Sruthi H Vellore (hv234)
Abhijeet Sahdev (as4673)
Subodh Dahal (sd983)

Group 9
CS 631 004 DMSD

Deliverable 2

Goal of this Phase:

The objective of this phase of the project is to refine the Enhanced Entity-Relationship (EER) conceptual model developed in Phase 1 and translate it into a relational schema for the Morris Health Services database. This involves revising the EER diagram based on feedback and new considerations, and then mapping the updated entities, relationships, and attributes to tables, columns, primary keys, and foreign keys in the relational schema.
Ensure that the mapping from EER to the relational schema:

- Preserves all information.
- Maintain the constraints as much as possible.
- Minimizes null values in the database.

Document any keys, constraints, etc., that go beyond what can be represented in the relational schema.
Highlight any difficulties faced.

Revisions from Phase 1:

Several significant changes have been made to the EER diagram based on feedback and new insights:

Dropped Attributes and Relationships:

- Removed the attributes Located_At and Appointments and transformed Makes_Appointment into a quaternary relationship involving Facility instead of Outpatient Surgery.
- Introduced Invoice_Detail for each appointment, leading to the inclusion of Appears_On as a strong entity.

Changes in Cardinalities:

- Few (min, max) and cardinalities have been adjusted based on assumptions made during the design process.

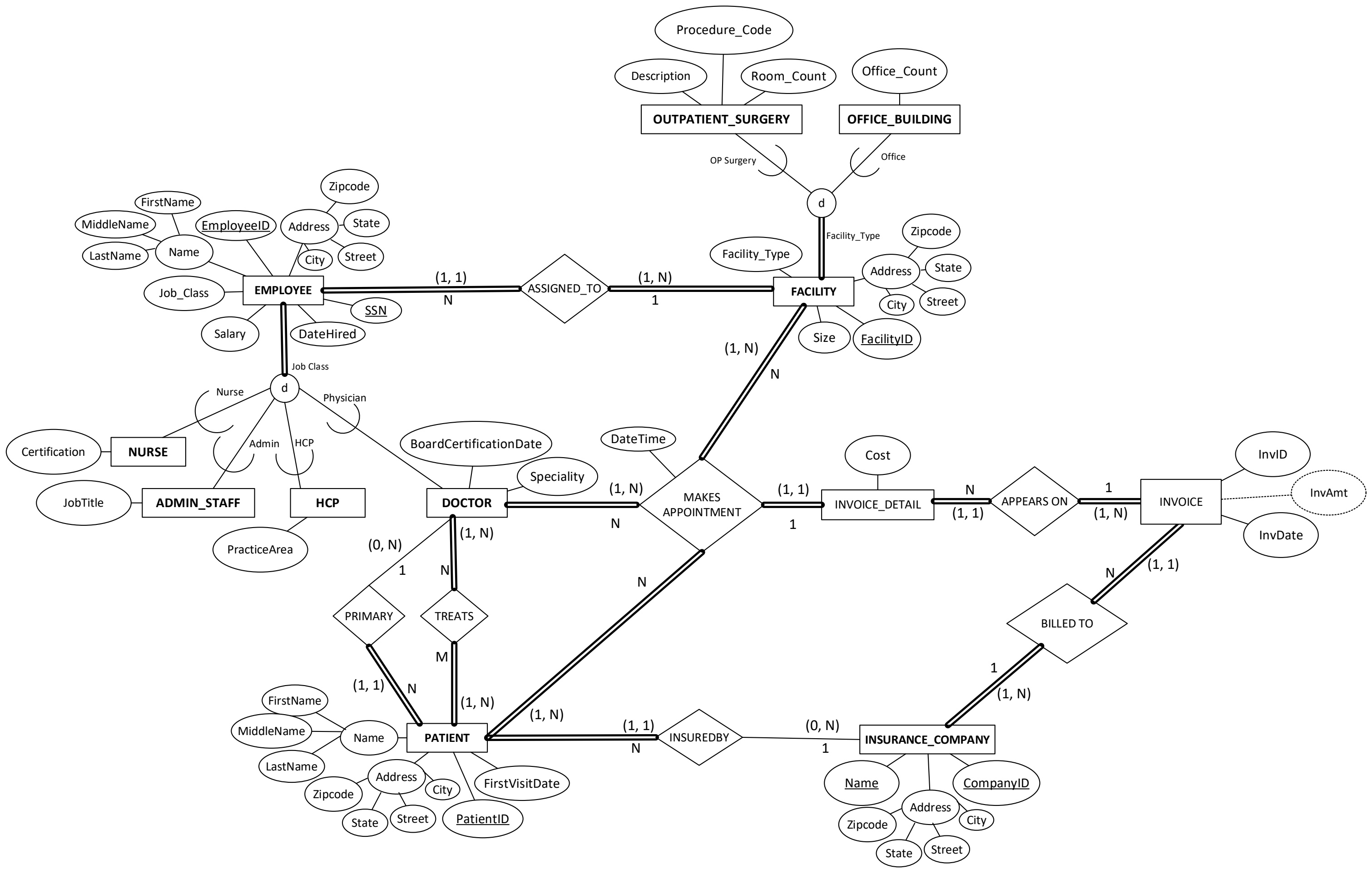
Normalization of Attributes:

- Doctor’s Specialty and Nurse’s Certification are no longer treated as multivalued attributes.
- Procedures are no longer multivalued; instead, they are represented as a composite attribute with PCode and Description.

