



# HEALTHCARE ANALYSIS





# WHAT WE ANALYSE...!

Patient Management: Analyzing patient demographics, their medical histories, and appointment trends.

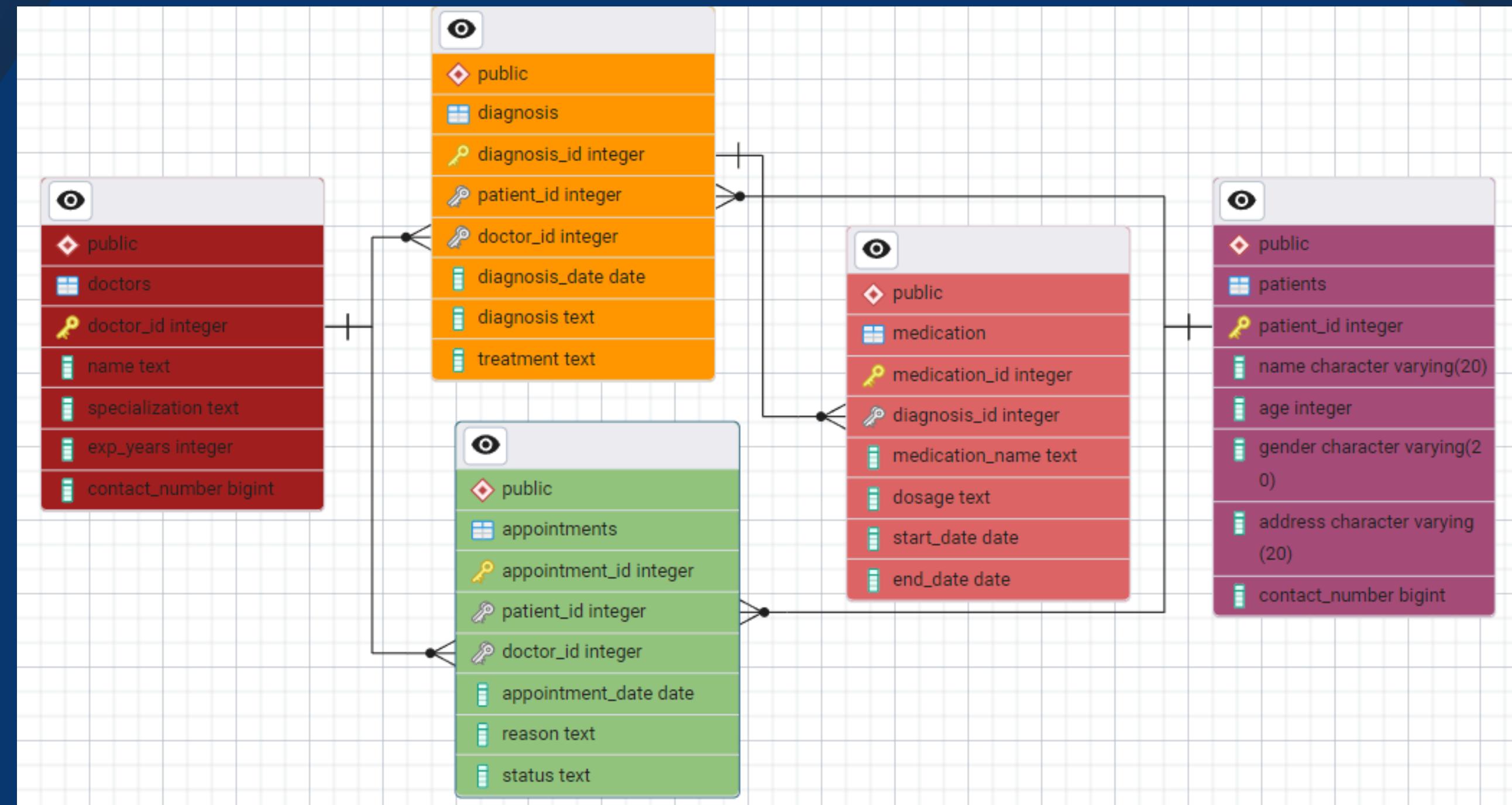
Doctor Performance Evaluation: Measuring doctor efficiency by tracking diagnoses and treatment frequency.

