Project: Healthcare Management System (HMS) in SQL

- Designed a relational database with **five interconnected tables** (Appointments, Medical Records, Doctors, Patients, Billing).
- Implemented **complex SQL queries** for patient history tracking, doctor schedules, and billing reports.
- Optimized performance using **indexes and normalization techniques**.
- Ensured data security with **user roles and access control**.

1. Business Requirements

A. Patient Management

- **Registration & Demographics:**
  - **Requirements:**
    - Store patient details (Patient_ID, first name, last name, DOB, gender, contact numbers, email, address, and optional Insurance_ID).
    - Ensure essential fields (names, DOB, gender) are NOT NULL.

Analysis Queries:

- Retrieve complete patient profiles.
- List patients with missing contact information (to ensure data quality).

1. **Retrieve complete patient profiles.**

```sql
/* Patient management.§ Retrieve complete patient profiles.*/

select *
from patient_s;
```

100 %

Results | Messages

	patient_id	name	gender	contact	age	address	insurance_id
1	1	Emily Johnson	Female	9183425786	16	FL	3319
2	2	John Martinez	Female	7609275414	36	FL	4338
3	3	Michael Miller	Male	1858242624	84	IL	9554
4	4	Sophia Davis	Female	1245563124	74	LA	6942
5	5	Michael Brown	Male	1966988290	19	NY	5667
6	6	Sarah Miller	Male	9625564756	7	FL	7156
7	7	Sarah Miller	Female	2392866599	89	IL	8937
8	8	Emily Wilson	Female	1320023570	52	LA	7887
9	9	Emily Miller	Female	8759471661	47	LA	9849
10	10	Olivia Garcia	Female	9401889887	99	LA	6236
11	11	Daniel Williams	Female	3505803304	79	TX	9964

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 8,500 rows

1. List patients with missing contact information (to ensure data quality).

```sql
/* List patients with missing contact information (to ensure data quality).*/

SELECT  patient_id,name,contact,age
from patient_s
where patient_id IS  null
OR name is    null
OR contact is  null
OR age is null;
```

100 %

Results | Messages

patient_id	name	contact	age

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 0 rows

It means this is indicates that all selected columns data are not null.

1. **Identify duplicate records**

```sql
/* identify the  duplicate records*/

select insurance_id,count(*) as count
from patient_s
group by insurance_id
having count(*)>1;
```

100 %

Results | Messages

	insurance_id	count
1	9392	2
2	7185	3
3	1543	4
4	1420	2
5	4414	2
6	6306	4
7	5642	2
8	2107	3
9	2771	2
10	4437	3
11	1048	2

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 2,209 rows

Duplicates names are following:

```sql
select name ,count(*) as count
from patient_s
group by name
having count(*)>1;
```

100 %

Results | Messages

	name	count
1	Olivia Wilson	70
2	Daniel Johnson	91
3	Michael Garcia	80
4	James Garcia	79
5	Sarah Wilson	109
6	Emma Garcia	98
7	John Williams	82
8	Sophia Jones	96
9	Daniel Davis	84
10	Sophia Miller	95
11	Michael Martinez	86

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 100 rows

Contacts are unique hence they are not duplicate.

```sql
select contact,count(*) as count
from patient_s
group by contact
having count(*)>1;
```

100 %

Results | Messages

contact	count

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 0 rows

Verified Address:

```sql
select address,count(*) as count
from patient_s
group by address
having count(*)>1;
```

100 %

Results | Messages

	address	count
1	TX	1698
2	IL	1652
3	FL	1727
4	NY	1717
5	LA	1706

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 5 rows

A. Doctor Management

- **Profile Maintenance:**
  - **Requirements:**
    - Capture doctor details (Doctor_ID, name, specialization, contact info).
    - Link doctor records to appointments and medical records.
  - **Analysis:**

- Query for doctors by specialization.
- Retrieve a doctor’s appointment history.

- **Query for doctors by specialization.**

```
SELECT *
from doctors
where specialization='Cardiology';
```

100 %

Results | Messages

	doctor_id	name	specialization	experience_years	contact_info
12	66	Michael Garcia	Cardiology	6	9917177031
13	72	David Jones	Cardiology	30	4250470125
14	73	Michael Marti...	Cardiology	3	9502613951
15	74	Sarah Davis	Cardiology	19	6569821494
16	83	Sarah Smith	Cardiology	11	6192991990
17	88	David Miller	Cardiology	25	6246015306
18	91	James Jones	Cardiology	10	5072505810
19	101	Emily Martinez	Cardiology	25	8892310852
20	113	John Brown	Cardiology	32	7768911725
21	115	David Wilson	Cardiology	31	7019731730
22	118	Sophia Williams	Cardiology	18	5698633437

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 100 rows

Retrieve a doctor’s appointment history.

```
/*retrives doctors appintment history*/

SELECT d.doctor_id,d.name,d.specialization,
a.appointment_id,a.patient_id,a.appointment_date,a.status
FROM doctors AS d
INNER JOIN appointments AS a
ON d.doctor_id=a.doctor_id
order by appointment_date;
```

