HO CHI MINH CITY UNIVERSITY OF SCIENCE, HO CHI MINH CITY NATIONAL UNIVERSITY

ADVANCED DATABASE SUBJECT

PHASE 1 - ADVANCED DATABASE DESIGN

Student made: 22127069 - Nguyễn Đặng Hoàng Dinh
22127107 - Nguyễn Thế Hiển
Lecturer in charge: Hồ Thị Hoàng Vy

SEMESTER III – SCHOOL YEAR 2024-2025

GROUP DETAILS INFORMATION TABLE

Group code:	**21VP.ADB.03**	
Number of members	2	
MSSV	**Full name**	**Email**
22127069	Nguyễn Đặng Hoàng Dinh	ndhdinh22@clc.fitus.edu.vn
22127107	Nguyễn Thế Hiển	nthien22@clc.fitus.edu.vn

Work assignment & completion evaluation table		
Work to be done	**The performer**	**Level of completion**
	22127069 - Nguyễn Đặng Hoàng Dinh	
	22127107 - Nguyễn Thế Hiển	

PROJECT REQUIREMENTS - EXERCISES

Type of exercise	☑ **Theory** • **Practice** • **Project** ☑ **Exercise**
Start date	**27/05/2025**
End date	**16/06/2025**

A. Project/Assignment Requirements

Clinic system management

Phase 1: Students research and describe in detail the process, data, related constraints, list of functions to be built with corresponding transaction frequency (frequency information is self-observed, analyzed and proposed by students), additional functions and functional constraints can be added. Design data at conceptual and logical levels (apply theoretical knowledge).

B. Result

I. Business process

Customer:
1. Register / Log in to the system (Web / App)
2. Select a doctor by specialty or suggested system.
3. View the doctor's available time slot → select the appropriate time.
4. The system confirms the schedule → sends notification via SMS / email / app.
5. The schedule is recorded in the system, creating a record in the Appointment table.
6. The receptionist checks and supports (if necessary).
7. The patient comes for examination, the doctor creates a medical record after the examination is completed.

Receptionist:
1. Receives the arrival (check-in)
2. Creates / pulls the patient's record.
3. Transfers the record to the doctor.
4. Receives the results after the examination.
5. Prints / saves the results and invoices

Doctor:
1. Views the assigned personal work schedule.
2. Accesses the patient's record.
3. Enter examination data (ICD-10, prescriptions, notes).
4. Update follow-up examination schedule if any.

Accounting:
1. Summarize the number of doctor examinations and staff working hours.
2. Apply salary calculation formula for doctors and staff (by examination, KPI, allowance).
3. Create monthly payroll.
4. Export reports by clinic or the whole system.

Clinic management:
1. Approve shift assignments for doctors.
2. Monitor capacity / vacant rooms.

3. Issue overload warnings.

Chain management board:

1. View KPI summary by day / week / month / quarter
2. Compare performance between facilities
3. Make decisions to open new / adjust resources.

II. List of functions and transaction frequency

Function	**User**	**Describe**	**Frequency (estimated)**
Register/Login account	Customer	Customer creates account and logs in to website/app	50-100 times/day
Make an appointment	Customer	Schedule, view time slots	300-500 times/day
Cancel/edit schedule	Customer	Change schedule	30-50 times/day
View medical records	Customer	View examination history	100-200 times/day
Create patient profile	Receptionist	Create/pull patient records, update contact information	100–150 visits/day/room
Support booking and guest check-in	Receptionist	Support customers to make appointments	100–150 visits/day/room
Doctor's schedule management	Clinic Management	Create shifts, edit work schedules	20–50 times/day
Enter medical records	Doctor	Enter patient diagnosis and examination information	25–40 visits/day/person
Salary and remuneration management	Accountant	Enter and approve doctor and staff salaries and wages	~1,000 views/month
Export financial reports	Accountant	Export financial reports (weekly / monthly)	~3-4 times/month

View overall statistics report	Chain Management	Weekly/Monthly/ Quarterly Statistics	~10–20 visits/month

