Database Schema design for a Relational Information System

Silvana Greca
Department of Informatics
Faculty of Natural Sciences
University of Tirana

Overview - Neurological Patient Information System

- Understand the problem domain (neurological patients)
- Explain the need for a relational information system
- Present the data dictionary
- Describe retrieval and maintenance requirements
- Show the ER database schema
- Explain design decisions
- Present Oracle DDL examples

Problem Domain Overview

- Neurological patients generate:
  - Demographic data
  - Clinical examinations
  - Neurological assessments
  - Diagnoses
  - Imaging studies (CT, MRI – DICOM)
- Data is:
  - Heterogeneous
  - Time-dependent
  - Clinically sensitive

Challenges in Managing Neurological Data

- Large amount of data per patient
- Multiple examinations over time
- Multiple diagnoses per patient
- Combination of clinical and imaging data
- Need for reliable long-term storage

Why This Is an Information System Problem

- Data must be:
  - Structured
  - Consistent
  - Searchable
- Manual systems or flat files:
  - Cause redundancy
  - Lead to inconsistencies
  - Do not scale
- A database-centered IS is required

What Is an Information System (IS)?

- An Information System includes:
  - Data storage
  - Data retrieval
  - Data maintenance
  - Support for decision making
- Database is the core component
- Application logic builds on top of the databas

Why a Relational Database?

- Strong data integrity guarantees
- Clear representation of relationships
- Support for complex queries (SQL)
- Transactions and consistency
- Well suited for healthcare systems

Why Oracle DBMS?

- Enterprise-grade relational DBMS
- Advanced constraint support
- Strong transaction management
- Widely used in healthcare and research
- Suitable for large and complex datasets

Data Dictionary: Definition

- A data dictionary describes:
  - Entities (tables)
  - Attributes (columns)
  - Data types
  - Constraints
- Serves as documentation
- Ensures shared understanding of data

Core Entity: PATIENT

- Central entity of the system
- Stores patient demographic data
- Each patient has a unique identifier
- One patient can have:
  - Many exams
  - Many diagnoses
  - Many imaging studies

PATIENT: Key Attributes

- patient_id (Primary Key)
- first_name, last_name
- date_of_birth
- gender (controlled values)
- health_insurance_number (unique)
- address and timestamps

Clinical Data Entities

- PHYSICAL_EXAM
- NEUROLOGIC_ASSESSMENT
- VITAL_SIGN_TYPE
- VITAL_SIGN_READING
- INTERVENTION
- HISTOLOGY

These entities:

- Represent events in time
- Allow historical tracking of patient condition

Neurologic Assessment Data

- Symptoms:
  - Headache
  - Seizures
  - Weakness
  - Dizziness
- Neurological scales:
  - Glasgow Coma Scale (GCS)
- Stored separately to support multiple assessments

Diagnosis Modeling

- Diagnosis is a medical concept:
  - Stored once with a unique code
- Patients may have:
  - Multiple diagnoses
  - Diagnoses at different times
- Modeled using an associative entity

PATIENT_DIAGNOSIS Entity

- Links PATIENT and DIAGNOSIS
- Stores:
  - Diagnosis date
  - Certainty level
  - Notes
- Supports medical history tracking

Dictionary Tables

- RISK_FACTOR
- ALLERGY
- COMPLAINT
- MEDICATION
- FAMILY_HISTORY_CONDITION

Purpose:

- Avoid data duplication
- Ensure consistent terminology
- Support filtering and analysis

Data Retrieval Requirements

The system must support:

- Patient-based queries
- Diagnosis-based queries
- Risk factor and allergy queries
- Clinical history retrieval
- Imaging metadata retrieval

Example Retrieval Scenarios

- Retrieve full patient profile
- Find all patients with a specific diagnosis
- Identify patients with certain risk factors
- Retrieve all exams for a patient over time
- List all MRI studies for a patient

Why Retrieval Requirements Matter

- Influence table structure
- Influence relationship design
- Influence indexing strategy
- Ensure efficient query execution