100 %

Results | Messages

	doctor_id	name	specialization	appointment_id	patient_id	appointment_date	status
1	393	David Smith	Dermatology	317	7718	2020-01-01	Canceled
2	359	Emily Williams	Neurology	802	7143	2020-01-01	Canceled
3	294	Sarah Johnson	Neurology	1141	8259	2020-01-01	Pending
4	44	Olivia Smith	Pediatrics	1796	1555	2020-01-01	Canceled
5	359	Emily Williams	Neurology	2855	2649	2020-01-01	Completed
6	429	Emma Wilson	Pediatrics	3836	4208	2020-01-01	Completed
7	370	Emma Jones	Neurology	7098	4505	2020-01-01	Completed
8	296	Olivia Jones	Cardiology	7269	8423	2020-01-02	Canceled
9	497	Michael Miller	Cardiology	2240	3380	2020-01-02	Completed
10	75	Michael Smith	Pediatrics	1676	1024	2020-01-02	Pending
11	415	Emily Johnson	Neurology	1679	4862	2020-01-02	Pending

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 8,500 rows

Appointment Scheduling & Management

- **Booking & Tracking:**
  - **Requirements:**
    - Record appointment details (Appointment_ID, Patient_ID, Doctor_ID, Appointment_Date, Status, optional notes).
    - Ensure proper status tracking (Scheduled, Completed, Cancelled).
  - **Analysis:**
    - List upcoming appointments with patient and doctor details.
    - Count appointments by status to identify cancellation or no-show trends.

- **List upcoming appointments with patient and doctor details.**

```sql
/*LIST UPCOMING  APPOINTMENTS WITH patient and DOCTOR DETAILS:*/

SELECT  a.appointment_id,a.appointment_date,a.status,
p.patient_id,p.name,p.age,p.insurance_id,p.contact,
d.doctor_id,d.name,d.specialization,d.contact_info
from appointments as a
inner join patient_s as  p
on a.patient_id=p.patient_id
inner join doctors as d
on a.doctor_id=d.doctor_id
where status='pending'
order by a.appointment_date asc;
```

100 %

Results | Messages

	appointment_id	appointment_date	status	patient_id	name	age	insurance_id	contact	doctor_id	name	specialization	contact_info
1	1141	2020-01-01	Pending	8259	Daniel Williams	71	5802	1402161481	294	Sarah Johnson	Neurology	8390738830
2	1676	2020-01-02	Pending	1024	Michael Garcia	32	5988	7728117305	75	Michael Smith	Pediatrics	8828393087
3	1679	2020-01-02	Pending	4862	James Martinez	9	1417	4443070785	415	Emily Johnson	Neurology	1988256958
4	981	2020-01-03	Pending	1328	Sophia Wilson	31	2770	3889551613	309	Sophia Johnson	Neurology	8640438182
5	1109	2020-01-03	Pending	258	Sophia Miller	12	8330	1196730958	12	Emma Wilson	Orthopedics	3940661712
6	3225	2020-01-03	Pending	1856	John Smith	80	7398	6849117435	350	John Wilson	Orthopedics	5136119564
7	4783	2020-01-04	Pending	8492	John Johnson	42	1348	9061021032	175	Sarah Wilson	Neurology	9786937955
8	4212	2020-01-05	Pending	3485	Olivia Brown	2	3151	3926868412	368	Emily Smith	Pediatrics	5821734590
9	3292	2020-01-06	Pending	17	David Garcia	87	4997	5643942608	400	Emily Johnson	Pediatrics	5050470274
10	1991	2020-01-07	Pending	9	Emily Miller	47	9849	8759471661	455	Sophia Martinez	Cardiology	4217579081
11	6714	2020-01-07	Pending	3564	James Brown	41	6900	9287923139	148	Emma Martinez	Dermatology	9554444486

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 2,875 rows

If you want to track future appointments also then you can use following

where status='pending' AND a.appointment_date >= CAST(GETDATE() AS DATE)

```sql
where status='pending' AND a.appointment_date >= CAST(GETDATE() AS DATE)
order by a.appointment_date asc;
```

If we want to fetch the data in between year, we can use following:

```sql
SELECT  a.appointment_id,a.appointment_date,a.status,
p.patient_id,p.name,p.age,p.insurance_id,p.contact,
d.doctor_id,d.name,d.specialization,d.contact_info
from appointments as a
inner join patient_s as  p
on a.patient_id=p.patient_id
inner join doctors as d
on a.doctor_id=d.doctor_id
where a.status='pending' and appointment_date between '1/7/2024' and '7/1/2024'

order by a.appointment_date asc;
```

100 %

Results | Messages

	appointment_id	appointment_date	status	patient_id	name	age	insurance_id	contact	doctor_id	name	specialization	contact_info
78	2216	2024-02-14	Pending	1651	Olivia Wilson	60	4721	1897194987	13	Emily Garcia	Cardiology	4940235793
79	7290	2024-02-15	Pending	6700	Olivia Williams	72	9231	3474435801	234	David Johnson	Dermatology	1736847649
80	4049	2024-02-15	Pending	2459	James Johnson	19	1987	9790362784	151	Sophia Miller	Pediatrics	1557029359
81	8347	2024-02-15	Pending	7332	Olivia Jones	67	7601	1787476641	378	Michael Brown	Pediatrics	8048314095
82	1312	2024-02-15	Pending	620	Olivia Martinez	80	4185	5326120650	247	James Davis	Cardiology	4076814378
83	6106	2024-02-15	Pending	6738	Emma Martinez	2	7928	6293036894	493	James Davis	Pediatrics	8586077135
84	482	2024-02-16	Pending	3768	James Miller	51	7054	9210464777	123	David Smith	Orthopedics	8544568677
85	6973	2024-02-17	Pending	7140	Olivia Brown	70	9749	2895363629	235	Sarah Wilson	Cardiology	4664827130
86	4756	2024-02-17	Pending	7547	David Williams	80	3066	6496191184	28	Sophia Johnson	Neurology	8220893204
87	7180	2024-02-17	Pending	5512	Daniel Williams	68	2390	4671452430	392	Daniel Garcia	Orthopedics	6490630327
88	385	2024-02-19	Pending	5976	Sophia Jones	82	3885	8461349363	279	James Wilson	Dermatology	3040770192

