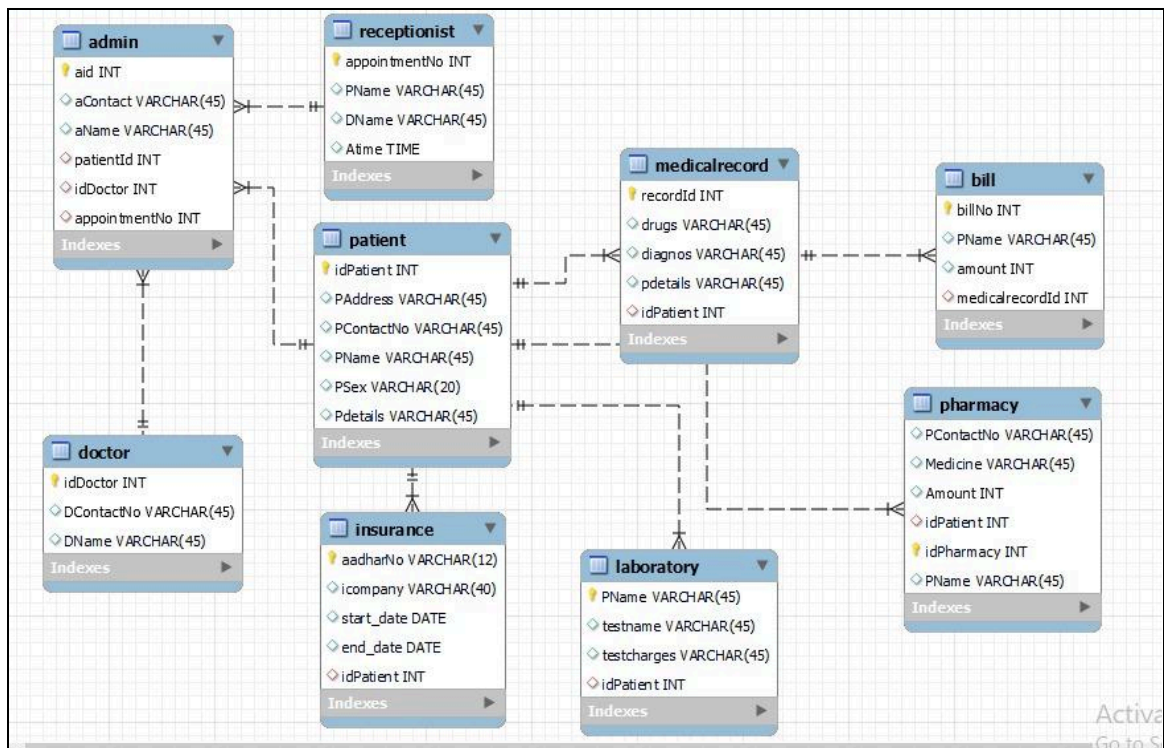


- Defining the schema diagram for the Hospital management system.

