CREATE DATABASE HEALTHCARE\_ANALYTICS;

USE HEALTHCARE\_ANALYTICS;

CREATE TABLE PATIENTS(

PATIENT\_ID INT AUTO\_INCREMENT PRIMARY KEY,

NAME VARCHAR(20) NOT NULL,

AGE INT NOT NULL,

GENDER VARCHAR(20) NOT NULL,

ADDRESS VARCHAR(50) NOT NULL,

CONTACT\_NUMBER VARCHAR(15) NOT NULL

);

DESC PATIENTS;

CREATE TABLE DOCTORS(

DOCTOR\_ID INT AUTO\_INCREMENT PRIMARY KEY,

NAME VARCHAR(50) NOT NULL,

SPECIALIZATION VARCHAR(50) NOT NULL,

EXPERIENCE\_YEAR INT NOT NULL,

CONTACT\_NUMBER VARCHAR(15) NOT NULL

);

CREATE TABLE APPOINMENTS(

APPOINTMENT\_ID INT AUTO\_INCREMENT,

PATIENT\_ID INT NOT NULL,

DOCTOR\_ID INT NOT NULL,

APPOINTMENT\_DATE DATE NOT NULL,

REASON VARCHAR(50) NOT NULL,

STATUS VARCHAR(50) NOT NULL,

PRIMARY KEY (APPOINTMENT\_ID, PATIENT\_ID),

FOREIGN KEY (PATIENT\_ID) REFERENCES PATIENTS(PATIENT\_ID) ON DELETE CASCADE,

FOREIGN KEY (DOCTOR\_ID) REFERENCES DOCTORS(DOCTOR\_ID)

);

CREATE TABLE DIAGNOSIS (

DIAGNOSIS\_ID INT AUTO\_INCREMENT,

PATIENT\_ID INT NOT NULL,

DOCTOR\_ID INT NOT NULL,

APPOINTMENT\_ID INT NOT NULL,

DIAGNOSIS\_DATE DATE NOT NULL,

DIAGNOSIS VARCHAR(50) NOT NULL,

TREATMENT VARCHAR(50) NOT NULL,

PRIMARY KEY (DIAGNOSIS\_ID, PATIENT\_ID, APPOINTMENT\_ID),

FOREIGN KEY (PATIENT\_ID) REFERENCES PATIENTS(PATIENT\_ID) ON DELETE CASCADE,

FOREIGN KEY (DOCTOR\_ID) REFERENCES DOCTORS(DOCTOR\_ID) ON DELETE CASCADE,

FOREIGN KEY (APPOINTMENT\_ID) REFERENCES APPOINMENTS(APPOINTMENT\_ID) ON DELETE CASCADE

);

CREATE TABLE MEDICATIONS(

MEDICATION\_ID INT AUTO\_INCREMENT,

DIAGNOSIS\_ID INT NOT NULL,

MEDICATION VARCHAR(50) NOT NULL,

DOSAGE VARCHAR(50) NOT NULL,

START\_DATE DATE NOT NULL,

END\_DATE DATE NOT NULL,

PRIMARY KEY ( MEDICATION\_ID ,DIAGNOSIS\_ID),

FOREIGN KEY (DIAGNOSIS\_ID) REFERENCES DIAGNOSIS(DIAGNOSIS\_ID) ON DELETE CASCADE

);

DESC PATIENTS;

DESC DOCTORS;

DESC APPOINMENTS;

DESC DIAGNOSIS;

DESC MEDICATIONS;

DROP TABLE MEDICATIONS;

SELECT \* FROM PATIENTS;

SELECT \* FROM DOCTORS;

SELECT \* FROM APPOINMENTS;

SELECT \* FROM DIAGNOSIS;

SELECT \* FROM MEDICATIONS;

CREATE TABLE MEDICATIONS (

MEDICATION\_ID INT AUTO\_INCREMENT PRIMARY KEY,

DIAGNOSIS\_ID INT NOT NULL,

MEDICATION VARCHAR(50) NOT NULL,

DOSAGE VARCHAR(50) NOT NULL,

START\_DATE DATE NOT NULL,

END\_DATE DATE NOT NULL,

FOREIGN KEY (DIAGNOSIS\_ID) REFERENCES DIAGNOSIS(DIAGNOSIS\_ID) ON DELETE CASCADE,

CHECK (END\_DATE > START\_DATE)

);

CREATE TABLE MEDICATIONS (

MEDICATION\_ID INT AUTO\_INCREMENT PRIMARY KEY,

DIAGNOSIS\_ID INT NOT NULL,

MEDICATION VARCHAR(50) NOT NULL,

DOSAGE VARCHAR(50) NOT NULL,

START\_DATE DATE NOT NULL,

END\_DATE DATE NOT NULL,

FOREIGN KEY (DIAGNOSIS\_ID) REFERENCES DIAGNOSIS(DIAGNOSIS\_ID) ON DELETE CASCADE

);