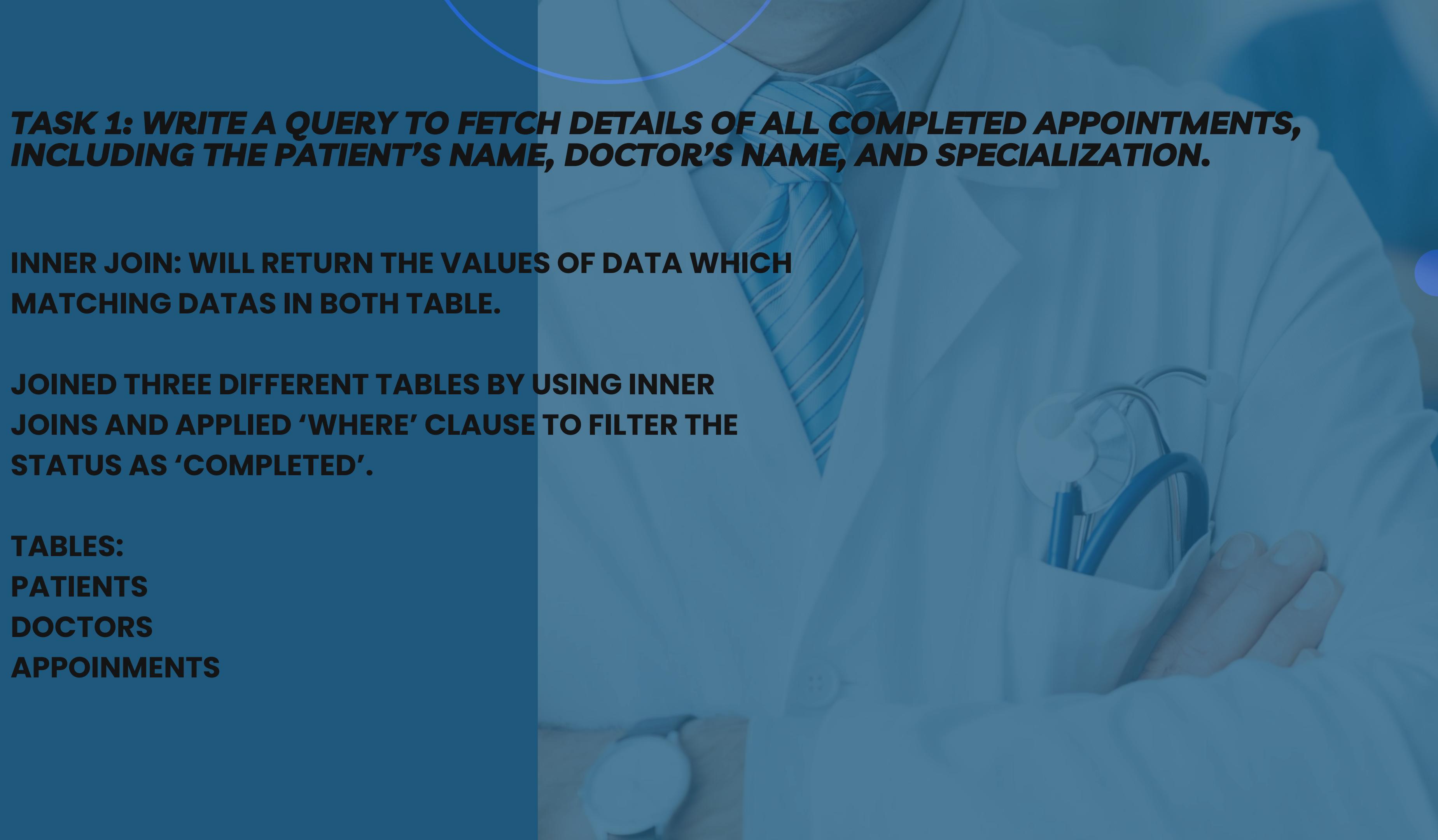
Appointment Scheduling & Completion: Understanding appointment trends, cancellations, and completions.