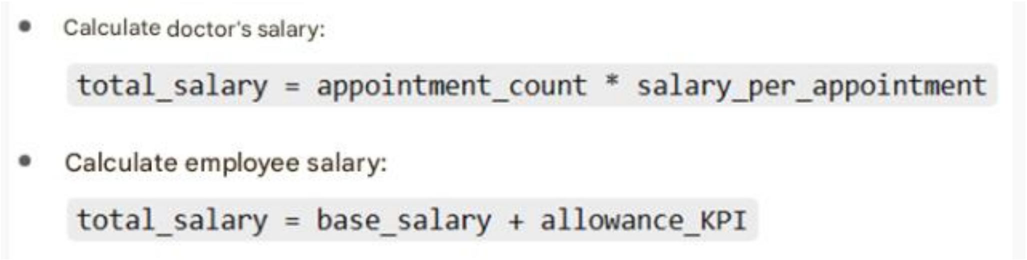
III. Data description and binding

Entity	Main properties	Main constraint
Patient	patient_id PK, fullname, dob, gender, address, phone (UNIQUE), email (UNIQUE), bhyt_info, created_at	● Phone/email format is valid ● Each patient_id has only one BHYT_info ● No null fullname/dob/gender ● Phone and email must be unique.
Account_Patient	account_id PK, patient_id FK, username (UNIQUE), password_hash, last_login, created_at	● patient_id must exist, each patient has only 1 account, username cannot be null
Employee (supertype) (Specialization)	emp_id PK, fullname, email (UNIQUE), phone, address, employee_type ∈ {D,S,R,K,M}	● email format, employee_type must be in the set {D=Doctor, S=Staff, R=Receptionist, K=Accountant, M=Manager} ● An Employee other than the Doctor role works for only one clinic.
Doctor (subtype of Employee)	doctor_id PK = emp_id FK, salary_per_appointment	● 1 doctor can have many specialties ● salary_per_appointment ≥ 0
Specialty	specialty_id PK, name (UNIQUE), description	● name is not null, unique ● 1 specialty can belong to many doctors
Staff(subtype of Employee)	staff_id PK = emp_id FK, position, clinic_id FK,	● base_salary >= 0 ● Each staff only works

	base_salary	at one clinic
Clinic	clinic_id PK, name (UNIQUE), address, opening_hours, status ∈ {active,inactive}	● Each facility has a unique address ● status is not null
Schedule	schedule_id PK, doctor_id FK, clinic_id FK, day_of_week, start_time, end_time, max_patients, manager_id	● No overlap with same doctor_id ● max_patients > 0 ● start_time < end_time ● manager_id must be Employee with role “Manager” ● Number of bookings ≤ max_patients/day
Appointment	app_id PK, patient_id FK, schedule_id FK, scheduled_time, status ∈ {booked,confirmed,canceled,completed}, created_at	● scheduled_time must be within the schedule's working time. ● status ∈ {‘booked’, ‘confirmed’, ‘cancelled’, ‘completed’} ● 1 patient has only 1 uncompleted appointment on 1 schedule
Medical_Record	record_id PK, app_id FK UNIQUE, diagnosis_code (ICD-10), prescription, notes, follow_up_date, created_at	● app_id must exist ● only 1 medical_record per appointment ● diagnosis_code must be valid according to ICD-10
Salary_Record (supertype) (Generalization)	(emp_id FK, month)->PK, calculated_at, total_salary, ketoan_id(FK)	● emp_id must exist ● total_salary >= 0 ● each (emp_id,month) only 1 record ● month must be in ‘YYYY-MM’ format ● ketoan_id must be Employee with role “Accountant”
Doctor_Salary_Record	(emp_id, month)->PK = FK	● appointment_count

	Salary_Record, appointment_count	>=0
Staff_Salary_Record	(emp_id, month)->PK = FK Salary_Record, allowance_KPI	● allowance_KPI >= 0

Salary calculation formula for doctors and staff::