ALTER TABLE DIAGNOSIS

DROP FOREIGN KEY diagnosis\_ibfk\_3,

DROP COLUMN APPOINTMENT\_ID;

ALTER TABLE DIAGNOSIS

DROP PRIMARY KEY,

ADD PRIMARY KEY (DIAGNOSIS\_ID, PATIENT\_ID);

TRUNCATE TABLE MEDICATIONS;

SELECT d.DIAGNOSIS\_ID, d.PATIENT\_ID, d.DOCTOR\_ID, a.APPOINTMENT\_ID, d.DIAGNOSIS\_DATE, d.DIAGNOSIS, d.TREATMENT

FROM DIAGNOSIS d

JOIN APPOINMENTS a

ON d.PATIENT\_ID = a.PATIENT\_ID AND d.DOCTOR\_ID = a.DOCTOR\_ID

AND d.DIAGNOSIS\_DATE = a.APPOINTMENT\_DATE;

ALTER TABLE DIAGNOSIS

ADD COLUMN APPOINTMENT\_ID INT NULL;

SET SQL\_SAFE\_UPDATES = 0;

SET SQL\_SAFE\_UPDATES = 1;

UPDATE DIAGNOSIS d

LEFT JOIN APPOINMENTS a

ON d.PATIENT\_ID = a.PATIENT\_ID

AND d.DOCTOR\_ID = a.DOCTOR\_ID

AND d.DIAGNOSIS\_DATE = a.APPOINTMENT\_DATE

SET d.APPOINTMENT\_ID = a.APPOINTMENT\_ID;

ALTER TABLE DIAGNOSIS

ADD CONSTRAINT fk\_appointment

FOREIGN KEY (APPOINTMENT\_ID) REFERENCES APPOINMENTS(APPOINTMENT\_ID)

ON DELETE SET NULL;

-- Inner and Equi Joins--

SELECT

PATIENTS.NAME AS PATIENT\_NAME,

DOCTORS.NAME AS DOCTOR\_NAME,

DOCTORS.SPECIALIZATION

FROM

APPOINMENTS

INNER JOIN PATIENTS ON APPOINMENTS.PATIENT\_ID=PATIENTS.PATIENT\_ID

INNER JOIN DOCTORS ON APPOINMENTS.DOCTOR\_ID=DOCTORS.DOCTOR\_ID

WHERE

APPOINMENTS.STATUS='COMPLETED';

-- Left Join with Null Handling --

SELECT

PATIENTS.NAME AS PATIENT\_NAME,

PATIENTS.CONTACT\_NUMBER,

PATIENTS.ADDRESS

FROM

PATIENTS

LEFT JOIN APPOINMENTS ON PATIENTS.PATIENT\_ID=APPOINMENTS.PATIENT\_ID

WHERE

APPOINMENTS.APPOINTMENT\_ID IS NULL;

-- Right Join and Aggregate Functions--

SELECT

d.NAME AS DOCTOR\_NAME,

d.SPECIALIZATION,

COALESCE(COUNT(di.DIAGNOSIS\_ID), 0) AS TOTAL\_DIAGNOSIS

FROM

DIAGNOSIS di

RIGHT JOIN DOCTORS d ON di.DOCTOR\_ID = d.DOCTOR\_ID

GROUP BY d.DOCTOR\_ID, d.NAME, d.SPECIALIZATION

ORDER BY TOTAL\_DIAGNOSIS DESC;

