CSC623 Project 1
Part 2 - Logical Model
Michelle Manfrini, Sydney Mazat, Daniella Amador

Part 2: Develop a logical data model based on the following requirements:

a. Derive relations from the conceptual model.

Employee (empNo, emp_firstname, emp_lastname, address, salary, telNo) **Primary Key** staffNo **Alternate Key** telNo
Client (clientNo, client_fname, client_lname, client_address, client_te_No) **Primary Key** clientNo **Alternate Key** client_te_No
Equipment (equipNo, description, usage, cost) **Primary Key** EquipNo
Service (serviceID, clientNo, empNo, start_date, start_time, duration, equipNo, comments) **Primary Key** serviceID **Alternate Key** {clientNo, start_date} **Foreign Key** clientNo **references** Client(clientNo)
EmployeeService (empNo, serviceID) **Foreign Key** empNo **references** Employee(empNo) **Foreign Key** serviceID **references** Service(serviceID)
EquipmentService (equipNo, serviceID) **Foreign Key** equipNo **references** Equipment(equipNo) **Foreign Key** serviceID **references** Service(serviceID)

b. Validate the logical model using normalization to 3NF.

The model is already in third normal form. There are no partial dependencies, which are removed when transforming from the first to second normal form. Partial dependencies exist when at least two attributes are repeated in a relation. This can cause update anomalies. For example, if empNo and emp_firstname were both in the Service relation, there would be a partial dependency because emp_firstname is dependent on empNo. There are no transitive dependencies, which are removed when transforming from the second to third normal form. A transitive dependency is a dependency within a partial dependency. There is no repeated data in our model and there are no functional dependencies.

c. Validate the logical model against user transactions.

Transaction #1 - List the details of services assigned to a named employee.

The details of services held in the Service entity, and the details of staff who are assigned to services are held in the EmployeeService entity. In this case, we can use the Employee Performs Service relationship to produce the required list.

Transaction #2 - Modify Equipment cost.

Before updating cost, check that equipNo exists and validate that the modification adheres to attribute domain constraints.

Transaction #3 - Insert a service with an invalid clientNo.

Validate that the foreign key constraints prevent the insertion of a service with non-existent or invalid clientNo according to the attribute domain constraints.

Transaction #4 - List the details of equipment assigned to a service.

The details of services held in the Service entity and the details of equipment assigned to services are held in the EquipmentService entity. In this case, we can use the Service Uses Equipment relationship to produce the required list.

d. Define integrity constraints:

i. *Primary key constraints*

Primary keys must be unique, not null, and a strong entity.

ii. *Referential integrity/Foreign key constraints*

Relation	Foreign Key	Constraint	Reason
Service	clientNo	ON UPDATE CASCADE ON DELETE CASCADE	If there is no client associated with the service, then it should be deleted from the table because the company won't get paid, and that service doesn't exist anymore if the client doesn't.
EmployeeService	empNo	ON UPDATE CASCADE ON DELETE CASCADE	If an employee no longer works for the company (deleted from employee table), they should be removed from EmployeeService table because they won't be covering that service anymore.
EmployeeService	serviceID	ON UPDATE CASCADE ON DELETE CASCADE	If a service is deleted, it should be deleted from this table too because the employees don't need to work on it anymore.
EquipmentService	equipNo	ON UPDATE CASCADE ON DELETE CASCADE	If a piece of equipment no longer exists, then it should be deleted from this table because it can't be used on that service anymore.
EquipmentService	serviceID	ON UPDATE CASCADE ON DELETE CASCADE	If a service is deleted, it should be deleted from this table too because equipment isn't needed for it anymore.

iii. *Alternate key constraints*

Employee Table: telNo
Service Table: {clientNo, start_date}

These must be not null and unique.

iv. *Required data*

Relation	**Attributes that require data and must not hold a null value**
Employee	empNo, telNo
Client	clientNo, client_te_No
Equipment	equipNo, Cost
Service	serviceID, clientNo
EmployeeService	empNo, serviceID
EquipmentService	equipNo, serviceID

v. *Attribute domain constraints*

Relation	Attribute	Domain Constraint
Employee	empNo	numerical values with max of ten digits
	emp_firstname	character string with max of 15 characters
	emp_lastname	character string with max of 15 characters
	address	numerical values and character strings
	salary	numerical values with max of ten digits
	telNo	numerical values with only ten digits
Client	clientNo	numerical values with max of ten digits
	client_fname	character string with max of 15 characters
	client_lname	character string with max of 15 characters
	client_address	numerical values and character strings
	client_te_No	numerical values with only ten digits
Equipment	equipNo	numerical values with max of ten digits
	description	character string with max of 200 characters
	usage	character string with max of 100 characters
	cost	numerical values with max of ten digits
Service	serviceID	numerical values with max of ten digits
	clientNo	numerical values with max of ten digits
	start_date	date in format MM/DD/YYYY
	start_time	numerical values in format HH:MM
	duration	float
	comments	character string with max of 100 characters
EmployeeService	empNo	numerical values with max of ten digits
	serviceID	numerical values with max of ten digits
EquipmentService	equipNo	numerical values with max of ten digits
	serviceID	numerical values with max of ten digits

vi. *General constraints - Other*

1) EquipNo in Service can only be assigned to one ServiceID at one time (Can't be double booked for same piece of equipment)
2) empNo in Service can only be assigned to one serviceID at one time so the start_date and start_time for two services cannot overlap (employee can't be double booked for same time - only work one cleaning job at once)

e. **Generate the E-R diagram for the logical level (contains FKs as attributes).**