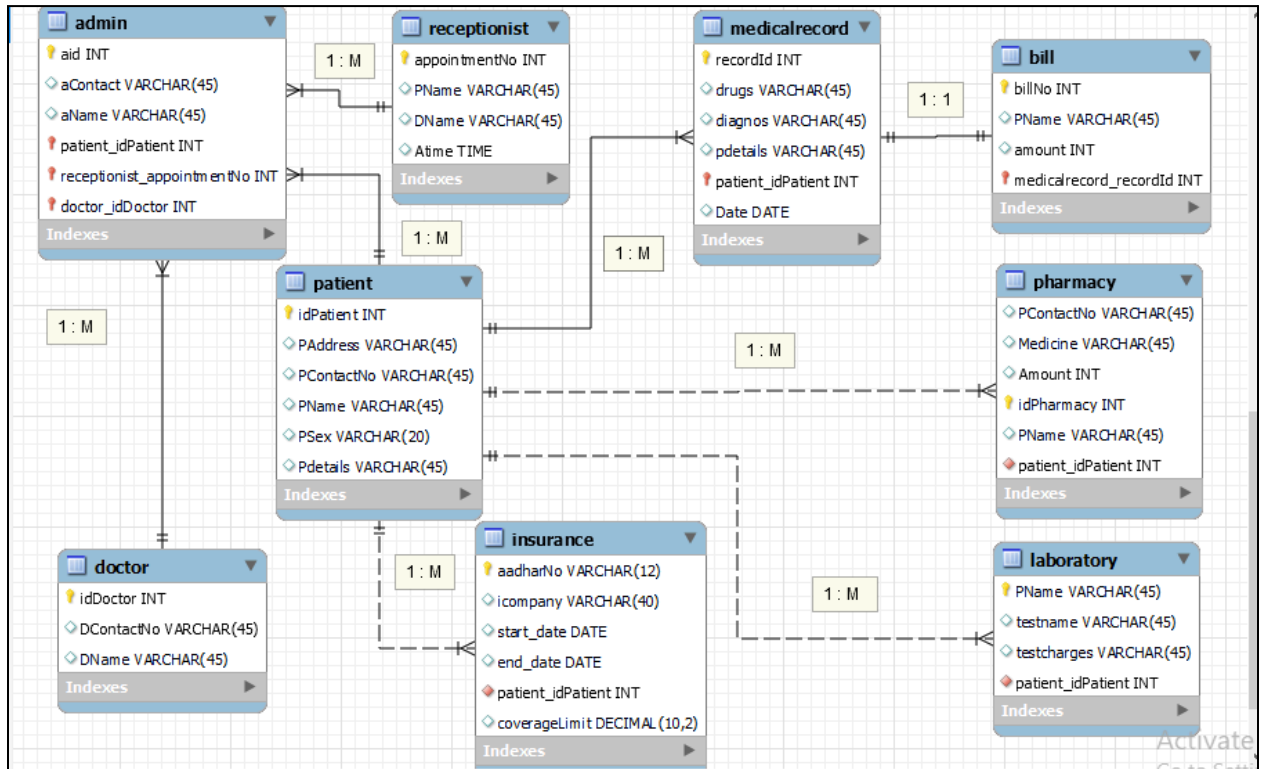
For this i referred various available schema diagram for hospital management system. I designed two versions of the schema diagram. Version 1 of the hospital system keeps things simple but repeats information. Version 2 is organized more efficiently by separating data into different parts, making it neater and preventing mistakes. The second version is normalized form of HMS.

- Step 1:** I decided total 9 tables for the HMS. i.e. Admin, Patient, Doctor, receptionist, medicalrecord, bill, pharmacy, insurance, laboratory.



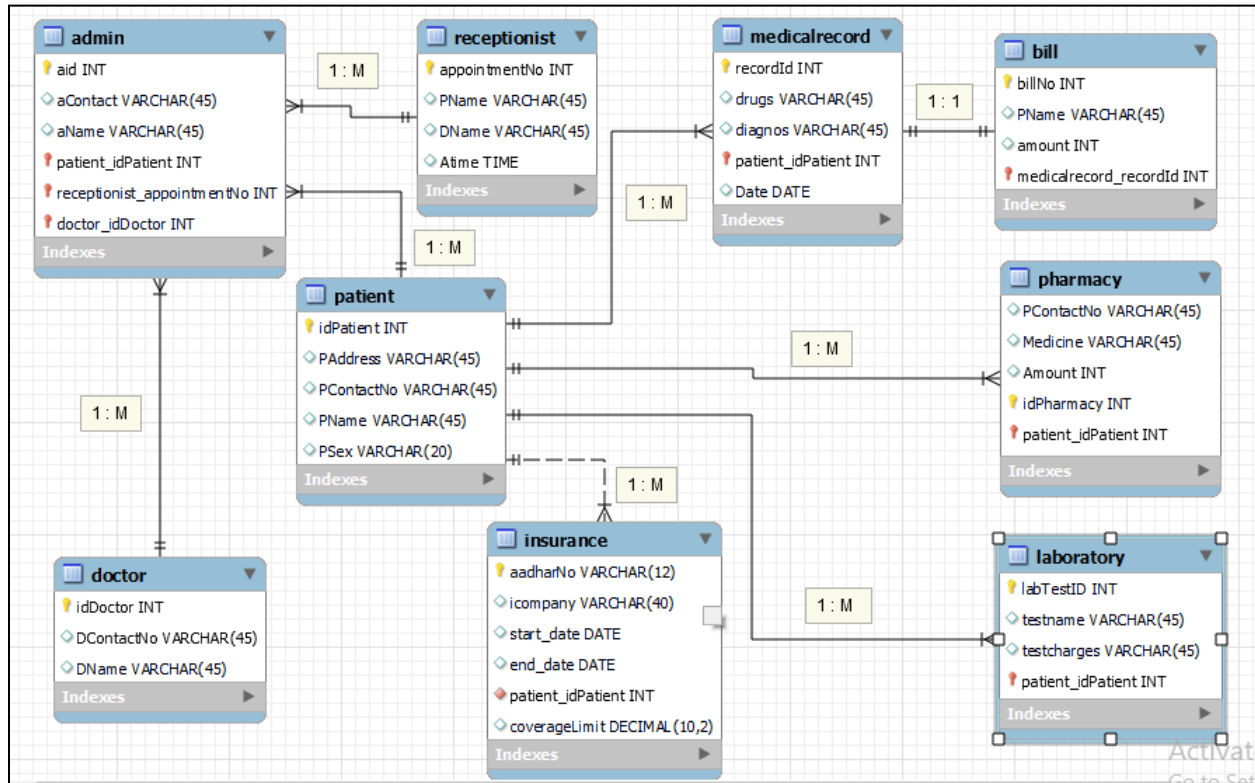
Deciding how many tables should be there in the DB