-- Full Join for Overlapping Data--

SELECT

A.APPOINTMENT\_ID,

A.PATIENT\_ID,

P.NAME AS PATIENT\_NAME,

A.DOCTOR\_ID,

D.NAME AS DOCTOR\_NAME,

A.APPOINTMENT\_DATE,

A.REASON,

DIAG.DIAGNOSIS\_ID,

DIAG.DIAGNOSIS\_DATE,

DIAG.DIAGNOSIS,

DIAG.TREATMENT

FROM APPOINMENTS A

LEFT JOIN DIAGNOSIS DIAG

ON A.PATIENT\_ID = DIAG.PATIENT\_ID

AND A.DOCTOR\_ID = DIAG.DOCTOR\_ID

AND A.APPOINTMENT\_DATE = DIAG.DIAGNOSIS\_DATE

LEFT JOIN PATIENTS P ON A.PATIENT\_ID = P.PATIENT\_ID

LEFT JOIN DOCTORS D ON A.DOCTOR\_ID = D.DOCTOR\_ID

UNION

SELECT

NULL AS APPOINTMENT\_ID,

DIAG.PATIENT\_ID,

P.NAME AS PATIENT\_NAME,

DIAG.DOCTOR\_ID,

D.NAME AS DOCTOR\_NAME,

NULL AS APPOINTMENT\_DATE,

NULL AS REASON,

DIAG.DIAGNOSIS\_ID,

DIAG.DIAGNOSIS\_DATE,

DIAG.DIAGNOSIS,

DIAG.TREATMENT

FROM DIAGNOSIS DIAG

LEFT JOIN PATIENTS P ON DIAG.PATIENT\_ID = P.PATIENT\_ID

LEFT JOIN DOCTORS D ON DIAG.DOCTOR\_ID = D.DOCTOR\_ID

WHERE NOT EXISTS (

SELECT 1 FROM APPOINMENTS A

WHERE A.PATIENT\_ID = DIAG.PATIENT\_ID

AND A.DOCTOR\_ID = DIAG.DOCTOR\_ID

AND A.APPOINTMENT\_DATE = DIAG.DIAGNOSIS\_DATE

);

SELECT CONSTRAINT\_NAME

FROM information\_schema.KEY\_COLUMN\_USAGE

WHERE TABLE\_NAME = 'DIAGNOSIS'

AND COLUMN\_NAME = 'APPOINTMENT\_ID';

ALTER TABLE DIAGNOSIS DROP FOREIGN KEY fk\_appointment;

ALTER TABLE DIAGNOSIS DROP COLUMN APPOINTMENT\_ID;

ALTER TABLE DIAGNOSIS DROP PRIMARY KEY, ADD PRIMARY KEY (DIAGNOSIS\_ID);

ALTER TABLE DIAGNOSIS

ADD CONSTRAINT fk\_patient FOREIGN KEY (PATIENT\_ID) REFERENCES PATIENTS(PATIENT\_ID) ON DELETE CASCADE,

ADD CONSTRAINT fk\_doctor FOREIGN KEY (DOCTOR\_ID) REFERENCES DOCTORS(DOCTOR\_ID) ON DELETE CASCADE;

DESC diagnosis;

-- Window Functions (Ranking and Aggregation) --

SELECT

doctor\_id,

COUNT(appointment\_id) AS total\_appointments

FROM Appoinments

GROUP BY doctor\_id

ORDER BY total\_appointments DESC;

SELECT

doctor\_id,

total\_appointments,

DENSE\_RANK() OVER (

ORDER BY total\_appointments DESC

) AS rank\_position

FROM (

SELECT

doctor\_id,

COUNT(appointment\_id) AS total\_appointments

FROM Appoinments

GROUP BY doctor\_id

) AS doctor\_counts

ORDER BY rank\_position;

-- Conditional Expressions --

SELECT

CASE

WHEN age BETWEEN 18 AND 30 THEN '18-30'

WHEN age BETWEEN 31 AND 50 THEN '31-50'

WHEN age >= 51 THEN '51+'

ELSE 'Unknown' -- Handles NULL or invalid ages

END AS age\_group,

COUNT(\*) AS patient\_count

FROM Patients

GROUP BY age\_group

ORDER BY age\_group;

-- Numeric and String Functions --

SELECT

UPPER(NAME) AS UPPERCASE\_NAME,

CONTACT\_NUMBER

FROM PATIENTS

WHERE CONTACT\_NUMBER LIKE '%1234';

-- Subqueries for Filtering --

SELECT p.PATIENT\_ID, p.NAME

FROM PATIENTS p

JOIN DIAGNOSIS d ON p.PATIENT\_ID = d.PATIENT\_ID

JOIN MEDICATIONS m ON d.DIAGNOSIS\_ID = m.DIAGNOSIS\_ID

GROUP BY p.PATIENT\_ID, p.NAME

HAVING COUNT(DISTINCT m.MEDICATION) = 1 AND MAX(m.MEDICATION) = 'Insulin';

-- Date and Time Functions --

SELECT

DIAGNOSIS\_ID,

AVG(DATEDIFF(END\_DATE,START\_DATE)) AS AVG\_DURATION\_DAYS

FROM MEDICATIONS

WHERE START\_DATE IS NOT NULL

GROUP BY DIAGNOSIS\_ID

ORDER BY AVG\_DURATION\_DAYS DESC;

-- Complex Joins and Aggregation --

SELECT

D.DOCTOR\_ID,

D.NAME,

D.SPECIALIZATION,

COUNT(DISTINCT A.PATIENT\_ID) AS UNIQUE\_PATIETNTS\_COUNT

FROM DOCTORS D

JOIN APPOINMENTS A ON D.DOCTOR\_ID = A.DOCTOR\_ID

GROUP BY D.DOCTOR\_ID , D.NAME, D.SPECIALIZATION

ORDER BY COUNT(DISTINCT A.PATIENT\_ID) DESC;