Medication Analysis: Identifying the most common medications prescribed to patients based on diagnoses.

Revenue and Billing Analysis: Linking appointment data with billing information to understand revenue generation.

# ER DIAGRAM





**TASK 1: WRITE A QUERY TO FETCH DETAILS OF ALL COMPLETED APPOINTMENTS, INCLUDING THE PATIENT'S NAME, DOCTOR'S NAME, AND SPECIALIZATION.**

**INNER JOIN:** WILL RETURN THE VALUES OF DATA WHICH MATCHING DATAS IN BOTH TABLE.

**JOINED THREE DIFFERENT TABLES BY USING INNER JOINS AND APPLIED 'WHERE' CLAUSE TO FILTER THE STATUS AS 'COMPLETED'.**

**TABLES:**  
**PATIENTS**  
**DOCTORS**  
**APPOINTMENTS**

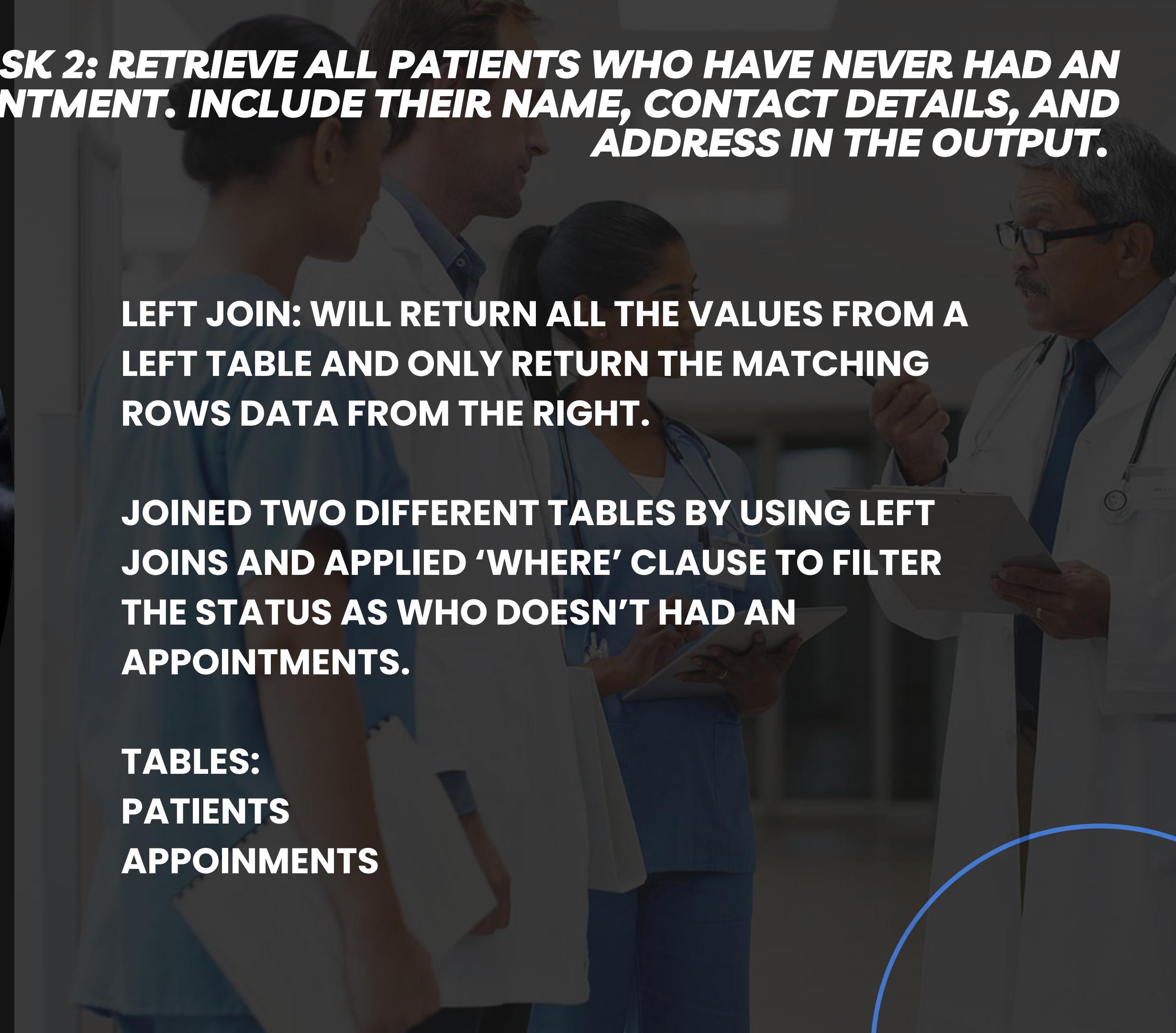


**TASK 2: RETRIEVE ALL PATIENTS WHO HAVE NEVER HAD AN APPOINTMENT. INCLUDE THEIR NAME, CONTACT DETAILS, AND ADDRESS IN THE OUTPUT.**

**LEFT JOIN:** WILL RETURN ALL THE VALUES FROM A LEFT TABLE AND ONLY RETURN THE MATCHING ROWS DATA FROM THE RIGHT.

**JOINED TWO DIFFERENT TABLES BY USING LEFT JOINS AND APPLIED 'WHERE' CLAUSE TO FILTER THE STATUS AS WHO DOESN'T HAD AN APPOINTMENTS.**

**TABLES:  
PATIENTS  
APPOINTMENTS**



**TASK 3: FIND THE TOTAL NUMBER OF DIAGNOSES FOR EACH DOCTOR, INCLUDING DOCTORS WHO HAVEN'T DIAGNOSED ANY PATIENTS. DISPLAY THE DOCTOR'S NAME, SPECIALIZATION, AND TOTAL DIAGNOSES.**

**RIGHT JOIN:** WILL RETURN ALL THE VALUES FROM A RIGHT TABLE AND ONLY RETURN THE MATCHING ROWS DATA FROM THE LEFT.

