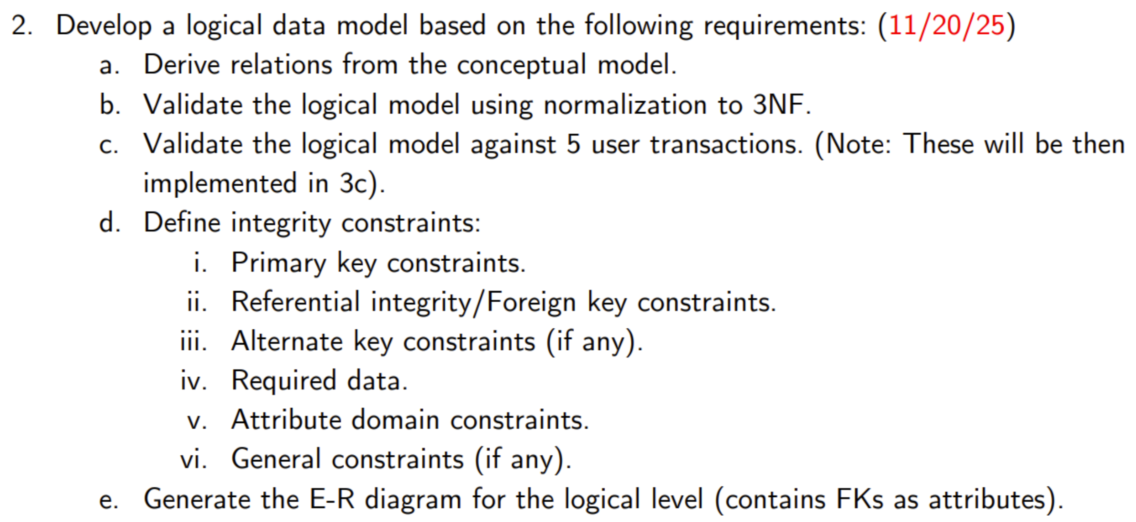
CSC423 Project Part 2

Dylan Zolnik

Kevin Marroquin



2. Develop a logical data model based on the following requirements: (11/20/25)
   a. Derive relations from the conceptual model.
   b. Validate the logical model using normalization to 3NF.
   c. Validate the logical model against 5 user transactions. (Note: These will be then implemented in 3c).
   d. Define integrity constraints:
      i. Primary key constraints.
      ii. Referential integrity/Foreign key constraints.
      iii. Alternate key constraints (if any).
      iv. Required data.
      v. Attribute domain constraints.
      vi. General constraints (if any).
   e. Generate the E-R diagram for the logical level (contains FKs as attributes).

a. Derive relations from the conceptual model
   - CLIENTS(clientNo PK, firstName, lastName, address, phoneNumber, preferences, feedback)
   - EMPLOYEES(staffNo PK, firstName, lastName, address, salary, phoneNumber)
   - EQUIPMENT(equipmentID PK, description, usage, cost, status)
   - REQUIREMENTS(requirementID PK, clientNo FK, startDate, startTime, duration, comments, priority)
     - ClientNo references clients(clientNo)
     - Startdate, starttime, and duration are separate attributes
   - REQUIRES_EQUIPMENT(requirementID FK, equipmentID FK, quantityNeeded, occasionsNeeded, totalOccasions)
     - Composite primary key is between requirementID and equipmentID
     - RequirementID references REQUIREMENTS(requirementID)
     - EquipmentID references EQUIPMENT(equipmentID)
   - ASSIGNED_TO(staffNo FK, requirementID FK, role)
     - Composite primary key between staffNo and requirementID
     - StaffNO references employees(staffNo)
     - RequirementID references REQUIREMENTS(requirementID)

b. Validate logic model using normalization to 3NF
   - All relations are 1NF
   - Clients

        - PK is clientNo
        - All others depend on clientNo
    - Employees
        - PK is staffNo
        - Others depend on this
    - Equipment
        - PK is equipmentID
        - Others depend on PK
    - Requirements
        - PK is requirementID
        - Same case
    - Requires_equipment
        - Composite PK of requirementID and equipmentID
        - Still valid in 3NF
    - Assigned_to
        - Composite PK of staffNo and requirementID
        - Valid in 3NF