Data Maintenance Requirements

- Insert new patients and exams
- Update clinical information
- Add new diagnoses
- Maintain controlled vocabularies
- Preserve historical data

Data Integrity and Validation

- Primary keys ensure uniqueness
- Foreign keys ensure referential integrity
- Check constraints enforce medical rules
- Example:
  - Gender values
  - GCS score ranges

Conceptual Design: ER Model (Chen)

- Used for conceptual modeling
- Entities represented as rectangles
- Relationships as diamonds
- Attributes as ellipses
- Foreign keys are NOT shown as attributes

ER Diagram: Patient & Clinical Data

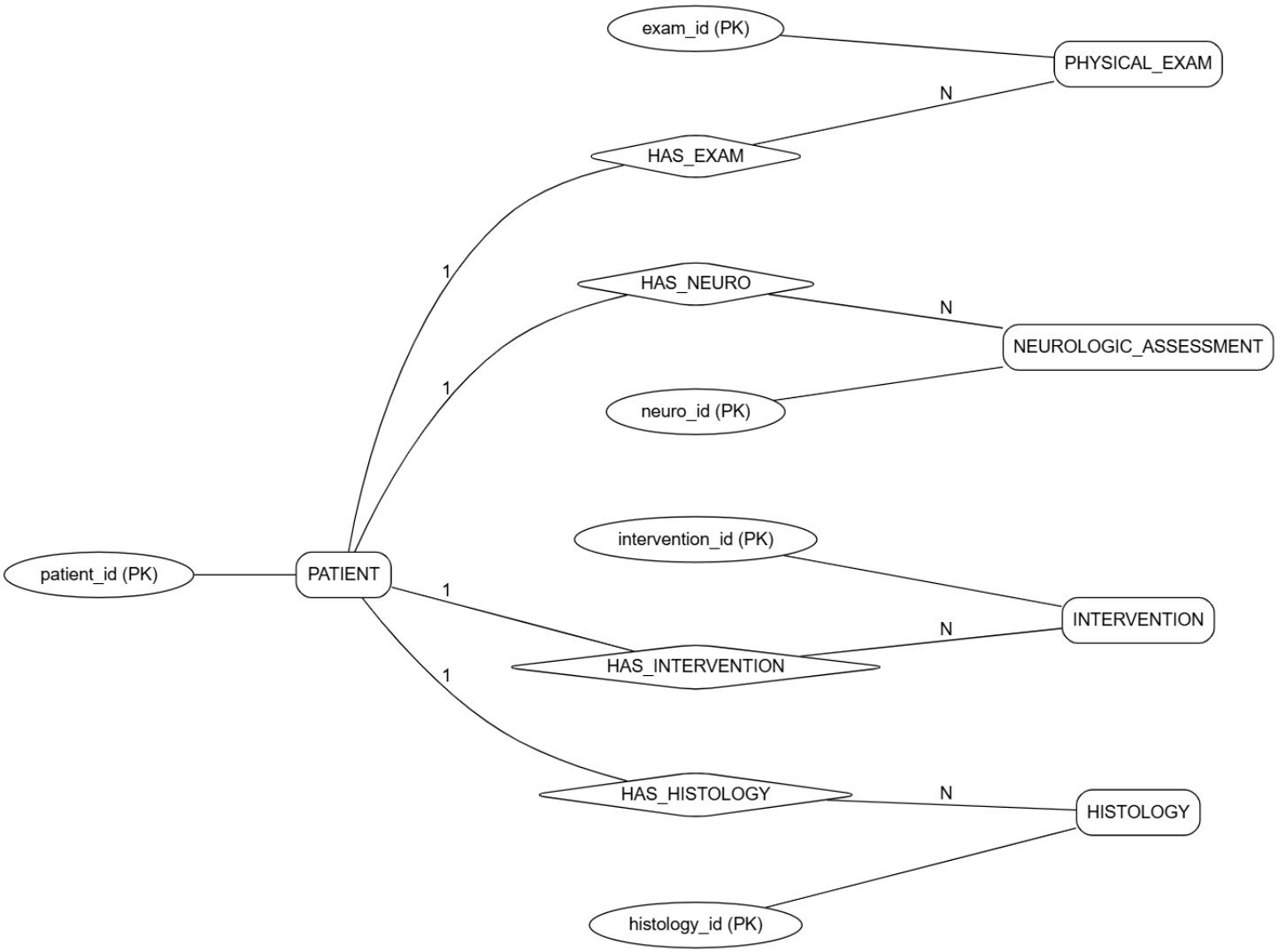
- One patient → many exams
- One patient → many neurologic assessments
- One patient → many interventions
- Captures temporal nature of clinical data

