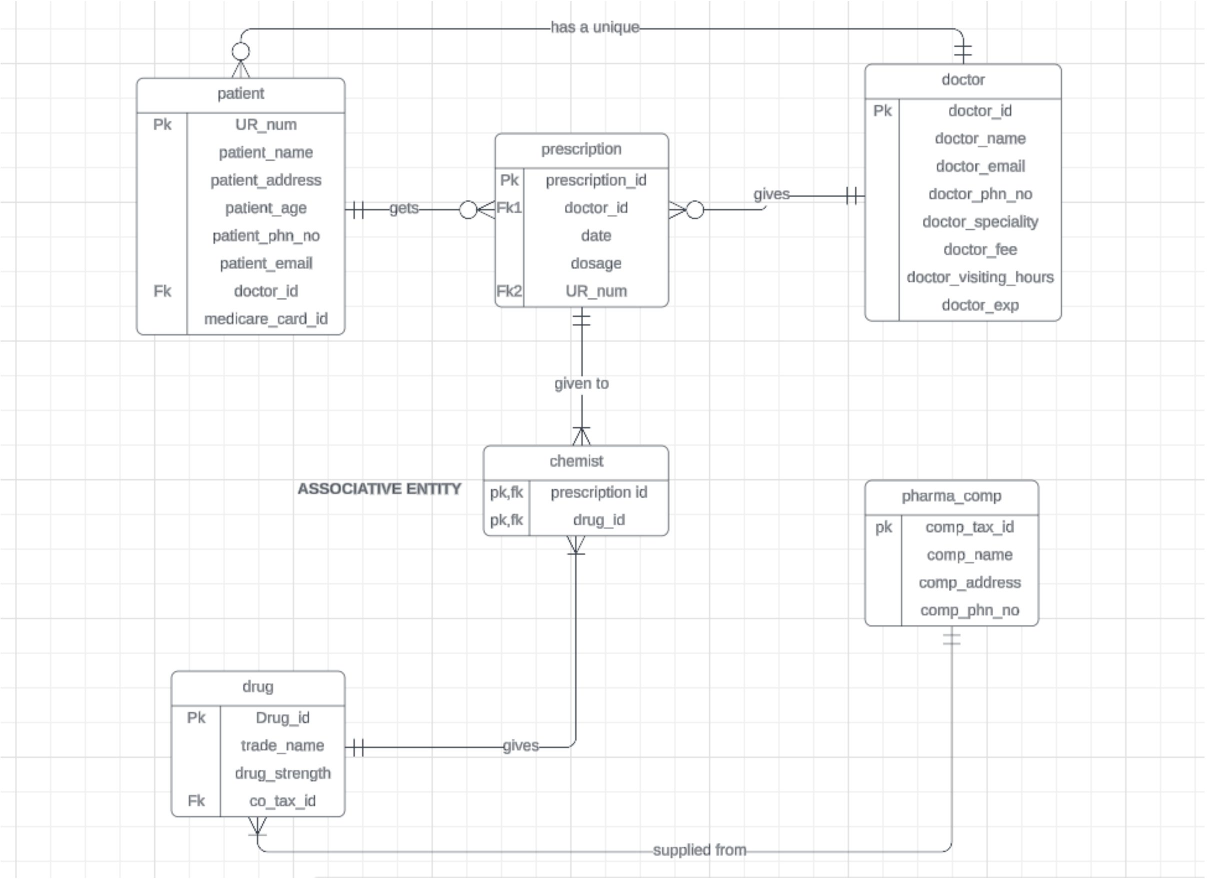
SIT-103 DATABASE FUNDAMENTALS:

TASK 3.1P – ENTITY RELATIONSHIP DIAGRAM



ASSUMPTIONS:

- Chemist is an associative entity
- Every patient has a unique doctor
- For this task I assumed that there is a single pharma company
- Doctor fee is an entity which shows how much fee the doctor charges
- Doctor visiting hours is an entity which indicates what times the doctor is available
- Prescription id is created to uniquely identify drugs
- Comp_tax_id is another unique identifier in the pharma company.

1. Identify all Entities, their Attributes, Constraints, and Relationships with Cardinalities

- ENTITIES: patient , prescription , doctors, chemist, pharma comp. , drug.

