# **Intermediate Project Report**

# Danielle Park, Gaurav Gulati, Rhett Bramfield

Faculty of Science, University of Calgary

CPSC 471: Database Management System

Prof. Reda Alhajj

February 12, 2024

### **Entities and Attributes**

* An **entity type** represents a real-world object or concept in our database that we want to store data about.
* Each **entity type** has attributes (properties or characteristics).
* For our **walk-in clinic database**:
  + **Patient**: Represents the clinic’s clients. Each Patient is uniquely identified by a PatientID.(with attributes like PatientID, Name, DOB,Age)
  + **Doctor**: Represents a medical practitioner at the clinic, identified by a unique DoctorID. (DoctorID, Name, Specialization)
  + **Appointment (Weak Entity) :** Represents a particular time booked for a patient and doctor. (AppointmentID(Partial key) , Date, Time)
  + **Clinic Staff :** Represents administrative users who manage scheduling, billing, and other operational tasks **(** StaffID, Name, Role, Contact Info**)**
  + **DoctorSchedule :** Captures the available time slots for each Doctor, ensuring that appointments are scheduled only when the doctor is available. **(**ScheduleID, DoctorID, Date, Start Time, End Time, Status**)**
  + **Prescription :** Records the medications or treatment instructions provided during an Appointment. (PrescriptionID, Medication, Dosage)
  + **MedicalRecord**: Documents the clinical details of an Appointment, including the diagnosis and treatment prescribed to the Patient.(RecordID, Diagnosis, Treatment)
  + **Billing**: Tracks the billing details associated with an Appointment, including the cost and payment status.(BillingID, Amount, PaymentStatus)

### **b. At least one (1) of your entity types must be weak**

* **Appointment as a Weak Entity:**In our system, the **Appointment** entity is modeled as a weak entity because:
  + **Incomplete Identification:**The attributes of Appointment (such as Date and Time) are insufficient on their own to uniquely identify an appointment. Therefore, Appointment requires additional context from the related Patient entity.
  + **Composite Primary Key:**To uniquely identify each Appointment, we use a composite key that combines the **PatientID** (from the Patient entity) with the **AppointmentID** (a partial key within the Appointment entity). This composite key *(PatientID, AppointmentID)* ensures that each Appointment is uniquely tied to a specific Patient.
  + **Mandatory Relationship:**Every Appointment must be associated with a Patient. This total participation (or mandatory dependency) ensures that no Appointment exists without a corresponding Patient, which maintains referential integrity within the database.

**Rationale and Benefits:**

* **Data Integrity:**By modeling Appointment as a weak entity, we enforce a strong dependency on the Patient entity. This prevents orphaned Appointment records, i.e., appointments that exist without an associated patient, thus maintaining data integrity.
* **Logical Accuracy:**The real-world scenario naturally dictates that an appointment cannot exist without a patient. Modeling Appointment as a weak entity accurately reflects this dependency, ensuring that the system's design aligns with actual clinic operations.
* **Simplified Maintenance:**The composite key structure (PatientID, AppointmentID) simplifies the maintenance of the database. It clearly outlines the relationship between patients and their appointments, making it easier to manage insertions, deletions, and updates while preserving referential integrity.