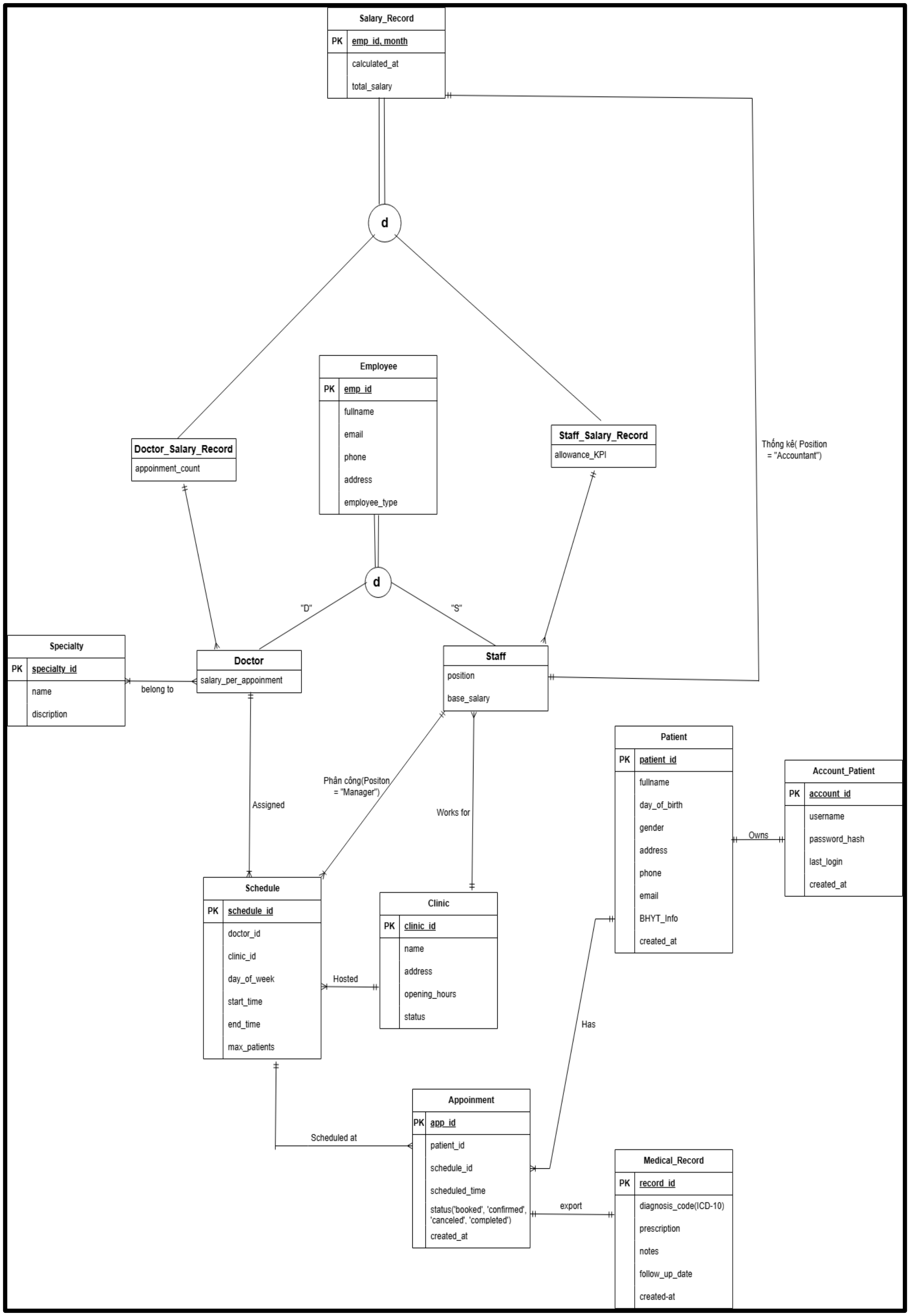
- Calculate doctor's salary:

```
total_salary = appointment_count * salary_per_appointment
```