- **Step 2 :** Defining the relationships between these 9 tables. i.e.(1:1, 1:M, M:1, M:M)



Schema_Version 1

- **Step 3** : Now we can see that there is redundancy of some attributes like **"PatientName"** from Laboratory and Pharmacy tables and **"patient details"** from MedicalRecord and Patient table. Hence we need to normalize the DB.



Schema_Version 2_Normalized

Changes applied to Schema_Version 1 to get the normalized form -

- 1) Removed "patient_details" from MedicalRecord and Patient table
- 2) Removed "PatientName" from Laboratory and Pharmacy tables
- 3) Introduced "labTestID" as primary key of the laboratory table

- Queires

1) Write necessary queries to register new user roles and personas

```
277 • INSERT INTO patient (idPatient, PAddress, PContactNo, PName, PSex)
278 VALUES
279 (11, '123 Main Street, Mumbai, India', '+91 9876543210', 'Aarav Patel', 'Male'),
280 (12, '456 Street Avenue, Delhi, India', '+91 8765432109', 'Zara Singh', 'Female'),
281 (13, '789 Garden Road, Bangalore, India', '+91 7654321098', 'Ishaan Sharma', 'Male'),
282 (14, '101 River View, Kolkata, India', '+91 6543210987', 'Anaya Gupta', 'Female'),
283 (15, '234 Hillside Lane, Chennai, India', '+91 5432109876', 'Vihaan Kapoor', 'Male'),
284 (16, '567 Skyline Plaza, Hyderabad, India', '+91 4321098765', 'Avani Mehra', 'Female'),
285 (17, '890 Lakeside Drive, Pune, India', '+91 3210987654', 'Arjun Reddy', 'Male'),
286 (18, '1122 Valley Street, Ahmedabad, India', '+91 2109876543', 'Kavya Patel', 'Female'),
287 (19, '334 Park Avenue, Jaipur, India', '+91 1098765432', 'Advait Kumar', 'Male'),
288 (20, '556 Mountain View, Chandigarh, India', '+91 9876543210', 'Myra Choudhary', 'Female');
289
```

Output

#	Time	Action	Message	Duration / Fetch
34	04:39:41	UPDATE medicalrecord SET date = '2024-02-14' WHERE idPatient = 10	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
35	04:42:26	INSERT INTO patient (idPatient, PAddress, PContactNo, PName, PSex) VALUES (11, '123 ...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec

```
290 • select * from patient
291
```

	idPatient	PAAddress	PContactNo	PName	PSex
	11	123 Main Street, Mu...	+91 98765...	Aarav Patel	Male
	12	456 Street Avenue, ...	+91 87654...	Zara Singh	Female
	13	789 Garden Road, Ba...	+91 76543...	Ishaan Sharma	Male
	14	101 River View, Kolka...	+91 65432...	Anaya Gupta	Female
	15	234 Hillside Lane, Ch...	+91 54321...	Vihaan Kapoor	Male
	16	567 Skyline Plaza, Hy...	+91 43210...	Avani Mehra	Female
	17	890 Lakeside Drive, P...	+91 32109...	Arjun Reddy	Male
	18	1122 Valley Street, A...	+91 21098...	Kavya Patel	Female