Other Changes:

- Dropped the Supervisors relationship.
- Introduced a new attribute, Job Class, for Employees, serving as the defining attribute of specialization (formerly JobType in the EER).
- Renamed certain relationships for clarity and consistency.
- Changed the Location attribute in Facility to Address, treating it as a composite attribute, and dropped it as a key.

Updated EER Diagram



EER to Relational Mapping:

We used the algorithm to map EER to Relational Schema as discussed in class.

- First, we created a new relation for each of the entities. This involved drawing a tuple for each relation, naming its header, and dividing it into different sections that represent attributes. If composite attributes were encountered, we represented their component attributes.
- Next, we represented the primary keys of each table by underlining them. A choice was made depending on the context of the miniworld to choose a primary key if there were multiple keys (EmployeeID as PK instead of SSN for EMPLOYEE).
- Then, we represented the superclass-subclass relationships as the entity Doctor participated in multiple relationships. This was our only **deviation** from the algorithm discussed in class.

Now, we moved onto mapping the relationships between entity types in the following order.

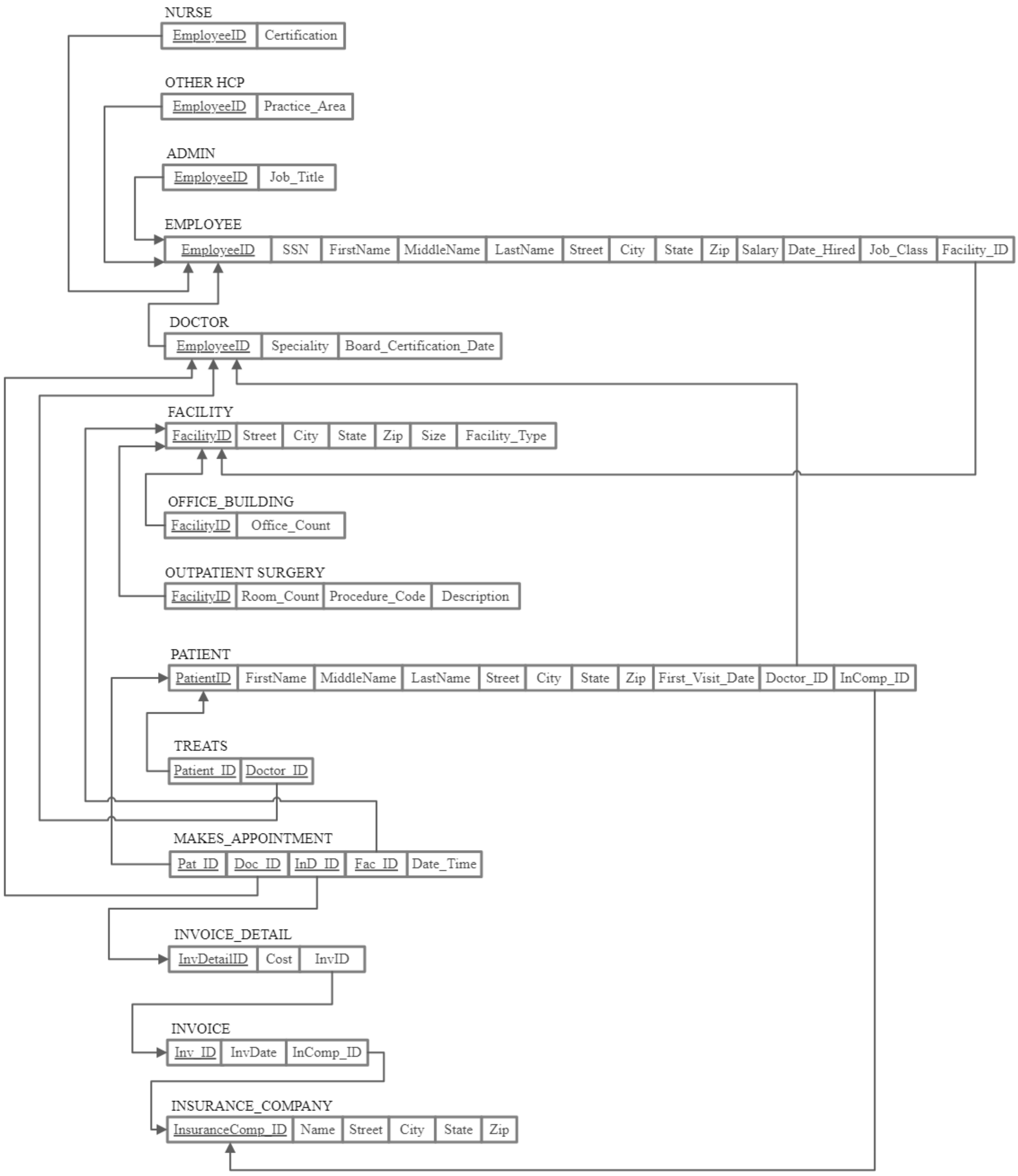
1. Binary Relationships in ER :