- Calculate employee salary:

```
total_salary = base_salary + allowance_KPI
```

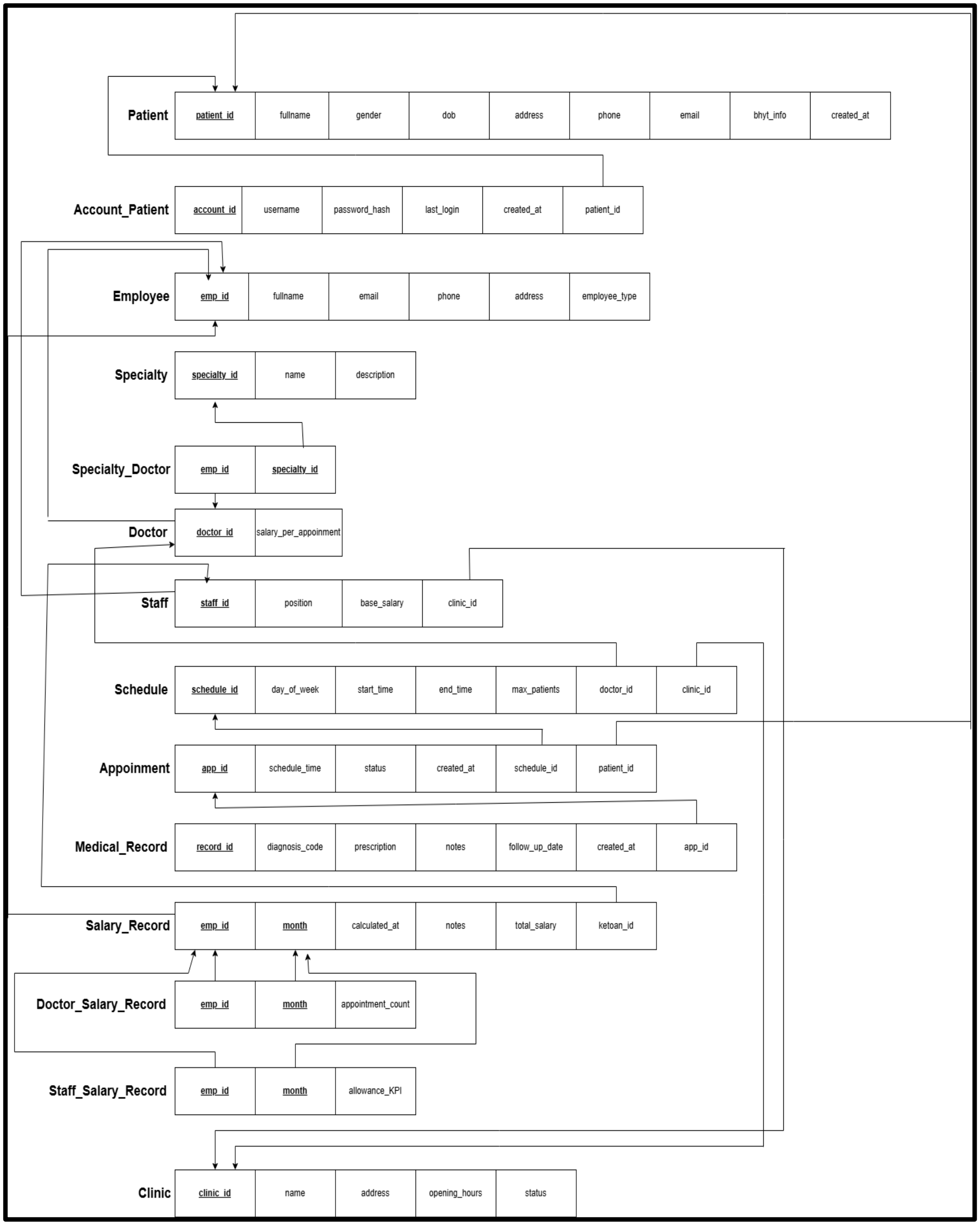
IV. ER diagram



See clearer image (open with draw.io) at:

https://app.diagrams.net/#G19wD6SRqrodMXPW_qwtEPcy-HJshKWAE8#%7B%22pageId%22%3A%22K0bcpYg1qCPGSAQD1Ldf%22%7D

