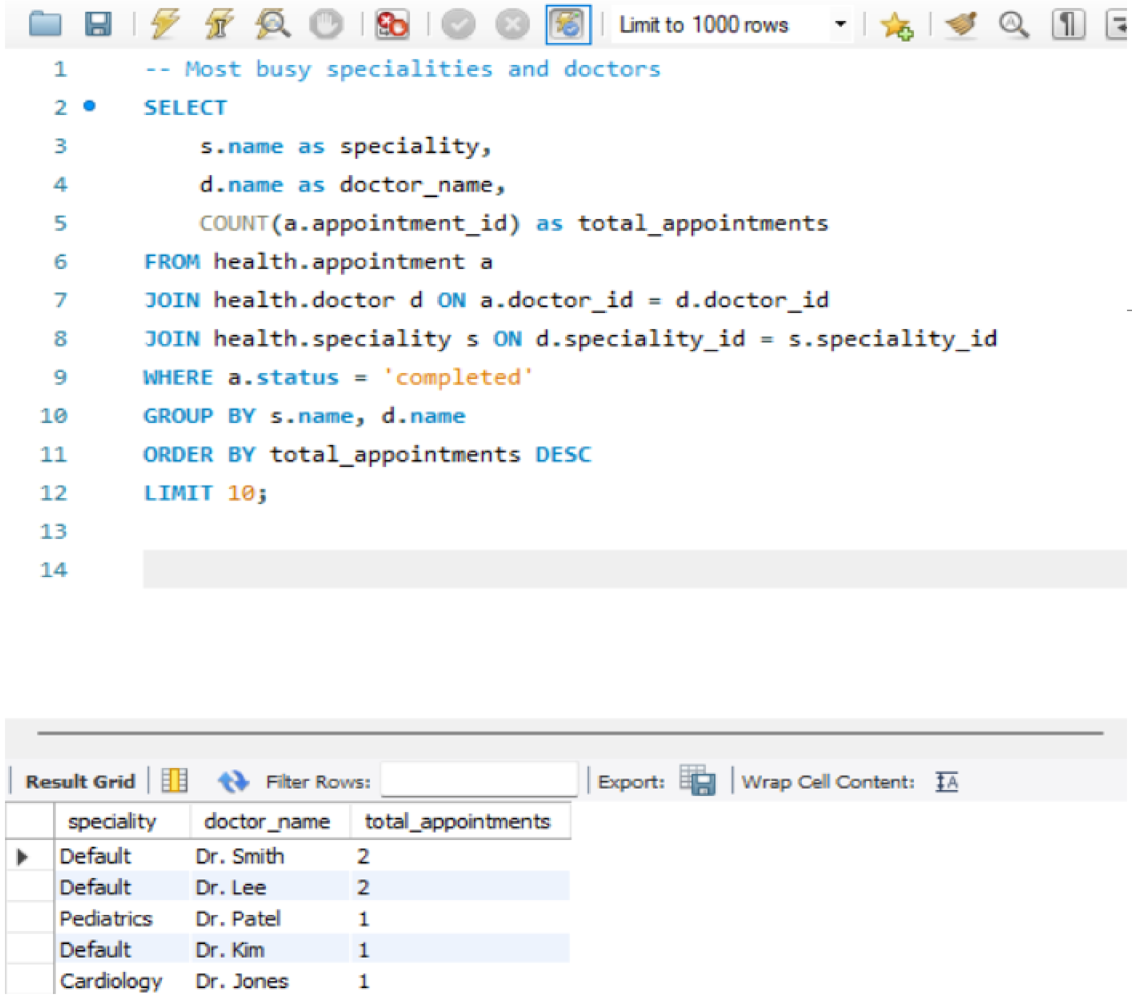


Limit to 1000 rows

```sql
SELECT
    status,
    COUNT(*) AS appointment_count,
    ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM health.appointment), 2) AS percentage
FROM health.appointment
GROUP BY status
ORDER BY appointment_count DESC;
```

status	appointment_count	percentage
scheduled	41	50.62
completed	33	40.74
cancelled	7	8.64