**JOINED THREE DIFFERENT TABLES BY USING RIGHT JOINS WITH COUNT() AGGREGATE FUNCTION.**

**TABLES:**  
**PATIENTS**  
**DOCTORS**  
**DIAGNOSIS**



**TASK 4: WRITE A QUERY TO IDENTIFY MISMATCHES BETWEEN THE APPOINTMENTS AND DIAGNOSES TABLES. INCLUDE ALL APPOINTMENTS AND DIAGNOSES WITH THEIR CORRESPONDING PATIENT AND DOCTOR DETAILS.**

FULL JOIN: WILL RETURN ALL THE VALUES OR ROWS DATA FROM BOTH THE TABLES.

JOINED FOUR DIFFERENT TABLES BY USING FULL JOINS AND ORDER BY ONE OF THE COLUMN.

TABLES:

PATIENTS

APPOINTMENT

DOCTORS

DIAGNOSIS



**TASK 5: FOR EACH DOCTOR, RANK THEIR PATIENTS BASED ON THE NUMBER OF APPOINTMENTS IN DESCENDING ORDER.**

APPLYING THE RANKING FUNCTIONS SUCH AS **RANK()** OR **DENSE\_RANK()**.

JOINED TWO DIFFERENT TABLES BY USING FULL JOINS. USED THE AGGREGATE FUNCTION **COUNT()**.

**TABLES:**  
**APPOINTMENT**  
**DOCTORS**

**RANK**

ASSIGNS A RANKING VALUE OF ORDERED PARTITION, IF ROWS HAVE THE SAME VALUE THEN THEY WILL ASSIGN A SAME RANK WITH THE NEXT RANKING VALUE SKIPPED

**DENSE\_RANK**

ASSIGNS A RANKING VALUE OF ORDERED PARTITION, IF ROWS HAVE THE SAME VALUE THEN THEY WILL ASSIGN A SAME RANK WITHOUT THE NEXT RANKING VALUE SKIPPED

**CONDITIONAL  
EXPRESSION:**  
ALLOWS  
YOU TO EXECUTE  
THE CODE BASED  
ON THE  
CONDITION.  
- IF CONDITION  
IS TRUE IT'LL RUN  
THE BLOCK OR  
ELSE IT'LL RUN THE  
CODE IN THE ELSE  
CONDITION.