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 125 rows

- Count appointments by status to identify cancellation or no-show trends.

```sql
/*Count appointments by status to identify cancellation or no-show trends.*/

SELECT appointment_date,status,count(*) as total_canceled
from appointments
where status='canceled'
group by appointment_date,status
order by appointment_date
```

100 %

Results | Messages

	appointment_date	status	total_canceled
1	2020-01-01	Canceled	3
2	2020-01-02	Canceled	1
3	2020-01-03	Canceled	4
4	2020-01-04	Canceled	3
5	2020-01-05	Canceled	3
6	2020-01-06	Canceled	2
7	2020-01-07	Canceled	5
8	2020-01-08	Canceled	3

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 1,272 rows

Medical Records Management

- **Record Keeping:**
  - **Requirements:**
    - Store medical records (Record_ID, Patient_ID, Doctor_ID, Diagnosis, Treatment, Record_Date).
    - Maintain a history of treatments and diagnoses per patient.

Analysis Queries:

- Retrieve complete medical histories for a given patient.
- Analyze diagnosis frequency or treatment types across the patient base.

- **Retrieve complete medical histories for a given patient.**

```sql
/*Retrieve complete medical histories for a given patient*/

select m.record_id,m.patient_id,m.diagnosis,m.prescription,m.record_date,
p.name,p.age,p.contact,p.insurance_id
from medical_records as m
inner join patient_s as p
on m.patient_id=p.patient_id
order by m.record_date ASC
```

100 %

Results | Messages

	record_id	patient_id	diagnosis	prescription	record_date	name	age	contact	insurance_id
1	5373	4909	Fracture	Med C	2020-01-01	James Miller	18	6837477995	7391
2	4665	8110	Flu	Med E	2020-01-01	Sarah Jones	37	7550790535	7763
3	327	4629	Heart Disease	Med E	2020-01-02	David Jones	26	3010227131	8806
4	5617	3881	Heart Disease	Med C	2020-01-02	Sophia Martinez	78	1604764594	9197
5	964	4056	Heart Disease	Med B	2020-01-03	David Garcia	17	2767273804	9154
6	938	1493	Flu	Med D	2020-01-03	Michael Jones	43	5626613018	5083
7	3131	3345	Fracture	Med A	2020-01-03	John Jones	66	2939151042	3637
8	102	7938	Migraine	Med B	2020-01-03	Olivia Miller	68	1905901159	1332

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 2,786 rows

And retrieve for specific name, id of patient (we can use here where clause)

```sql
/*Retrieve complete medical histories for a given patient*/

select m.record_id,m.patient_id,m.diagnosis,m.prescription,m.record_date,
p.name,p.age,p.contact,p.insurance_id
from medical_records as m
inner join patient_s as p
on m.patient_id=p.patient_id
where m.patient_id=7537
order by m.record_date ASC;
```

100 %

Results | Messages

	record_id	patient_id	diagnosis	prescription	record_date	name	age	contact	insurance_id
1	52	7537	Heart Disease	Med A	2021-10-18	John Williams	84	8095992471	2865

- **Analyze diagnosis frequency or treatment types across the patient base.**

```sql
/* analyse diagnosis frequency*/

select diagnosis,count(*) as count_frequ
from medical_records
group by diagnosis
order by count_frequ DESC
```

100 %

Results | Messages

	diagnosis	count_frequ
1	Fracture	572
2	Heart Disease	563
3	Migraine	559
4	Flu	559
5	Allergy	533

Treatment type across the patient:

```sql
SELECT m.diagnosis, d.specialization, COUNT(*) AS treatment_count,
       AVG(m.treatment_cost) AS avg_treatment_cost
FROM medical_records AS m
INNER JOIN doctors AS d ON m.doctor_id = d.doctor_id
GROUP BY m.diagnosis, d.specialization
ORDER BY avg_treatment_cost DESC;
```

100 %

Results | Messages

	diagnosis	specialization	treatment_count	avg_treatment_cost
1	Fracture	Dermatology	94	5636
2	Allergy	Orthopedics	97	5504
3	Migraine	Orthopedics	110	5460
4	Migraine	Pediatrics	112	5378
5	Flu	Cardiology	110	5364
6	Fracture	Neurology	129	5308
7	Migraine	Cardiology	125	5285
8	Allergy	Cardiology	106	5259
9	Migraine	Dermatology	105	5215
10	Flu	Orthopedics	112	5123
11	Flu	Pediatrics	126	5080
12	Heart D...	Cardiology	114	5053
13	Fracture	Pediatrics	110	5013
14	Allergy	Dermatology	92	4981
15	Flu	Neurology	121	4958
16	Heart D...	Neurology	114	4954
17	Allergy	Pediatrics	133	4872
18	Heart D...	Dermatology	101	4850
19	Fracture	Orthopedics	125	4779
20	Fracture	Cardiology	114	4772
21	Allergy	Neurology	105	4758
22	Heart D...	Pediatrics	115	4750
23	Heart D...	Orthopedics	119	4731
24	Flu	Dermatology	90	4703
25	Migraine	Neurology	107	4653

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:0

E. Billing & Payment Management

- **Invoice & Payment Processing:**
  - **Requirements:**
    - Record billing details (Billing_ID, Patient_ID, Appointment_ID, Amount, Payment_Status, Payment_Date).
    - Track payments to flag outstanding balances.