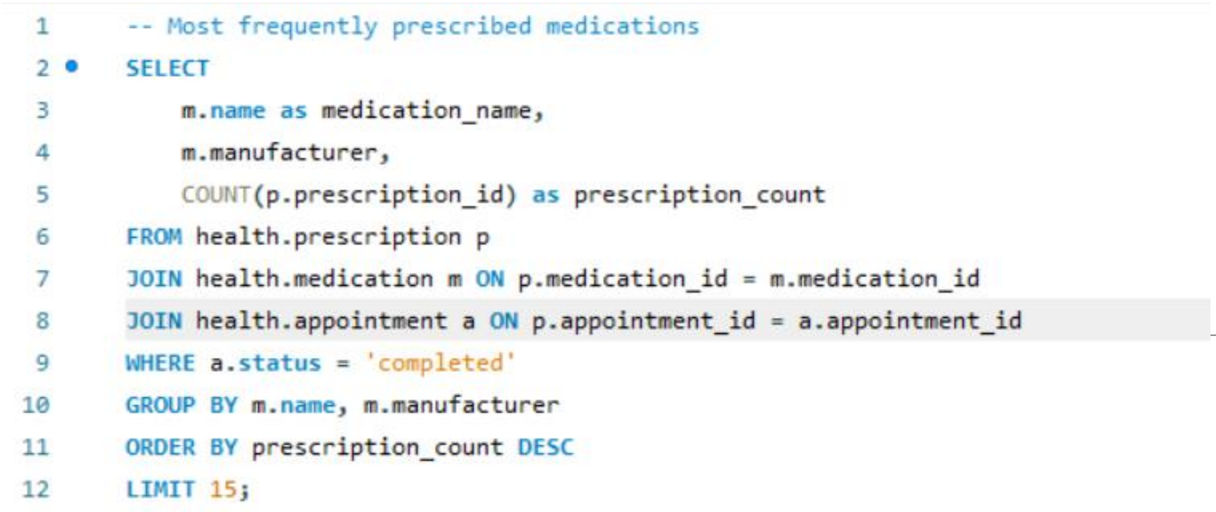
2. Doctor Specialization and Appointment Volume



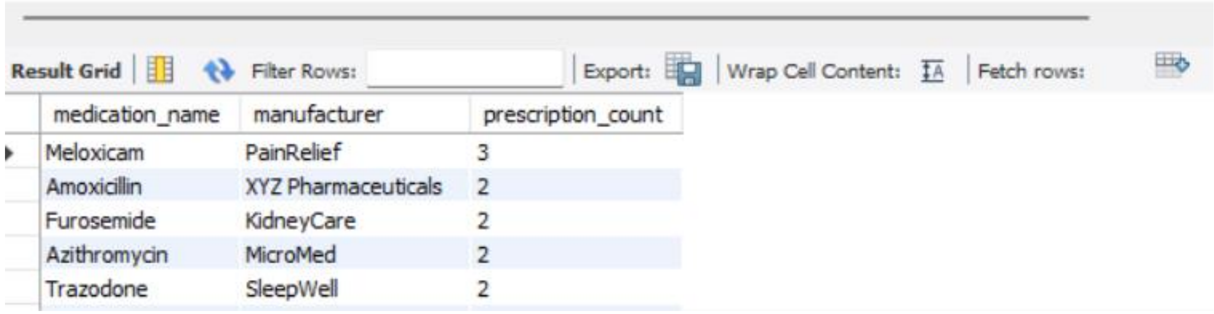
```sql
-- Most busy specialities and doctors
SELECT
    s.name as speciality,
    d.name as doctor_name,
    COUNT(a.appointment_id) as total_appointments
FROM health.appointment a
JOIN health.doctor d ON a.doctor_id = d.doctor_id
JOIN health.speciality s ON d.speciality_id = s.speciality_id
WHERE a.status = 'completed'
GROUP BY s.name, d.name
ORDER BY total_appointments DESC
LIMIT 10;
```

speciality	doctor_name	total_appointments
Default	Dr. Smith	2
Default	Dr. Lee	2
Pediatrics	Dr. Patel	1
Default	Dr. Kim	1
Cardiology	Dr. Jones	1