2) Write necessary queries to add to the list of diagnosis of the patient tagged by date.

```

292 • select * from medicalrecord;
293
294 • INSERT INTO medicalrecord (drugs, diagnos, pdetails, idPatient, date)
295   VALUES ('Paracetamol, Ibuprofen', 'Fever', 'Patient complaints of high temperature', 11, '2024-02-01');
296
297
298

```

Result Grid

recordId	drugs	diagnos	pdetails	idPatient	date
1	Paracetamol	Common Cold	Prescription for flu symptoms	1	2024-02-05
2	Aspirin	Headache	Prescription for headache relief	2	2024-02-06
3	Amoxicillin	Sinus Infection	Prescription for antibiotic	3	2024-02-07
4	Ibuprofen	Back Pain	Prescription for pain management	4	2024-02-08
5	Omeprazole	Acid Reflux	Prescription for digestive issues	5	2024-02-09
6	Metformin	Diabetes	Prescription for diabetes control	6	2024-02-10
7	Atorvastatin	High Cholesterol	Prescription for cholesterol control	7	2024-02-11
8	Losartan	Hypertension	Prescription for blood pressure	8	2024-02-12

medicalrecord 6 x

```

294 • INSERT INTO medicalrecord (drugs, diagnos, pdetails, idPatient, date)
295   VALUES ('Paracetamol, Ibuprofen', 'Fever', 'Patient complaints of high temperature', 11, '2024-02-01');
296
297 • select * from medicalrecord
298

```

Result Grid

recordId	drugs	diagnos	pdetails	idPatient	date
15	NULL	NULL	NULL	NULL	2024-02-09
16	NULL	NULL	NULL	NULL	2024-02-10
17	NULL	NULL	NULL	NULL	2024-02-11
18	NULL	NULL	NULL	NULL	2024-02-12
19	NULL	NULL	NULL	NULL	2024-02-13
20	NULL	NULL	NULL	NULL	2024-02-14
21	Paracetamol, ...	Fever	Patient complaints of high temper...	11	2024-02-01
*	NULL	NULL	NULL	NULL	NULL

medicalrecord 7 medicalrecord 8 x

3) Write necessary queries to fetch required details of a particular patient.

```

299 • select * from patient where idPatient='1';
300

```








Result Grid

idPatient	PAddress	PContactNo	PName	PSEX
1	123 Main Street	9876543210	Aarav Sharma	Male
*	NULL	NULL	NULL	NULL

301 • `select * from patient where PName='Aarav Sharma';`

302

<

Result Grid  Filter Rows: Edit:    Export/Import:   Wrap Cell Content: 

	idPatient	PAddress	PContactNo	PName	PSex
▶	1	123 Main Street	9876543210	Aarav Sharma	Male
•	NULL	NULL	NULL	NULL	NULL

4) Write necessary queries to prepare bill for the patient at the end of checkout

303 • `Select`

304 `p.idPatient,`

305 `p.PName,`

306 `p.PAddress,`

307 `p.PContactNo,`

308 `SUM(m.Amount) AS totalBill`

309 `from`

310 `patient p`

311 `join`




312 `pharmacy m ON p.idPatient = m.idPatient`

313 `where`

314 `p.idPatient = 1;`

315

<

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	idPatient	PName	PAddress	PContactNo	totalBill
▶	1	Aarav Sharma	123 Main Street	9876543210	50

5) Write necessary queries to fetch and show data from various related tables (Joins)

```
303 • select
304     p.idPatient,
305     p.PName,
306     p.PAddress,
307     p.PContactNo,
308     m.recordID,
309     m.drugs,
310     m.diagnos,
311     m.pdetails,
312     m.date
313 from
314     patient p
315 join
316     medicalrecord m ON p.idPatient = m.idPatient
317 where
318     p.idPatient = 1;
```