**TASK 6: WRITE A QUERY TO CATEGORIZE PATIENTS BY AGE GROUP (E.G., 18-30, 31-50, 51+). COUNT THE NUMBER OF PATIENTS IN EACH AGE GROUP.**

WHEN AGE BETWEEN 18 AND 30 THEN  
'SENIORS'  
WHEN AGE BETWEEN 31 AND 50 THEN  
'SUPER\_SENIORS'  
else 'Senior\_Citizens'

**TABLES:  
PATIENTS**



## **TASK 7: RETRIEVE A LIST OF PATIENTS WHOSE CONTACT NUMBERS END WITH "1234" AND DISPLAY THEIR NAMES IN UPPERCASE.**

### **LIKE**

LIKE CONDITION IS USED TO GET SPECIFIED PATTERN VALUE IN THE COULMN.

### **UPPER() OR LOWER()**

CONVERTS A STRING TO UPPERCASE OR LOWERCASE.

### **NUMERIC FUNCTION**

NUMERIC FUNCTIONS IN SQL ARE USED TO PERFORM OPERATIONS ON NUMERIC DATA TYPES.

### **STRING FUNCTION**

STRING FUNCTIONS IN SQL CAN BE USED OR APPLIED TO STRING DATA TYPES.

**TASK 8: FIND PATIENTS WHO HAVE ONLY BEEN PRESCRIBED "INSULIN" IN ANY OF THEIR DIAGNOSES.**

**GET THE DATA OF 'INSULIN' MEDICATION BY EACH PATIENTS.**

**JOINED THREE DIFFERENT TABLES BY USING FULL JOINS WITH THE FILTER OF MEDICATION NAME 'INSULIN'.**

**TABLES:**  
**PATIENTS**  
**DIAGNOSIS**  
**MEDICATION**



# DATEDIFF (DATE DIFFERENCES)

CALCULATE DIFFERENCE  
BETWEEN TWO DIFFERENT DATES  
OR TIMESTAMP.

**TASK 9: CALCULATE THE AVERAGE DURATION (IN DAYS)  
FOR WHICH MEDICATIONS ARE PRESCRIBED FOR EACH  
DIAGNOSIS.**

WORKING WITH DATE FUNCTIONS LIKE  
DATEDIFF().

JOINED TWO DIFFERENT TABLES BY USING  
FULL JOINS. USED THE DATEDIFF()  
FUNCTION TO GET THE AVERAGE TIME  
PERIODS BETWEEN START AND END DATE OF  
THE MEDICATION.

TABLES:  
MEDICATION.  
DIAGNOSIS.

**TASK 10: WRITE A QUERY TO IDENTIFY THE DOCTOR WHO HAS ATTENDED THE MOST UNIQUE PATIENTS. INCLUDE THE DOCTOR'S NAME, SPECIALIZATION, AND THE COUNT OF UNIQUE PATIENTS.**

GROUP BY STATEMENT IN SQL IS USED TO ARRANGE DATA INTO GROUPS BASED ON SPECIFIED COLUMNS.

JOINED THREE DIFFERENT TABLES BY USING FULL JOINS. USED THE AGGREGATE FUNCTION COUNT().

**TABLES:**  
**DIAGNOSIS**  
**DOCTORS**  
**PATIENTS**

DISTINCT

THIS KEYWORD IS USED TO RETURN THE UNIQUE VALUES FROM THE COLUMN. IT HELPS TO ELIMINATE THE DUPLICATE RECORDS.

COUNT()

COUNT FUNCTION IS COMES UNDER THE AGGREGATE FUNCTIONS, WHICH IS USED TO RETURNS THE NO OF ROWS OF THE VALUES FROM THE COLUMN.



# END METRICS...

Accuracy of SQL Queries: Correctness in terms of logic and SQL syntax.

Data Insights: Ability to extract meaningful information from the dataset.

Efficiency: Optimization of queries to reduce execution time.

Complexity: Proper use of window functions, joins, and subqueries.

Code Documentation: Clear and concise documentation of the code.

# THANK YOU