- **Analysis:**
  - Calculate total revenue (sum of paid bills).
  - List billing records with pending payments.

- **Calculate total revenue (sum of paid bills).**

```
select  sum(total_amount) as sum_paidbills
from billing
where payment_status='Paid'
```

100 %

Results | Messages

	sum_paidbills
1	74030708

-

- List billing records with pending payments.

```
select *
from billing
where payment_status='pending'
```

100 %

Results | Messages

	bill_id	patient_id	total_amount	payment_status	payment_date
1	4	5753	12732	Pending	NULL
2	5	5656	80389	Pending	NULL
3	10	2495	3349	Pending	NULL
4	11	1189	1844	Pending	NULL
5	12	7468	4731	Pending	NULL
6	15	2383	1850	Pending	NULL
7	16	6832	2562	Pending	NULL
8	18	7139	2111	Pending	NULL
9	19	6054	1183	Pending	NULL
10	20	4324	17100	Pending	NULL
11	21	5085	3427	Pending	NULL
12	22	5085	82879	Pending	NULL
13	23	5085	4351	Pending	NULL
14	24	6359	3864	Pending	NULL

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 5,626 rows

F. Data Integrity & Security

- **Integrity & Access:**
  - **Requirements:**
    - Enforce referential integrity through primary/foreign key constraints.
    - Validate data types and use check constraints (e.g., valid phone numbers).

Analysis:

- Identify records with missing or anomalous data.
- Audit changes in data (if triggers or logs are implemented).

added foreign keys ensures **data integrity** and prevents invalid or orphaned records.

HealthcareQueries....NP7JSN\Priya (77))*

```sql
/*1.linking appointments to the patients  and doctors */

ALTER TABLE appointments
add constraint fk_appointpatients
foreign key (patient_id) references patient_s(patient_id);

ALTER TABLE appointments
add  constraint fk_appointdoctors
foreign key(doctor_id) references doctors(doctor_id);

/*2linking medical records with patint and doctors*/

ALTER TABLE medical_records
add constraint fk_medpatient
foreign key (patient_id) references patient_s(patient_id);

ALTER TABLE medical_records
add constraint fk_meddoctors
foreign key (doctor_id) references doctors(doctor_id);

/*3linking billing with patient and appointment  */

ALTER TABLE billing
add constraint fk_billpatient
foreign key (patient_id) references patient_s(patient_id);
```

100 %

Messages

```
Commands completed successfully.
```

100 %

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 0 rows

- **Identify records with missing or anomalous data.**

Billing table has null values for payment_date is due to payment_status is pending

```sql
select *
from  billing
where patient_id is null or total_amount is null  or payment_status is null
or payment_date is null;
```

100 %

Results | Messages

	bill_id	patient_id	total_amount	payment_status	payment_date
1	4	5753	12732	Pending	NULL
2	5	5656	80389	Pending	NULL
3	10	2495	3349	Pending	NULL
4	11	1189	1844	Pending	NULL
5	12	7468	4731	Pending	NULL
6	15	2383	1850	Pending	NULL
7	16	6832	2562	Pending	NULL
8	18	7139	2111	Pending	NULL
9	19	6054	1183	Pending	NULL
10	20	4324	17100	Pending	NULL

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 5,626 rows

Otherwise, another tables have not null values:

```sql
/* I dentify missing data values*/

select *
from appointments
WHERE patient_id IS NULL OR doctor_id IS NULL OR appointment_date IS NULL;

select *
from patient_s
where name is null or contact is null or age is null;

/*select *
from  billing
where patient_id is null or total_amount is null  or payment_status is null
or payment_date is null;*/

select *
from doctors
where name is null or contact_info is null;

select *
from medical_records
where patient_id is null or doctor_id is null or treatment_cost is null
or record_date is null;
```

100 %

Results | Messages

appointment_id	patient_id	doctor_id	appointment_date	status

patient_id	name	gender	contact	age	address	insurance_id

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 0 rows

- **Check anomalous data**

- HealthcareQueries....NP7JSN\Priya (68))*

```
/* identify anolumes data inavlid values */

/*check total amount which is less than or equal to zero*/
select *
from billing
where total_amount<= 0;

/*check for invalid appointment dates* (future or old dates)*/

select *
from appointments
where appointment_date < = '1/1/2000';

/*find invalid patient ages */

select *
from patient_s
where age<=0 or age>=120;
```

- **For payementstatus= no incorrect status**

```
/* check incorrect payemnt status */
select distinct payment_status
from billing;
```

100 %

Results | Messages

	payment_status
1	Paid
2	Pending

Query executed successfully.

Analytical Reporting & Decision Support

- **Reporting & Dashboards:**
  - **Requirements:**

        - Create views, stored procedures, or reports that consolidate key information from multiple tables.
        - Enable real-time dashboards for KPIs and operational metrics.
    - **Analysis:**
        - Generate trend reports (e.g., monthly appointments, revenue trends).
        - Aggregate metrics for performance reviews.