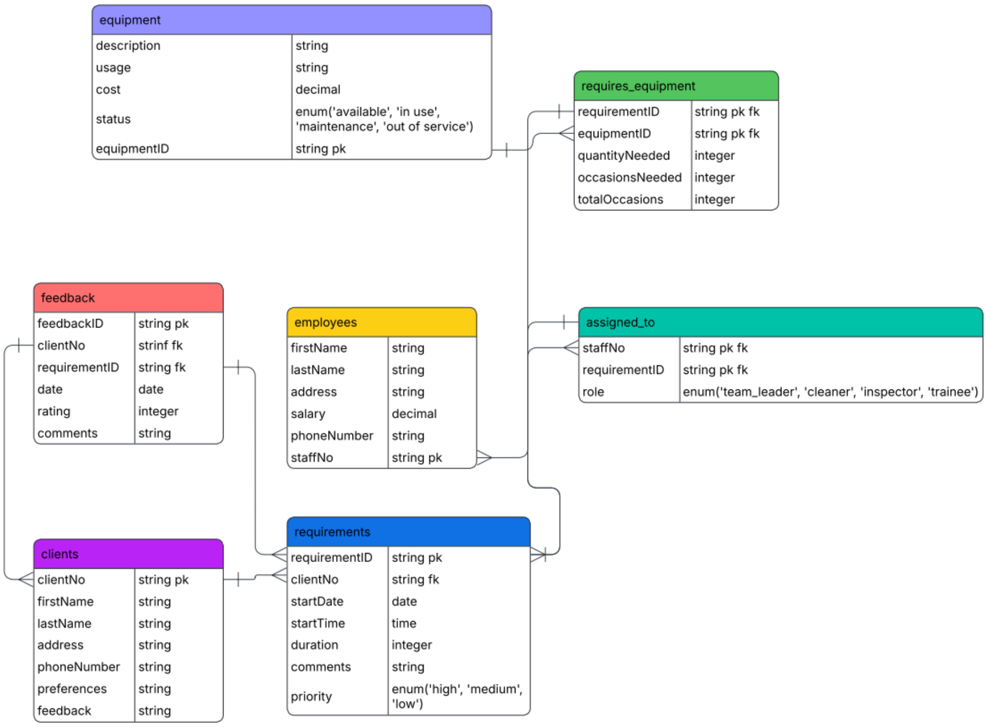
c. Validate the logic model against 5 user transactions
    1. Register new client and their preferences
        a. Insert into CLIENTS
        b. Create REQUIREMENTS that will reference clientNo
    2. Create a new requirement with date, time, duration and priority
        a. Insert into REQUIREMENTS
        b. System can schedule through other fields
    3. Specify equipment needed for a requirement
        a. Insert into REQUIRES_EQUIPMENT
        b. Schedule considers CLIENT(preferences) and EQUIPMENT(status)
    4. Assign staff to a requirement
        a. Insert into ASSIGNED_TO
        b. Manager query each requirement staffing
    5. Use client feedback to influence future employee assignment
        a. Query CLIENTS, REQUIREMENTS, ASSIGNED_TO, and EMPLOYEES to avoid staff with negative feedback

d. Define integrity constraints
    a. Primary key constraints
        i. ClientNo, staffNo, equipmentID, requirementID, (requirementID, equipmentID), (staffNo, requirementID) all must be unique and not NULL

b. Referential integrity/foreign key constraints
  i. REQUIREMENTS.clientNo -> CLIENTS.clientNo
  ii. REQUIRES_EQUIPMENT.requirementID -> REQUIREMENTS.requirementID
  iii. REQUIRES_EQUIPMENT.equipmentID -> EQUIPMENT.equipmentID
  iv. ASSIGNED_TO.staffNo -> EMPLOYEES.staffNo
  v. ASSIGNED_TO.requirementID -> REQUIREMENTS.requirementID
c. Alternate key constraints (if any)
  i. n/a
d. Required data
  i. CLIENTS: clientNo, firstName, lastName, address, phoneNumber
  ii. EMPLOYEES: staffNo, firstName, lastName, address, salary, phoneNumber
  iii. EQUIPMENT: equipmentID, description, cost, status
  iv. REQUIREMENTS: requirementID, clientNo, startDate, startTime, duration, priority
  v. REQUIRES_EQUIPMENT: requirementID, equipmentID, quantityNeeded, occasionsNeeded, totalOccasions
  vi. ASSIGNED_TO: staffNo, requirementID, role
  vii. Preferences, feedback, comments
e. Attribute domain constraints
  i. IDs
    1. Fixed length
  ii. Salary, cost
    1. Positive dec values
  iii. StartDate, startTime
    1. Valid dates
  iv. QuantityNeeded, occasionsNeeded, totalOccasions
    1. Pos int
    2. Occasions < total
  v. Status
    1. Can be available, in use, maintenance, out of service
  vi. Priority
    1. Can be high medium low
  vii. Role
    1. Team leader, cleaner, inspector, trainee
f. General constrains (if any)
  i. Each requirement needs one employee

ii. Each requirement needs one team leader

iii. Employees cannot be assigned to overlapping requirements



equipment

description	string
usage	string
cost	decimal
status	enum('available', 'in use', 'maintenance', 'out of service')
equipmentID	string pk

requires_equipment

requirementID	string pk fk
equipmentID	string pk fk
quantityNeeded	integer
occasionsNeeded	integer
totalOccasions	integer

feedback

feedbackID	string pk
clientNo	strinf fk
requirementID	string fk
date	date
rating	integer
comments	string

employees

firstName	string
lastName	string
address	string
salary	decimal
phoneNumber	string
staffNo	string pk

assigned_to

staffNo	string pk fk
requirementID	string pk fk
role	enum('team_leader', 'cleaner', 'inspector', 'trainee')

clients

clientNo	string pk
firstName	string
lastName	string
address	string
phoneNumber	string
preferences	string
feedback	string

requirements

requirementID	string pk
clientNo	string fk
startDate	date
startTime	time
duration	integer
comments	string
priority	enum('high', 'medium', 'low')