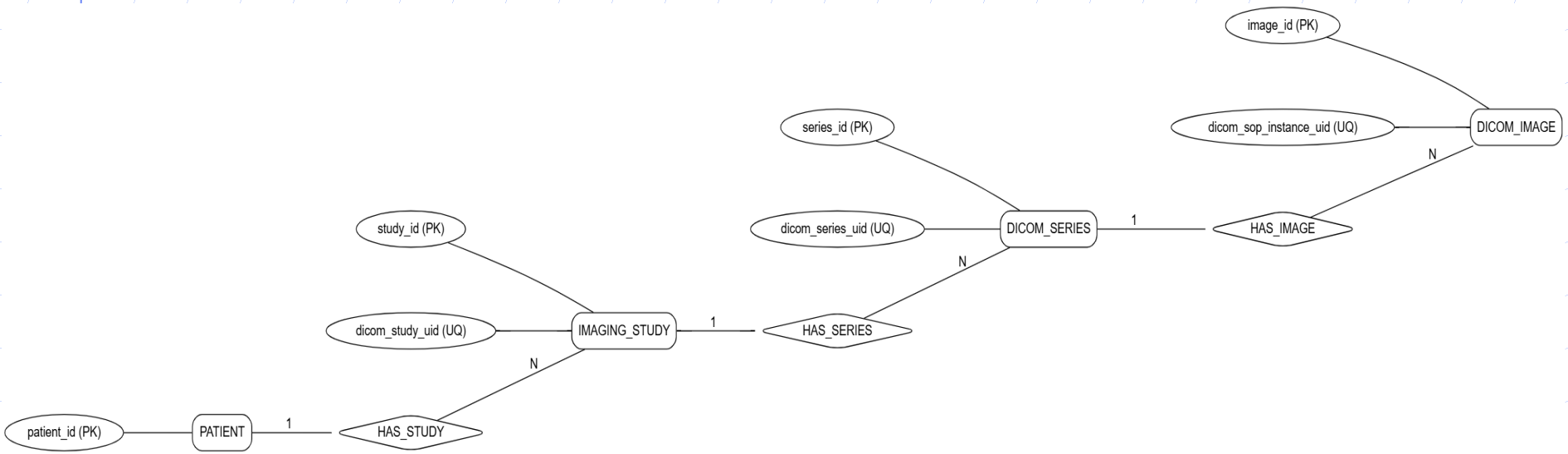
ER Diagram: Diagnosis

- Patient linked to diagnosis via associative entity
- Supports many-to-many relationship
- Allows diagnosis history tracking

ER Diagram: Imaging (DICOM)

- Hierarchical structure:
  - Patient → Study → Series → Image
- Matches DICOM standard
- Imaging metadata stored separately from clinical data


exam_id (PK)
PHYSICAL_EXAM
N
HAS_EXAM
1
HAS_NEURO
N
NEUROLOGIC_ASSESSMENT
1
neuro_id (PK)
intervention_id (PK)
patient_id (PK)
PATIENT
1
INTERVENTION
N
HAS_INTERVENTION
1
HAS_HISTOLOGY
N
HISTOLOGY
histology_id (PK)


image_id (PK)
series_id (PK)
dicom_sop_instance_uid (UQ)
DICOM_IMAGE
N
study_id (PK)
dicom_series_uid (UQ)
DICOM_SERIES
1
HAS_IMAGE
N
dicom_study_uid (UQ)
IMAGING_STUDY
1
HAS_SERIES
N
patient_id (PK)
PATIENT
1
HAS_STUDY