- **Generate trend reports (e.g., monthly appointments, revenue trends).**

The views calculated total appointments per month

```sql
create view monthly_appointment
as
select format(appointment_date, 'YYY-MM') as month,
count (appointment_id) as toatal_appointments
from appointments
group by format(appointment_date, 'YYY-MM')

select *
from monthly_appointment;
```

Results | Messages

	month	toatal_appointments
1	YYY-01	849
2	YYY-02	759
3	YYY-03	752
4	YYY-04	664
5	YYY-05	691
6	YYY-06	633
7	YYY-07	711
8	YYY-08	739
9	YYY-09	594
10	YYY-10	704
11	YYY-11	686
12	YYY-12	718

The views calculated total revenue collected per month:

```sql
CREATE VIEW Monthly_Revenue AS
SELECT
    FORMAT(payment_date, 'yyyy-MM') AS month,
    SUM(total_amount) AS total_revenue
FROM Billing
WHERE payment_date IS NOT NULL
GROUP BY FORMAT(payment_date, 'yyyy-MM');

select *
from Monthly_Revenue;

select format(total_revenue/1000000,'n2') + 'M' as revenue_m
from Monthly_Revenue
```

.00 %

Results | Messages

	month	total_revenue
1	2024-12	24713279
2	2025-01	37634320
3	2025-02	11683109

With converstion by using format()

```sql
select format(total_revenue/1000000,'n2') + 'M' as revenue_m
from Monthly_Revenue
```

Results | Messages

	revenue_m
1	24.00M
2	37.00M
3	11.00M

view for yearly revenue trends:

```sql
create view yearly_revenue
as
select format( payment_date,'YYY') as year,
       sum(total_amount) as total_year

    from billing
    where payment_date is not null
    group by format( payment_date,'YYY');

    select *
    from yearly_revenue;

    select  round(total_year/1000000,2) as total,year
    from yearly_revenue
```

Results | Messages

	year	total_year
1	YYY	74030708

Key Performance Indicators (KPIs)

A. Appointment Efficiency

- **Average Wait Time:**
    - *Definition:* Average days between record date and the actual appointment.

- *Sample Query:*

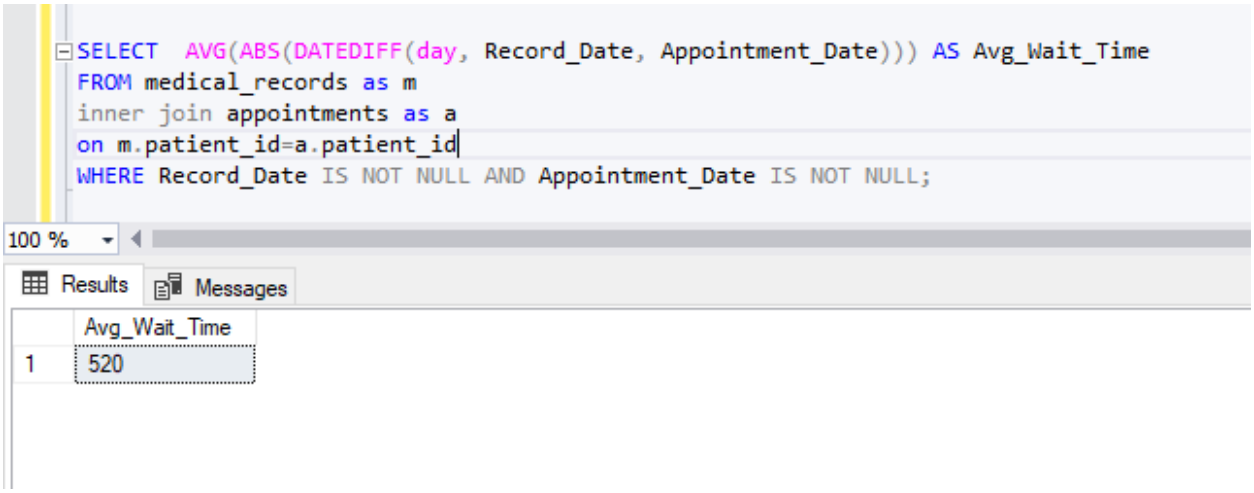
  sql
  CopyEdit
  SELECT AVG(DATEDIFF(day, Booking_Date, Appointment_Date)) AS Avg_Wait_Time
  FROM Appointments
  WHERE Booking_Date IS NOT NULL AND Appointment_Date IS NOT NULL;

- **No-Show Rate:**
  - *Definition:* Percentage of appointments where the patient did not attend.
- **Cancellation Rate:**
  - *Definition:* Percentage of appointments cancelled by patients or doctors.

Analysis:

Wrong data entry observed: In some cases, record date is after appointment date this is because appointment is scheduled before record date.

So used ABS () to avoid negative values. It will drop incorrect records.



/* fetch the incorrect records */

```sql
/* to findincorrecy records*/
SELECT a.Appointment_ID, m.Record_Date, a.Appointment_Date,
DATEDIFF(day, m.Record_Date, a.Appointment_Date) AS Wait_Days
FROM Appointments as a
inner join medical_records as m
on a.patient_id=m.patient_id
WHERE m.Record_Date > a.Appointment_Date OR DATEDIFF(day, m.Record_Date, a.Appointment_Date) > 365;
```