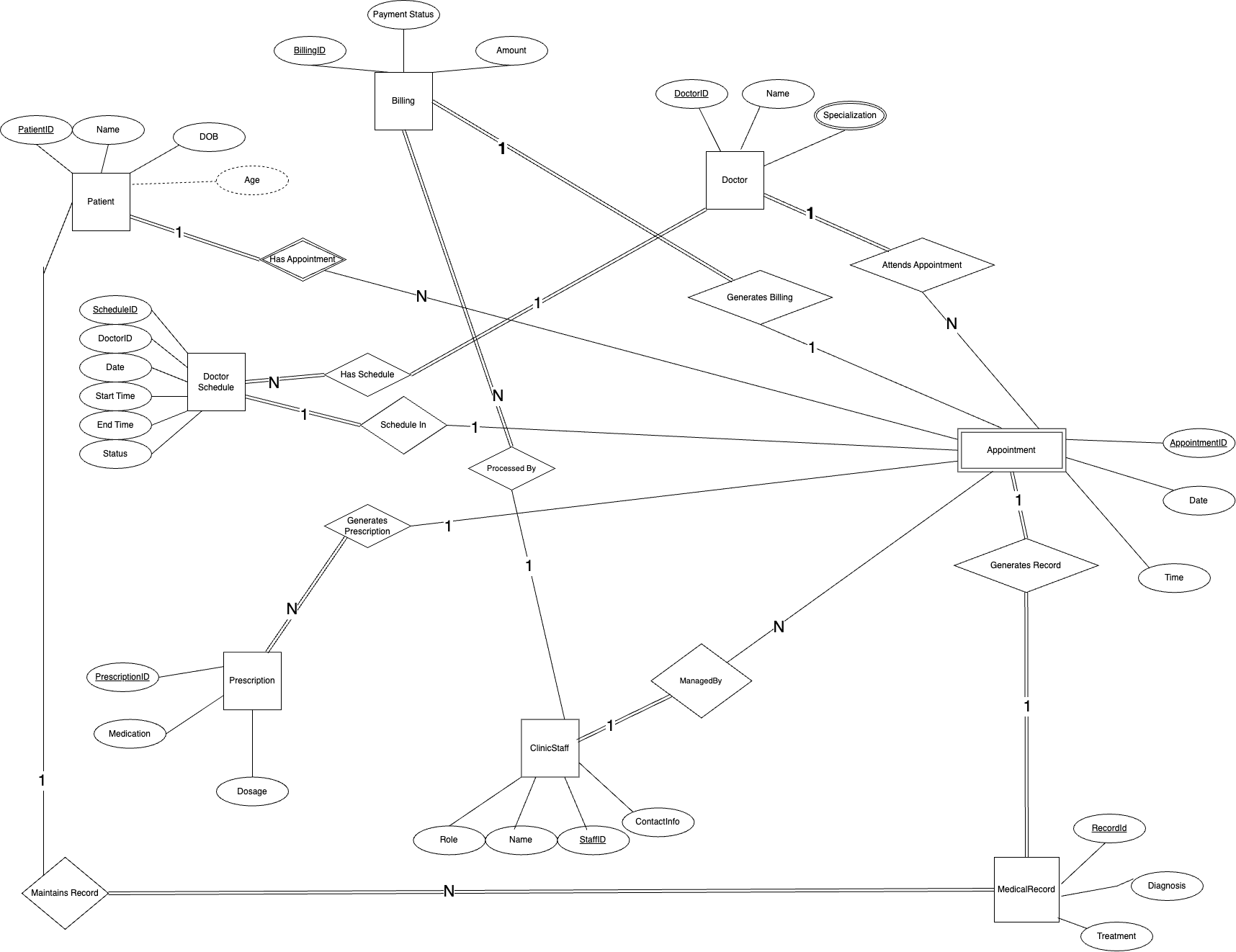
## **Relationships and Cardinalities**

1. **Patient — Appointment (HasAppointment)**
   * **Description:** A Patient books one or more Appointments.
   * **Cardinality:** One Patient : Many Appointments
   * **Note:** Appointment is weak; its identification depends on the Patient who books it.
2. **Doctor — Appointment (AttendsAppointment)**
   * **Description:** A Doctor attends to many Appointments, and each Appointment is handled by one Doctor.
   * **Cardinality:** One Doctor : Many Appointments
3. **Appointment — Clinic Staff (ManagedBy)**
   * **Description:** Each Appointment is scheduled and managed by a Clinic Staff member (who may handle rescheduling, cancellations, etc.).
   * **Cardinality:** One Clinic Staff : Many Appointments
4. **Doctor — DoctorSchedule (HasSchedule)**
   * **Description:** A Doctor can have multiple entries in the DoctorSchedule that indicate their available time slots.
   * **Cardinality:** One Doctor : Many DoctorSchedule entries
5. **DoctorSchedule — Appointment (ScheduledIn)**
   * **Description:** Each Appointment is scheduled in a specific DoctorSchedule slot to ensure it falls within the Doctor’s available hours.
   * **Cardinality:** Typically One DoctorSchedule slot : One Appointment
   * *(Adjustments can be made if a schedule slot can accommodate multiple short appointments.)*
6. **Appointment — Prescription (GeneratesPrescription)**
   * **Description:** An Appointment may result in one or more Prescriptions being issued.
   * **Cardinality:** One Appointment : Many Prescriptions
7. **Appointment — MedicalRecord (GeneratesRecord)**
   * **Description:** Each Appointment generates a corresponding MedicalRecord that documents the consultation’s details.
   * **Cardinality:** One Appointment : One MedicalRecord
8. **Patient — MedicalRecord (MaintainsRecord)**
   * **Description:** A Patient can have multiple MedicalRecords over time, reflecting their historical clinical data.
   * **Cardinality:** One Patient : Many MedicalRecords
9. **Appointment — Billing (GeneratesBilling)**
   * **Description:** Each Appointment results in a Billing record that captures the cost and payment status.
   * **Cardinality:** One Appointment : One Billing
10. **Billing — Clinic Staff (ProcessedBy)**
    * **Description:** Billing records are processed (verified, updated, or resolved) by Clinic Staff.
    * **Cardinality:** One Clinic Staff : Many Billing records

## **End-User Types**

* **Patients (Clients):**
  + Use the system to book appointments, view their medical records, and track billing information.
* **Clinic Staff (Administrators):**
  + Manage appointments, update doctor schedules, process billing, and review prescriptions and medical records.

**ERD Diagram**

****

## 

## 

## 

## 

## **Summary of Design Assumptions**

* **Weak Entity Designation:**
  + **Appointment** is designated as a weak entity. Although it has an AppointmentID, it is contextually dependent on the Patient who books the appointment (and potentially the DoctorSchedule slot) for complete identification.
* **Integration of Clinical Operations:**
  + The system integrates appointment scheduling, medical record keeping, prescription management, and billing, ensuring a comprehensive approach to clinic management.
* **Clear Role Separation:**
  + Two distinct end-user roles are defined (Patients and Clinic Staff) to ensure appropriate access and operations management within the system.