Design Rationale

- Normalization to reduce redundancy
- Separation of clinical and imaging data
- Temporal modeling of events
- Use of associative entities
- Focus on data integrity

Logical Design to Physical Design

- ER model → relational schema
- Entities → tables
- Relationships → foreign keys
- Constraints enforce business rules

Oracle DDL Overview

- CREATE TABLE statements
- PRIMARY KEY constraints
- FOREIGN KEY constraints
- CHECK constraints
- UNIQUE constraints

Example Oracle DDL

```
CREATE TABLE patient (
  patient_id NUMBER GENERATED AS IDENTITY PRIMARY KEY,
  first_name VARCHAR2(60) NOT NULL,
  last_name VARCHAR2(60) NOT NULL,
  date_of_birth DATE NOT NULL,
  gender CHAR(1) CHECK (gender IN ('M','F','O')),
  health_insurance_no VARCHAR2(40) UNIQUE
);
```

Benefits of the Proposed System

- High data consistency
- Reliable retrieval
- Supports clinical and research needs
- Scalable and extensible
- Suitable for healthcare environments

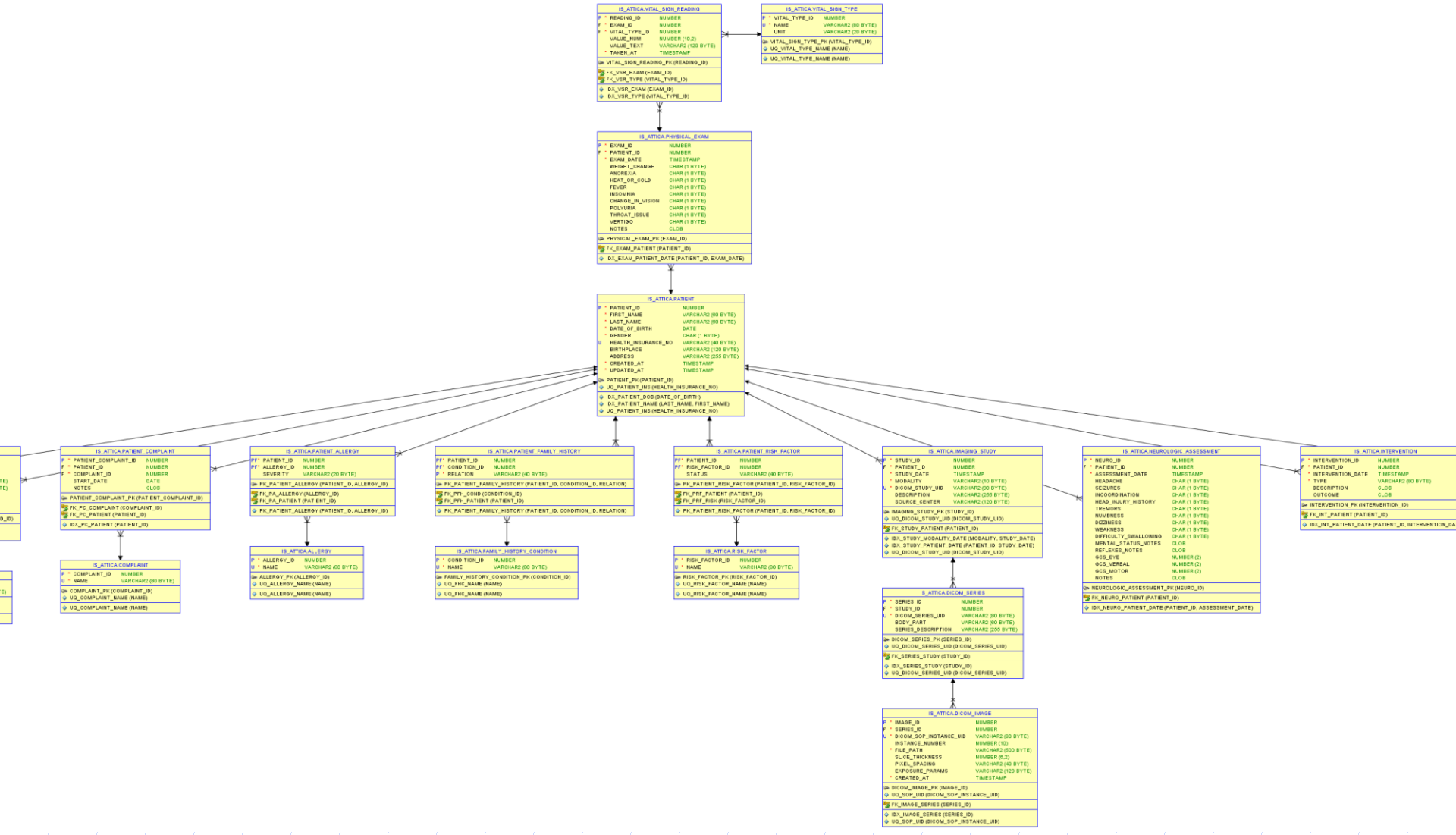
Limitations and Future Extensions

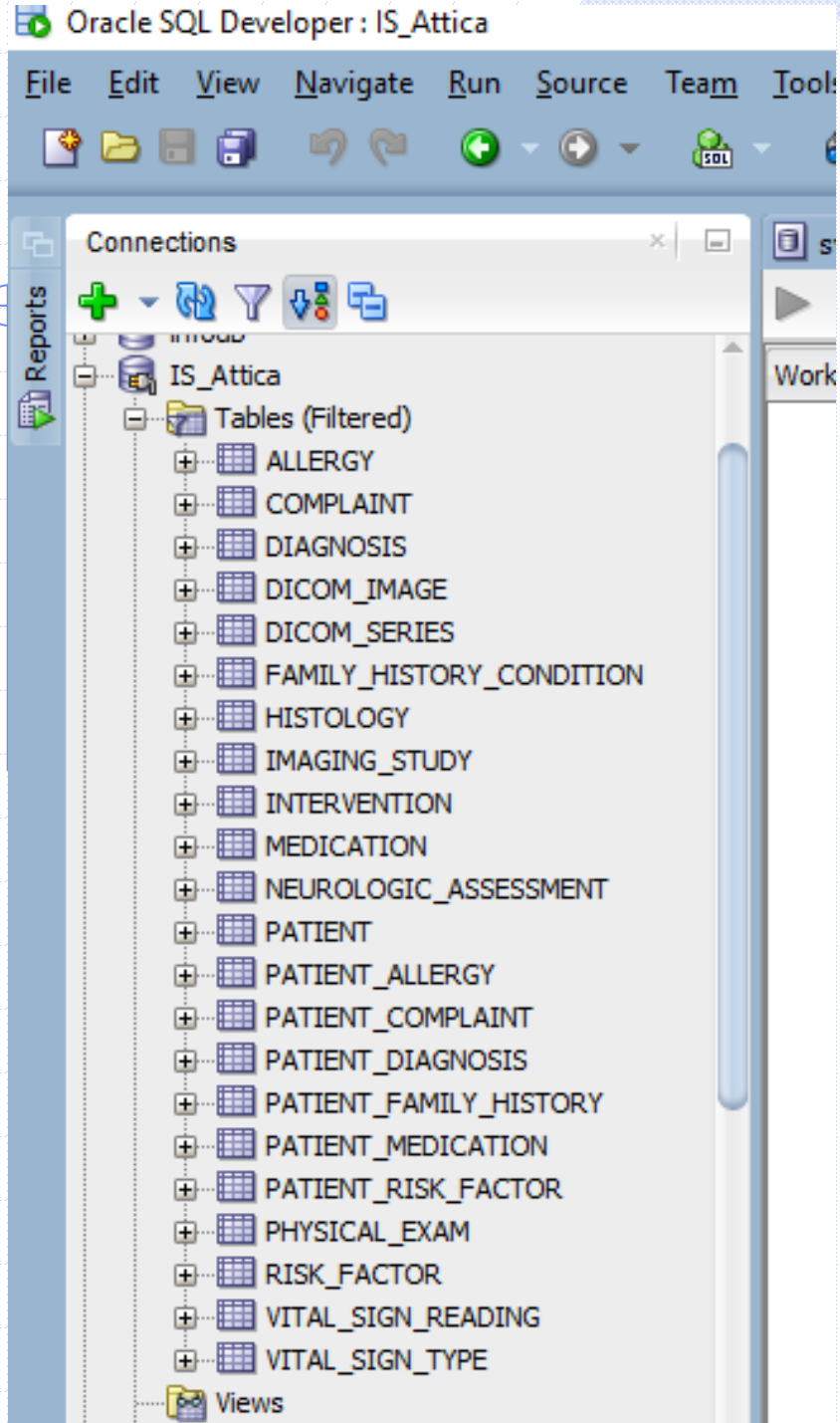