Results | Messages

	Appointment_ID	Record_Date	Appointment_Date	Wait_Days
677	2480	2023-05-28	2022-04-14	-409
678	2488	2023-04-12	2020-01-20	-1178
679	2495	2022-10-03	2022-09-13	-20
680	2498	2020-12-25	2023-07-21	938
681	2500	2022-01-04	2021-08-21	-136
682	2502	2023-02-04	2021-04-28	-647
683	2505	2023-05-09	2023-04-07	-32
684	2506	2022-10-12	2024-01-31	476
685	2509	2021-03-16	2020-11-25	-111
686	2510	2023-01-06	2020-06-17	-933
687	2512	2020-03-13	2024-02-05	1424
688	2514	2021-08-07	2022-09-25	414
689	2530	2020-01-22	2022-08-21	942
690	2532	2022-05-31	2022-03-24	-68

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 2,215 rows

Only fetch correct records with appointment and record date which is not greater than 365 and appointment is scheduled after the record date.

```sql
/*      Only fetch correct records with appointmnet and record date which is not graeter than 365 and appointment is schduled after the record date.
   */

   SELECT a.Appointment_ID, m.Record_Date, a.Appointment_Date,
      DATEDIFF(day, m.Record_Date, a.Appointment_Date) AS Wait_Days
FROM Appointments as a
inner join medical_records as m
on a.patient_id=m.patient_id
WHERE m.Record_Date < a.Appointment_Date and DATEDIFF(day, m.Record_Date, a.Appointment_Date) < 365;
```

100 %

Results | Messages

	Appointment_ID	Record_Date	Appointment_Date	Wait_Days
199	2996	2020-01-23	2021-01-02	345
200	2996	2020-02-28	2021-01-02	309
201	3022	2021-04-06	2021-08-03	119
202	3031	2023-03-24	2023-10-10	200
203	3047	2022-08-21	2023-01-22	154
204	3051	2021-02-28	2021-05-11	72
205	3057	2020-09-16	2020-10-07	21
206	3059	2020-05-24	2021-01-03	224
207	3065	2021-09-02	2022-05-16	256
208	3067	2022-02-22	2022-06-24	122
209	3074	2022-03-22	2022-10-01	193
210	3077	2021-09-27	2022-02-27	153

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 600 rows

For fixing incorrect dates we can use following query:

```sql
update m
set m.record_date = DATEADD(day,-30,a.appointment_date)
from medical_records as m
join appointments as a
on m.patient_id=a.patient_id
where m.record_date > a.appointment_date or datediff(day,m.record_date,a.appointment_date)>365;
```

100 %

Messages

```
(587 rows affected)

Completion time: 2025-03-25T13:03:57.2122010+05:30
```

Appointment Cancellation Rate

The percentage of appointments that were **cancelled** by either the patient or doctor.

High cancellation rates may show **scheduling issues**.

```sql
/* use check constraint for future */

select
count(case when status ='canceled' then 1 end) * 100.0 /count(*) as canellation_rate
from appointments;
```

100 %

Results | Messages

	canellation_rate
1	33.400000000000

Here we used percentage formula for calculation;

Appoint cancellation rate = (total count of cancel appointments *100)/total appointments

Financial Performance

- **Total Revenue:**

Total amount collected from paid billing records.

```
select round(sum(total_amount),2) as toatlbilling_amount
 from billing
 where payment_status ='paid';
```

100 %

Results | Messages

	toatlbilling_amount
1	74030708

```
select sum(total_amount) as totalbilling_amount
from billing
where payment_status='paid';
```

100 %

Results | Messages

	toatlbilling_amount
1	74030708

-
- **Payment Collection Efficiency:**
    - *Definition:* Ratio of payments received versus total billed.
- **Average Billing Turnaround Time:**
    - *Definition:* Time between appointment completion and payment receipt.

- **Payment Collection Efficiency:**
- *Definition:* Ratio of payments received versus total billed.

```sql
SELECT
    FORMAT((SUM(b.total_amount) * 100.0 / SUM(m.treatment_cost)), 'N2') AS Payment_Collection_
FROM billing AS b
JOIN medical_records AS m
ON b.patient_id = m.patient_id;
```

Results | Messages

Payment_Collection_Efficiency
331.04

-

Formula we have used here is:

Formula:

$$\left(\frac{\text{Total Payments Received}}{\text{Total Billed Amount}}\right) \times 100$$

- **Average Billing Turnaround Time:**

Time between appointment completion and payment receipt.

```sql
select AVG(datediff(day,a.appointment_date,b.payment_date) )as average_billing
from appointments as a
join billing as b
on a.patient_id=b.patient_id
where a.appointment_date is not  null and
b.payment_date is not null
and b.payment_status='paid'
and b.payment_date >a.appointment_date
and datediff(day,a.appointment_date,b.payment_date)<365;
```

Results | Messages

	average_billing
1	331

For the wrong data entry, I have checked column payment date and appointment date of two tables;

```sql
select b. payment_date, a.appointment_date
from billing as b
join appointments as a
on b.patient_id=a.patient_id
where b.payment_status='paid'
order by b.payment_date,a.appointment_date asc;
```

	payment_date	appointment_date
53	2024-12-12	2022-04-18
54	2024-12-12	2022-05-06
55	2024-12-12	2022-06-10
56	2024-12-12	2022-06-21
57	2024-12-12	2022-07-05
58	2024-12-12	2022-07-11
59	2024-12-12	2022-08-21
60	2024-12-12	2022-09-13
61	2024-12-12	2022-10-14

Query executed successfully. DESKTOP-HNP7JSN\SQLEXPRESS ... DESKTOP-HNP7JSN\Priya ... HEALTHCARE 00:00:00 5,854 rows

