Patient Table

Column Name	Data Type	Constraints
patient_id	INT	PRIMARY KEY
first_name	VARCHAR(50)	NOT NULL
last_name	VARCHAR(50)	NOT NULL
dob	DATE	NOT NULL
gender	CHAR(1)	CHECK (gender IN ('M','F'))
contact_no	Number	UNIQUE
email	VARCHAR(100)	UNIQUE
blood_group	VARCHAR(3)	CHECK (blood_group IN ('A+','A-','B+','B-','AB+','AB-','O+','O-'))

Doctor Table

Column Name	Data Type	Constraints
doctor_id	INT	PRIMARY KEY
doctor_name	VARCHAR(100)	NOT NULL
specialization	VARCHAR(50)	NOT NULL
contact_no	Number	UNIQUE
salary	DECIMAL(10,2)	DEFAULT 50000 CHECK (salary >= 30000)

Appointment Table

Column Name	**Data Type**	**Constraints**
appointment_id	INT	PRIMARY KEY
patient_id	INT	NOT NULL, FOREIGN KEY REFERENCES PATIENT(patient_id)
doctor_id	INT	NOT NULL, FOREIGN KEY REFERENCES DOCTOR(doctor_id)
appointment_date	DATE	NOT NULL
status	VARCHAR(20)	DEFAULT 'Scheduled' CHECK (status IN ('Scheduled','Completed','Cancelled'))

BILL Table

Column Name	**Data Type**	**Constraints**
bill_id	INT	PRIMARY KEY
appointment_id	INT	NOT NULL, FOREIGN KEY REFERENCES APPOINTMENT(appointment_id)
total_amount	DECIMAL(10,2)	CHECK (total_amount >= 0)
paid_amount	DECIMAL(10,2)	CHECK (paid_amount >= 0 AND paid_amount <= total_amount)
payment_mode	VARCHAR(20)	DEFAULT 'Cash' CHECK (payment_mode IN ('Cash','Card','UPI'))

1. Create the above tables (Patient, Doctor, Appointment, Bill) by using the above information.

2. Add a column address (VARCHAR(200), NOT NULL) to the PATIENT table.

3. Change doctor_name in the DOCTOR table to VARCHAR(120) and ensure it remains NOT NULL.

4. Remove the email column from the PATIENT table.

5. Add a department_id column in DOCTOR and make it a foreign key referencing a new table DEPARTMENT(department_id)

6. Rename the BILL table to BILLING.

7. Rename the status column in APPOINTMENT to appointment_status.

8. Remove the foreign key from doctor_id in APPOINTMENT table.

9. Apply a constraint in case Doctor left then appointment automatically cancelled.