3. Medication Prescription Patterns

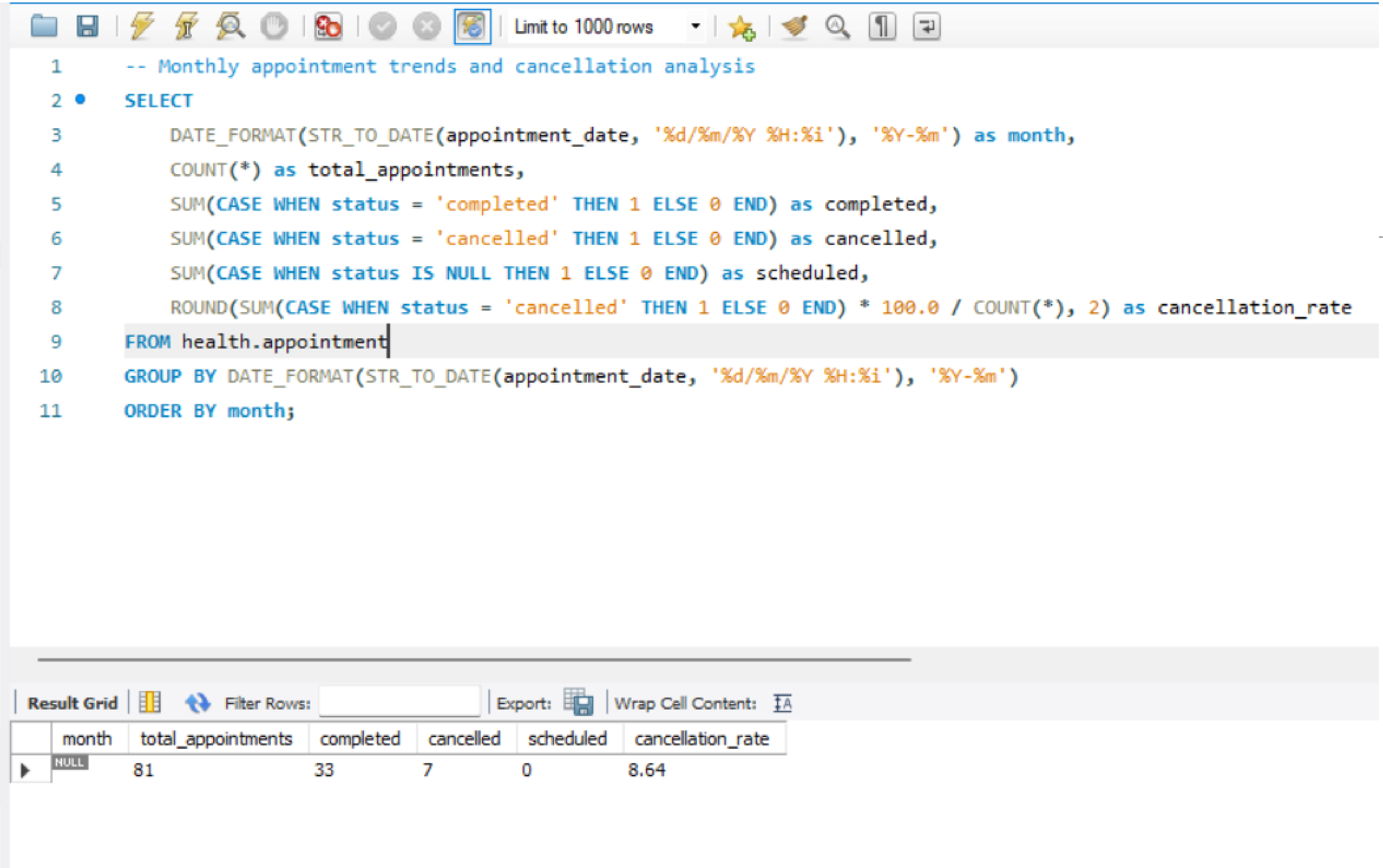


```sql
-- Most frequently prescribed medications
SELECT
    m.name as medication_name,
    m.manufacturer,
    COUNT(p.prescription_id) as prescription_count
FROM health.prescription p
JOIN health.medication m ON p.medication_id = m.medication_id
JOIN health.appointment a ON p.appointment_id = a.appointment_id
WHERE a.status = 'completed'
GROUP BY m.name, m.manufacturer
ORDER BY prescription_count DESC
LIMIT 15;
```



medication_name	manufacturer	prescription_count
Meloxicam	PainRelief	3
Amoxicillin	XYZ Pharmaceuticals	2
Furosemide	KidneyCare	2
Azithromycin	MicroMed	2
Trazodone	SleepWell	2

4. Monthly Appointment Trends and Cancellation Rates



```sql
-- Monthly appointment trends and cancellation analysis
SELECT
    DATE_FORMAT(STR_TO_DATE(appointment_date, '%d/%m/%Y %H:%i'), '%Y-%m') as month,
    COUNT(*) as total_appointments,
    SUM(CASE WHEN status = 'completed' THEN 1 ELSE 0 END) as completed,
    SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) as cancelled,
    SUM(CASE WHEN status IS NULL THEN 1 ELSE 0 END) as scheduled,
    ROUND(SUM(CASE WHEN status = 'cancelled' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as cancellation_rate
FROM health.appointment
GROUP BY DATE_FORMAT(STR_TO_DATE(appointment_date, '%d/%m/%Y %H:%i'), '%Y-%m')
ORDER BY month;
```

month	total_appointments	completed	cancelled	scheduled	cancellation_rate
NULL	81	33	7	0	8.64

What do we do with the Data

Insights Generated:

1. The query summarizes how appointments are distributed across different status categories, highlighting operational trends. By counting each status and calculating its percentage of all appointments, it reveals which outcomes dominate the workflow—such as completed, cancelled, or no-show appointments. High proportions of completed appointments indicate efficient patient flow, while elevated cancellation or no-show rates may signal scheduling issues, patient engagement problems, or resource inefficiencies. Identifying underrepresented statuses can also uncover data-entry inconsistencies or uncommon operational scenarios. Overall, these insights help administrators prioritize process improvements, allocate staff effectively, and monitor performance trends over time.