C. Doctor Performance

- **Appointments per Doctor:**

Average number of appointments handled per doctor.

```sql
select count(a.appointment_id) as toatalapooi,d.doctor_id,d.name,d.specialization,(COUNT(a.appointment_id) * 1.0 / (SELECT COUNT(DISTINCT doctor_id) FRO
    AS avg_appointments_per_doctor
from appointments as a
inner join doctors as d
on a.doctor_id=d.doctor_id
where a.status='completed'
group by d.name,d.specialization,d.doctor_id
order by d.doctor_id asc
```

	toatalapooi	doctor_id	name	specialization	avg_appointments_per_doctor
3	5	3	David Jones	Dermatology	0.010000000000
4	2	4	James Garcia	Neurology	0.004000000000
5	6	5	Sarah Brown	Pediatrics	0.012000000000
6	6	6	John Williams	Cardiology	0.012000000000
7	8	7	Sophia Willi...	Pediatrics	0.016000000000
8	2	8	David John...	Cardiology	0.004000000000
9	4	9	Michael Ma...	Dermatology	0.008000000000
10	9	10	John Wilson	Cardiology	0.018000000000

Query executed successfully. DESKTOP-HNP7JSN\SQLEXPRESS ... DESKTOP-HNP7JSN\Priya ... HEALTHCARE 00:00:00 500 rows

D. Patient Engagement

- **Patient Retention Rate:**

Percentage of patients returning for multiple appointments.

- **New Patient Growth:**

Number of new patient registrations over a period.

- **Patient Retention Rate:**

Percentage of patients returning for multiple appointments.

```sql
SELECT
    FORMAT((COUNT(DISTINCT CASE WHEN appointment_count > 1 THEN patient_id END) * 100.0
            / COUNT(DISTINCT patient_id)), 'N2') AS Returning_Patient_Percentage
FROM (
    SELECT patient_id, COUNT(appointment_id) AS appointment_count
    FROM appointments
    GROUP BY patient_id
) AS subquery
```

00 %

Results | Messages

	Returning_Patient_Percentage
1	41.85

-

- **New Patient Growth:**

Number of new patient registrations over a period.

```sql
SELECT COUNT(DISTINCT patient_id) AS new_registrations, record_date
FROM medical_records a
WHERE record_date BETWEEN '2022-04-08' AND '2022-09-10'
GROUP BY record_date
ORDER BY record_date;
```

100 %

Results | Messages

	new_registrations	record_date
1	3	2022-04-08
2	4	2022-04-09
3	1	2022-04-11
4	4	2022-04-12
5	2	2022-04-13
6	1	2022-04-14
7	7	2022-04-16
8	1	2022-04-17
9	1	2022-04-18
10	2	2022-04-19
11	1	2022-04-20
12	3	2022-04-21
13	1	2022-04-22
14	1	2022-04-23
15	1	2022-04-24
16	1	2022-04-25

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 129 rows

E. Operational Efficiency

- **Data Completeness:**
  - *Definition:* Percentage of records that have all required fields completed.
- **System Performance Metrics:**
  - *Definition:* Query response times, system uptime, etc.

- **Data Completeness:**

Percentage of records that have all required fields completed.

I got result 100 %, it means all records completely present in the table.

```sql
select
(count (case when patient_id is not null and
doctor_id is not null and
diagnosis is not null and
treatment_cost is not null and
record_date is not null then
 1 end)* 100 /count(*))  as percentage_completerecords
from medical_records ;
```

100 %

Results | Messages

	percentage_completerecords
1	100

- **System Performance Metrics:**

Query response times, system uptime, etc.

3. Mapping Technical Analysis Query Questions to Business Requirements & KPIs

Below are sample technical questions you might ask—and corresponding SQL query examples—that align with your business requirements and KPIs:

Patient Management

How many patients have incomplete contact information?

```sql
select
count(case when  patient_id is  null and
name is  null and gender is  null and contact is  null and age is  null
and address is  null and insurance_id is  null then 1 end) as incomplete_info
from patient_s;
```

100 %

Results | Messages

	incomplete_info
1	0

Retrieve full profile and appointment details for a specific patient.

```sql
/* Retrieve full profile and appointment details for a specific patient.
Query Example*/


select  p.name,p.gender,p.contact,p.age,p.address, p.insurance_id,a.appointment_date,a.doctor_id,a.appointment_date,
a.status
from patient_s as p
join appointments  as a
on p.patient_id=a.patient_id
where p.patient_id=8;
```

100 %

Results | Messages

	name	gender	contact	age	address	insurance_id	appointment_date	doctor_id	appointment_date	status
1	Emily Wilson	Female	1320023570	52	LA	7887	2022-04-02	392	2022-04-02	Canceled
2	Emily Wilson	Female	1320023570	52	LA	7887	2023-04-17	463	2023-04-17	Completed

Doctor Management

Which doctors specialize in a given field?

```sql
/*  Doctor Management
Question: Which doctors specialize in a given field?
Query Example:*/

select doctor_id,name,experience_years,contact_info
from doctors
where specialization='Neurology';
```

100 %

Results | Messages

	doctor_id	name	experience_years	contact_info
1	2	Emily Williams	7	1842218175
2	4	James Garcia	2	6220112610
3	17	Emma Martinez	22	7763127090
4	20	Daniel Wilson	6	6970706596
5	23	Olivia Johnson	39	3771556826
6	28	Sophia Johnson	24	8220893204
7	29	James Wilson	21	6561524819
8	30	Sarah Williams	33	7442712371
9	32	Michael Martinez	12	1207411740
10	40	Michael Smith	21	4779757864
11	41	Sarah Jones	8	2841691554