V. Relationship diagram



See clearer image (open with draw.io) at:

https://app.diagrams.net/#G19wD6SRqrodMXPW_qwtEPcy-HJshKWAE8#%7B%22pageId%22%3A%22P-dOSpbg9qXmgmomRKJV%22%7D

VI. Standard form assessment

1. PATIENT

- Source set: patient_id

- Intermediate set: Ø

- Target set: fullname, day_of_birth, gender, address, phone, email, BHYT_info, created_at

-> Key: patient_id

→ The schema meets BCNF because:

+ There are no transitive or multivalued functional dependencies

+ All non-key attributes are fully functionally dependent on patient_id

2. ACCOUNT_PATIENT

- Source set: account_id

- Intermediate set: Ø

- Target set: username, password_hash, last_login, created_at, patient_id

-> Key: account_id

→ The schema meets BCNF because:

+ account_id is the only key that identifies all information

+ patient_id is a foreign key but does not violate functional dependencies

3. EMPLOYEE

- Set source: emp_id

- Intermediate set: Ø

- Target set: fullname, email, phone, address, employee_type

-> Key: emp_id

→ The schema meets BCNF because:

+ There are no non-key functional dependencies

+ All attributes describing employees are fully dependent on emp_id

4. DOCTOR (subtype of EMPLOYEE)

- Source set: emp_id

- Intermediate set: Ø

- Target set: salary_per_appointment

  -> Key: emp_id

  → The schema meets BCNF because:

    + The emp_id uniquely determines the medical examination salary

    + There are no dependencies that violate

5. STAFF (subtype of EMPLOYEE)

- Source set: emp_id

- Intermediate set: Ø

- Target set: position, base_salary

  -> Key: emp_id

  → The schema meets BCNF because:

    + No transitive or partial dependencies

6. CLINIC

- Source set: clinic_id

- Intermediate set: Ø

- Target set: name, address, opening_hours, status

  -> Key: clinic_id

  → The schema meets BCNF because:

    + All attributes are completely dependent on the primary key

7. SPECIALTY

- Source set: specialty_id

- Intermediate set: Ø

- Target set: name, description

  -> Key: specialty_id

→ The schema meets BCNF because:

+ No functional dependencies between name and description

8. DOCTOR_SPECIALTY (INTERMEDIATE TABLE N:N)

- Source set: (doctor_id, specialty_id)

- Intermediate set: Ø

- Target set: —

-> Key: (doctor_id, specialty_id)

→ The schema meets BCNF because:

+ It is a linked table, does not contain additional descriptive attributes

9. SCHEDULE (WORK SCHEDULE)

- Source set: schedule_id

- Intermediate set: Ø

- Target set: doctor_id, clinic_id, day_of_week, start_time, end_time, max_patients, manager_id

-> Key: schedule_id

→ The schema meets BCNF because:

+ All attributes are fully functionally dependent on schedule_id

10. APPOINTMENT (APPOINTMENT)

- Source set: app_id

- Intermediate set: Ø

- Target set: patient_id, schedule_id, scheduled_time, status, created_at

-> Key: app_id

→ The schema meets BCNF because:

+ No non-key dependencies exist

+ The states cannot determine the schedule other than app_id

11. MEDICAL_RECORD (PROFILE MEDICAL RECORD)

- Source set: record_id

- Intermediate set: Ø

- Target set: app_id, diagnosis_code, prescription, notes, follow_up_date, created_at

    -> Key: record_id

    → The schema meets BCNF because:

        + record_id identifies all medical record information

12. SALARY_RECORD (EMPLOYEE SALARY – COMMON)

- Source set: (emp_id, month)

- Intermediate set: Ø

- Target set: calculated_at, total_salary

    -> Key: (emp_id, month)

    → The schema meets BCNF because:

        + No attribute depends on emp_id or month separately