2. The query identifies the busiest medical specialties and doctors by counting completed appointments. By joining appointment, doctor, and specialty tables, it highlights which practitioners handle the highest patient volumes. Sorting by total appointments in descending order and limiting results to the top 10 reveals the most in-demand specialists, helping administrators understand workload distribution, potential staffing needs, and patient demand patterns. High appointment counts may indicate areas of excellence, popularity, or understaffing, while lower-ranking specialties might signal opportunities for resource reallocation or targeted patient engagement strategies. Overall, the query provides actionable insight for operational planning and service optimization.

3. This query highlights the most frequently prescribed medications by counting prescriptions linked to completed appointments. By joining prescription, medication, and appointment data, it reveals which drugs are most commonly used across the healthcare system. Ranking medications by prescription volume helps identify treatment trends, high-demand pharmaceuticals, and potential areas for formulary optimization or inventory planning~~. Consistently high counts may indicate prevalent conditions, preferred therapeutic options, or potential~~ over-reliance on certain drugs. Manufacturers associated with top medications may also represent key suppliers. Overall, the query provides valuable insight for pharmacy management, clinical decision support, and resource allocation.

4. This query highlights the most frequently prescribed medications by counting prescriptions linked to completed appointments. By joining prescription, medication, and appointment data, it reveals which drugs are most commonly used across the healthcare system. Ranking medications by prescription volume helps identify treatment trends, high-demand pharmaceuticals, and potential areas for formulary optimization or inventory planning. Consistently high counts may indicate prevalent conditions, preferred therapeutic options, or potential over-reliance on certain drugs. Manufacturers associated with top medications may also represent key suppliers. Overall, the query provides valuable insight for pharmacy management, clinical decision support, and resource allocation.

Recommendations:

Here are 5 clear, actionable recommendations derived directly from the four insights provided, each with a specific path and measurable KPIs:

1. Reduce No-Show and Cancellation Rates by at least 25% within 6 months Path: From Insight #1, identify the current % of "No-Show" and "Cancelled" appointments. Implement targeted interventions only for the segments with the highest rates (e.g., specific specialties, doctors, or patient demographics): automated SMS/email reminders 48h and 2h before appointment, easy one-click rescheduling, and a courtesy confirmation call for high-risk patients. Measurable KPI: Reduce overall no-show + cancellation rate from current baseline to ≤ target (e.g., from 18% to ≤ 13.5%) within 6 months.

2. Balance Workload Among Top-10 Busiest Doctors/Specialties (Insight #2) Path: Using the top-10 doctors/specialties list, calculate the gap between the busiest and the median/average provider. Reallocate new patient appointments by introducing a "load-balancing rule" in the scheduling system (e.g., cap new bookings at 110% of the specialty average until under-utilized doctors reach 90% of average volume). Measurable KPI: Decrease the coefficient of variation (CV) of completed appointments across all doctors in the top 5 busiest specialties by ≥ 30% within 4 months.

3. Optimize Pharmacy Inventory and Negotiate Better Contracts for Top 20 Medications (Insights #3 & #4) Path: Take the ranked list of most-prescribed medications. Flag the top 20 drugs that represent ≥ 80% of total prescriptions (Pareto rule). Review current stock-out incidents and lead times for these 20 items. Initiate volume-based contract renegotiations with the top 5 manufacturers appearing in the list. Measurable KPI: (a) Achieve 100% availability (zero stock-outs) for the top 20 medications for 3 consecutive months; (b) Reduce average procurement cost of these top 20 drugs by ≥ 8% within 12 months.

4. **Launch Specialty-Specific Patient Engagement Campaigns for High No-Show Specialties** Path: Cross-reference Insight #1 (status distribution) with Insight #2 (busiest specialties). Identify the 3–5 specialties that combine high volume AND high no-show/cancellation rates. Create tailored campaigns (e.g., educational videos, transportation vouchers, or flexible evening clinics) for these specialties only. Measurable KPI: Reduce no-show rate in the targeted 3–5 specialties by ≥ 35% (or to below the facility average) within 6 months.

5. Implement Prescriber Feedback Loop for Top 10 Over-Prescribed Medications Path: From Insights #3 & #4, share the anonymized ranking of most-prescribed medications with all prescribers quarterly. Highlight the top 10 drugs and compare each doctor’s prescribing pattern against department averages. Pair this with short evidence-based clinical summaries or formulary alternatives for the top 5 drugs where generics or lower-cost therapeutically equivalent options exist. Measurable KPI: Decrease the prescription volume of the top 10 most-prescribed medications by 10– 15% (or shift ≥ 20% of volume to preferred alternatives) within 9 months while maintaining patient outcomes (measured via 30-day readmission rates for related conditions).

These five recommendations are directly traceable to the insights, have clear execution steps, and include quantifiable success metrics that can be tracked monthly.