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 111 rows

```sql
/* What is the appointment history for a particular doctor?*/

select d.name,d.doctor_id,d.contact_info, a.appointment_id,a.patient_id,a.appointment_date,a.status
from doctors as d
inner join appointments as a
on d.doctor_id=a.doctor_id
where d.name='David Davis';
```

100 %

Results | Messages

	name	doctor_id	contact_info	appointment_id	patient_id	appointment_date	status
1	David Davis	217	5961022755	161	2647	2022-04-12	Canceled
2	David Davis	303	5781930106	408	7240	2021-02-26	Completed
3	David Davis	122	4376776843	462	897	2024-03-07	Completed
4	David Davis	463	7955508355	485	7605	2022-06-23	Pending
5	David Davis	303	5781930106	531	3582	2021-04-12	Canceled
6	David Davis	120	7621026923	583	7050	2023-01-21	Canceled
7	David Davis	293	5388864272	724	2891	2021-02-20	Canceled
8	David Davis	303	5781930106	777	7964	2022-09-03	Completed
9	David Davis	1	9017508553	881	4202	2021-06-17	Pending
10	David Davis	1	9017508553	929	6461	2023-02-15	Canceled
11	David Davis	1	9017508553	1095	7982	2021-07-24	Pending

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 119 rows

Appointment Scheduling & Management

What are the upcoming appointments?

```sql
SELECT
    (CAST(SUM(CASE WHEN Status = 'Cancelled' THEN 1 ELSE 0 END) AS FLOAT) / COUNT(*)) * 100 AS Cancellation_Rate
FROM Appointments;
```

100 %

Results | Messages

	Cancellation_Rate
1	0

Medical Records Management

- **Question:** Retrieve a full medical history for a patient.

```sql
select *
from medical_records as m
join doctors as d
on m.doctor_id=d.doctor_id
join patient_s as p
on m.patient_id=p.patient_id
where p.name='Sarah Miller';
```

100 %

Results | Messages

	record_id	patient_id	doctor_id	diagnosis	prescription	treatment_cost	record_date	doctor_id	name	specialization	experience_years	contact_info	patient_id	name	gender	contact	age	addr
2	577	4080	37	Fracture	Med C	7367	2022-11-21	37	Olivia Brown	Orthopedics	13	6253983040	4080	Sarah Miller	Female	5730326299	25	NY
3	746	922	290	Flu	Med A	2924	2022-09-12	290	Emily Garcia	Cardiology	23	8170947880	922	Sarah Miller	Female	2485004151	93	NY
4	1166	4187	44	Fracture	Med C	6728	2023-02-04	44	Olivia Smith	Pediatrics	14	2861628580	4187	Sarah Miller	Female	3017038050	65	FL
5	1877	6108	410	Flu	Med D	6012	2023-07-29	410	Sophia Jones	Orthopedics	2	6834927399	6108	Sarah Miller	Male	5863577629	20	IL
6	2311	5970	212	Allergy	Med A	5641	2020-01-28	212	Daniel Davis	Neurology	9	5823400168	5970	Sarah Miller	Male	1121925907	25	LA
7	2367	3660	11	Migraine	Med E	5789	2021-09-20	11	James Garcia	Pediatrics	34	7066347716	3660	Sarah Miller	Male	7933760613	17	IL
8	3272	3309	349	Fracture	Med C	3830	2022-04-02	349	Emily Jones	Pediatrics	26	6991137512	3309	Sarah Miller	Female	7405375826	56	TX
9	3480	4745	381	Migraine	Med E	6529	2020-01-06	381	Emma Smith	Orthopedics	29	9897590238	4745	Sarah Miller	Male	7739682656	93	LA

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 22 rows

Billing & Payment Management

What is the total revenue collected?

```sql
/*  Billing & Payment Management
Question: What is the total revenue collected? */
select *
from billing;

select sum(total_amount) as total_revenue
from billing
where  payment_status='paid';
```

00 %

Results | Messages

	total_revenue
1	74030708

What are the outstanding (pending) payments?

```sql
/*What are the outstanding (pending) payments?*/

select bill_id,patient_id,total_amount,payment_date
from billing
where payment_status='pending';
```

100 %

Results | Messages

	bill_id	patient_id	total_amount	payment_date
1	4	5753	12732	NULL
2	5	5656	80389	NULL
3	10	2495	3349	NULL
4	11	1189	1844	NULL
5	12	7468	4731	NULL
6	15	2383	1850	NULL
7	16	6832	2562	NULL
8	18	7139	2111	NULL
9	19	6054	1183	NULL
10	20	4324	17100	NULL
11	21	5085	3427	NULL

Query executed successfully. | DESKTOP-HNP7JSN\SQLEXPRESS ... | DESKTOP-HNP7JSN\Priya ... | HEALTHCARE | 00:00:00 | 5,626 rows

Operational & Data Integrity Checks

How many records in a table have null values in critical fields?

```
select count(*)
from patient_s
where name is null or contact is null or age is null or
insurance_id  is null
```

100 %

Results | Messages

	(No column name)
1	0

Audit query to check the integrity of foreign key relationships.

```
/*Audit query to check the integrity of foreign key relationships. */

SELECT a.Appointment_ID
FROM Appointments a
LEFT JOIN patient_s p ON a.Patient_ID = p.Patient_ID
WHERE p.Patient_ID IS NULL;
```

100 %

Results | Messages

Appointment_ID