< Filter Rows: Export: Wrap Cell Content:

	idPatient	PName	PAddress	PContactNo	recordID	drugs	diagnos	pdetails	date
▶	1	Aarav Sharma	123 Main Street	9876543210	1	Paracetamol	Common Cold	Prescription for flu symptoms	2024-02-05

6) Optimize repeated read operations using views/materialized views.

```
371 • CREATE VIEW patient_view AS
372     SELECT
373         idPatient,
374         PName,
375         PAddress,
376         PContactNo,
377         PSex
378     FROM
379         patient;
380
381 • SELECT * FROM patient_view WHERE PName = 'Aarav Sharma';
382
```

< Filter Rows: Export: Wrap Cell Content:

	idPatient	PName	PAddress	PContactNo	PSex
▶	1	Aarav Sharma	123 Main Street	9876543210	Male

7) Optimize read operations using indexing wherever required. (Create index on at least 1 table)

- Without indexing:

```
394 • SET profiling=1;
395 • select * from medicalrecord where recordid='1';
396 • SHOW PROFILES;
397
```

Query_ID	Duration	Query
1	0.00433250	SHOW WARNINGS
2	0.00061050	select * from medicalrecord where recordid='1' ...
3	0.00048000	SET profiling=1
4	0.00461100	SHOW WARNINGS
5	0.00406375	select * from medicalrecord where recordid='1' ...

- With indexing:

```
399 • SET profiling=1;
400 • CREATE INDEX idx_medicalrecord_recordID ON medicalrecord (recordid);
401 • SHOW PROFILES;
```

Query_ID	Duration	Query
1	0.00433250	SHOW WARNINGS
2	0.00061050	select * from medicalrecord where recordid='1' ...
3	0.00048000	SET profiling=1
4	0.00461100	SHOW WARNINGS
5	0.00406375	select * from medicalrecord where recordid='1' ...
6	0.00034925	SET profiling=1
7	0.00522825	SHOW WARNINGS
8	0.10496325	CREATE INDEX idx_medicalrecord_recordID ON ...

8) Try optimizing bill generation using stored procedures.

```
DELIMITER //
```

```
CREATE PROCEDURE generateBillForMedicalRecord(IN recordid INT)
BEGIN
    DECLARE totalAmount INT;
    SELECT COALESCE(SUM(charges), 0)
    INTO totalAmount
    FROM medicalrecord
    WHERE recordid = recordid;

    -- Insert the generated bill into the 'bill' table
    INSERT INTO bill (recordid, Patient_name, amount)
    SELECT
        m.recordid,
        p.PName,
        totalAmount
    FROM
        medicalrecord m
    INNER JOIN
        patient p ON m.idPatient = p.idPatient
```

9) Add necessary triggers to indicate when patients medical insurance limit has expired.

```
476 CREATE TRIGGER before_insert_medicalrecord
477 BEFORE INSERT ON medicalrecord
478 FOR EACH ROW
479 BEGIN
480     DECLARE insurance_limit DECIMAL(10,2);
481
482     -- Get the insurance coverage limit for the patient
483     SELECT coverageLimit
484     INTO insurance_limit
485     FROM insurance
486     WHERE idPatient = NEW.idPatient;
487
488     -- Check if the coverage limit is exceeded
489     IF insurance_limit <= 0 THEN
490         SIGNAL SQLSTATE '45000'
491         SET MESSAGE_TEXT = 'Insurance coverage limit exceeded. Cannot insert medical record.';
492     END IF;
493 END
494
495 DELIMITER ;
```

Output

#	Time	Action	Message	Duration / Fetch
15	05:50:16	CREATE TRIGGER before_insert_medicalrecord BEFORE INSERT ON medicalrecord FOR ...	0 row(s) affected	0.016 sec
16	05:50:16	CREATE TRIGGER before_update_medicalrecord BEFORE UPDATE ON medicalrecord FO...	0 row(s) affected	0.015 sec

Activate Windows
Go to Settings to activate Windows.

References -

<https://medium.com/@sarahalalawi0/hospital-management-database-3d19dc240446>

https://www.w3schools.com/sql/sql_view.asp

Aarya Teli

B.Tech (CSE)

Kolhapur Institute of Technology's

College of Engineering (Autonomous) Kolhapur