  Attributes for patient :

  A. UR_num
  B. Patient_name
  C. Patient_address
  D. Patient_age
  E. Patient_phn_no
  F. Patient_email
  G. Doctor_id
  H. Medicare_card_id

Attributes for prescription

A. Prescription_id
B. Doctor_id
C. Date
D. Dosage
E. UR_number

Attributes for doctors

A. Doctor_id
B. Doctor_name
C. Doctor_email
D. Doctor_phone_no
E. Doctor_speciality
F. Doctor_exp
G. Doctor_visiting_hours
H. Doctor_fee

Attribute for chemist

A. Prescription_id
B. Drug_id

Attribute for pharma company

A. Comp_tax_id
B. Comp_name
C. Comp_address
D. Comp_phn_no

Constraints and Relationships

1. Patient and Prescription

- **Cardinality**: A patient can have multiple prescriptions, but each prescription is associated with exactly one patient.
- **Constraint**: UR_number in Prescription must exist in the UR_num of Patient.

2. Prescription and Doctors

- **Cardinality**: A prescription is written by one doctor, but a doctor can write multiple prescriptions.
- **Constraint**: Doctor_id in Prescription must exist in Doctor_id of Doctors.

3. Doctors and Patient

- **Cardinality**: A doctor can have multiple patients, but a patient is associated with one doctor at a time.
- **Constraint**: Doctor_id in Patient must exist in Doctor_id of Doctors

.

4. Chemist and Prescription

- **Cardinality**: A chemist handles multiple prescriptions, but each prescription is handled by one chemist.
- **Constraint**: Prescription_id in Chemist must exist in Prescription_id of Prescription.

5. Chemist and Drug

- **Cardinality**: A chemist can have multiple drugs, and a drug can be available at multiple chemists.
- **Constraint**: Drug_id must exist in the list of drugs available in the Chemist.

6. Pharma Company and Drug

- **Cardinality**: A pharma company can manufacture multiple drugs, but each drug is manufactured by one pharma company.

- **Constraint**: Drug_id in Chemist must be linked to a pharma company via a drug record, though the specific relationship to pharma companies isn't explicitly stated in the attributes.

2. . Implement and label all relationships using Primary/Foreign Key (PK/FK) or Associative Entities

Entities and Primary/Foreign Keys

1. Patient

- **Primary Key (PK)**: UR_num
- **Foreign Key (FK)**: Doctor_id (refers to Doctor)

2. Prescription

- **Primary Key (PK)**: Prescription_id
- **Foreign Keys (FK)**:
  - Doctor_id (refers to Doctor)
  - UR_number (refers to Patient)

3. Doctors

- **Primary Key (PK)**: Doctor_id

4. Chemist

- **Foreign Keys (FK)**:
  - Prescription_id (refers to Prescription)
  - Drug_id (refers to Drug)

5. Pharma Company

- **Primary Key (PK)**: Comp_tax_id

Associative Entities and Relationships

1. Patient and Prescription

- **Patient** (UR_num PK) ↔ **Prescription** (UR_number FK)
- **Constraint**: UR_number in Prescription must be a valid UR_num in Patient.

2. Prescription and Doctors

- **Prescription** (Doctor_id FK) ↔ **Doctors** (Doctor_id PK)
- **Constraint**: Doctor_id in Prescription must be a valid Doctor_id in Doctors.

3. Doctors and Patient

- **Doctors** (Doctor_id PK) ↔ **Patient** (Doctor_id FK)
- **Constraint**: Doctor_id in Patient must be a valid Doctor_id in Doctors.

4. Chemist and Prescription

- **Chemist** (Prescription_id FK) ↔ **Prescription** (Prescription_id PK)
- **Constraint**: Prescription_id in Chemist must be a valid Prescription_id in Prescription.

5. Chemist and Drug

- This relationship requires an associative entity to manage the many-to-many association.
- **Associative Entity**: Chemist_Drug
  - **Primary Key (PK)**: Composite key of Prescription_id and Drug_id
  - **Foreign Keys (FK)**:
    - Prescription_id (refers to Prescription)
    - Drug_id (refers to Drug)
- **Constraint**: Each entry in Chemist_Drug must have valid Prescription_id and Drug_id.

6. Pharma Company and Drug

- **Pharma Company** (Comp_tax_id PK) ↔ **Drug** (Drug_id FK)
- **Constraint**: This relationship assumes each drug is associated with one pharma company. (Note: Drug_id is not explicitly given in the initial attributes, so this relationship is inferred based on common data modeling practices.)

JUSTIFICATION:

The database schema for Barwon Health's prescription system is designed to ensure effective data management and integrity by using unique primary keys for each entity to eliminate redundancy. Relationships between entities, such as Patient to Prescription and Doctor to Prescription, are structured to mirror real-world interactions, facilitating accurate tracking of prescriptions and doctor-patient dynamics. The schema integrates each entity with unique identifiers to streamline data management and reduce redundancy. The connections between Patients, Prescriptions, and Doctors reflect their real-world interactions, enabling precise tracking of prescriptions and doctor-patient relationships. Foreign keys throughout the schema uphold data integrity, ensuring seamless and efficient management of the prescription system.