- Multimedia feature extraction not implemented
- No user roles or access control
- Performance tuning required for large datasets
- Future integration with CBIR systems

```
CREATE TABLE patient (
  patient_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  first_name VARCHAR2(60) NOT NULL,
  last_name  VARCHAR2(60) NOT NULL,
  date_of_birth DATE NOT NULL,
  gender CHAR(1) NOT NULL,
  health_insurance_no VARCHAR2(40),
  birthplace VARCHAR2(120),
  address VARCHAR2(255),
  created_at TIMESTAMP DEFAULT SYSTIMESTAMP NOT NULL,
  updated_at TIMESTAMP DEFAULT SYSTIMESTAMP NOT NULL,
  CONSTRAINT uq_patient_ins UNIQUE (health_insurance_no),
  CONSTRAINT chk_patient_gender CHECK (gender IN ('M','F','O'))
);

CREATE INDEX idx_patient_name ON patient(last_name, first_name);
CREATE INDEX idx_patient_dob  ON patient(date_of_birth);
```

```
CREATE TABLE risk_factor (
  risk_factor_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  name VARCHAR2(80) NOT NULL,
  CONSTRAINT uq_risk_factor_name UNIQUE (name)
);

CREATE TABLE patient_risk_factor (
  patient_id NUMBER NOT NULL,
  risk_factor_id NUMBER NOT NULL,
  status VARCHAR2(40),
  CONSTRAINT pk_patient_risk_factor PRIMARY KEY (patient_id, risk_factor_id),
  CONSTRAINT fk_prf_patient FOREIGN KEY (patient_id) REFERENCES
patient(patient_id),
  CONSTRAINT fk_prf_risk FOREIGN KEY (risk_factor_id) REFERENCES
risk_factor(risk_factor_id)
);
```

Visual Database Schema




Oracle SQL Developer : IS_Attica
File Edit View Navigate Run Source Team Tools
Connections
Reports
IS_Attica
Tables (Filtered)
ALLERGY
COMPLAINT
DIAGNOSIS
DICOM_IMAGE
DICOM_SERIES
FAMILY_HISTORY_CONDITION
HISTOLOGY
IMAGING_STUDY
INTERVENTION
MEDICATION
NEUROLOGIC_ASSESSMENT
PATIENT
PATIENT_ALLERGY
PATIENT_COMPLAINT
PATIENT_DIAGNOSIS
PATIENT_FAMILY_HISTORY
PATIENT_MEDICATION
PATIENT_RISK_FACTOR
PHYSICAL_EXAM
RISK_FACTOR
VITAL_SIGN_READING
VITAL_SIGN_TYPE
Views

Conclusion

- A complete relational IS design was presented
- Supports both retrieval and maintenance
- Based on sound database design principles
- Ready for future extensions

Questions

- **Thank you for your attention**
- Questions?