- 1:1 - none observed.
- 1:N - Foreign key approach that required updating the table of the relation on the N side by appending the primary key of the relation on 1 side as its foreign key.
- N:M - A relationship of relations was created with a composite primary key that included the primary keys of each of the relations.

2. N-ary Relationships in ER:

   Finally, we mapped the quaternary relationship (Makes_Appointment) by creating a new relation with a composite primary key that included primary keys of the participating relations as its foreign keys.
   In each of the cases discussed above in 1 and 2, an arrowed line was drawn from the referencing relationship (with the foreign key) to referenced relationship (primary key). We didn't have any multivalued attributes and weak entities here.

Relational Schema:


NURSE
EmployeeID | Certification
OTHER HCP
EmployeeID | Practice_Area
ADMIN
EmployeeID | Job_Title
EMPLOYEE
EmployeeID | SSN | FirstName | MiddleName | LastName | Street | City | State | Zip | Salary | Date_Hired | Job_Class | Facility_ID
DOCTOR
EmployeeID | Speciality | Board_Certification_Date
FACILITY
FacilityID | Street | City | State | Zip | Size | Facility_Type
OFFICE_BUILDING
FacilityID | Office_Count
OUTPATIENT SURGERY
FacilityID | Room_Count | Procedure_Code | Description
PATIENT
PatientID | FirstName | MiddleName | LastName | Street | City | State | Zip | First_Visit_Date | Doctor_ID | InComp_ID
TREATS
Patient_ID | Doctor_ID
MAKES_APPOINTMENT
Pat_ID | Doc_ID | InD_ID | Fac_ID | Date_Time
INVOICE_DETAIL
InvDetailID | Cost | InvID
INVOICE
Inv_ID | InvDate | InComp_ID
INSURANCE_COMPANY
InsuranceComp_ID | Name | Street | City | State | Zip

Constraints:

1. FACILITY :

- Validity constraint for Type, it can either be OP Surgery or Office. (Domain Constraint)
- Not null constraint for Type, Street, City, State, or Zipcode (Domain Constraint)

2. OFFICE BUILDING:

- Nonnegative constraint for Office_Count (Domain Constraint)

3. OUTPATIENT SURGERY:

- Nonnegative constraint for Room_Count (Domain Constraint)
- Not null constraint for Procedure_Code (Domain Constraint)

4. EMPLOYEE:

- Unique and not null constraint for SSN. (Domain Constraint)
- Not null constraint for FirstName and LastName (Domain Constraint)
- Not null constraint for Street, City, State, or Zipcode (Domain Constraint)
- Not null constraint for Date_Hired and FacilityID(Domain Constraint)
- Nonnegative constraint for Salary (Domain Constraint)
- Validity constraint for Job_Class (can have only 4 values) (Domain Constraint)

5. DOCTOR:

- Not null constraint for Specialty, BoardCertificationDate (Domain Constraint)

6. NURSE:

- Not null constraint for Certification (Domain Constraint)

7. HCP:

- Not null constraint for PracticeArea (Domain Constraint)

8. ADMIN_STAFF:

- Not null constraint for JobTitle (Domain Constraint)

9. PATIENT:

- Not null constraint for FirstName and LastName (Domain Constraint)
- Not null constraint for Street, City, State, or Zipcode (Domain Constraint)
- Not null constraint for DoctorID, InComp_ID and FirstVisitDate (Domain Constraint)

10. INVOICE DETAIL:

- Nonnegative constraint for Cost (Domain Constraint)
- Not null constraint for InvID (Domain Constraint)

11. INVOICE:

- Not null constraint for InComp_ID (Domain Constraint)

12. INSURANCE COMPANY:

- Not null and unique constraint for Name (Domain Constraint)
- Not null constraint for Street, City, State, or Zipcode (Domain Constraint)

13. OTHER:

- Not more than one appointment at the same time for a doctor, a patient, or a facility.
- Number of appointments at the same time does not exceed available Facilities at that time.
- Handling the minimum number of entities participating in a relationship.
- Maintaining the integrity of the database while executing statements.

Challenges Encountered:

Factoring in the changes from the EER uploaded on Canvas was a bit tedious but necessary. For example, adopting the quaternary relationship Makes_Appointment reduces the number of tables stored in our database.

Each of us had a different approach to represent the derived attribute, Inv_Amount. After much discussion, we decided to manage it by calculating it as we query the table.