Database Design Project

HealthOne Medical Database

Project Scenario:

You are a small database consulting company specializing in developing databases for the medical industry. You have just been awarded the contract to develop a data model for a database application system for a mid-size health insurance company to keep track of health claims including patient information, provider(doctor) information, information about patient visits to their doctor as well as prescription drugs prescribed to patients.

Information such as patient name, address, phone, email etc. are needed as well as who each patient's primary care doctor is, their insurance ID number and insurance company name. We also want information on each doctor such as their specialty and what hospitals they are affiliated with as well as their phone, address etc. Regarding the hospitals themselves we will need to know where they are located and how to contact them.

The prescriptions given to each patient by a health-care provider also need to be tracked in this particular database at this time to determine claim eligibility including some basic information on the drug being prescribed to make sure there are no conflicts with a patient's other prescriptions. We need to know each drug's name, purpose/use and possible side effects.

Eventually, the database will be used to track trends and for some extrapolative modeling based on the accumulated data. The database will be accessible in English only right now, although plans include making it available in multiple languages eventually.

Step 1 : Determining entities, attributes, UIDs (Sections 2 and 3)

Based on the business scenario stated above you will identify the database needs, and then create a conceptual data model to support these needs.

1. Review the types of information a medical database may contain. Read articles and check the internet to understand the challenges of tracking this information.
2. Research medical information to better understand your topic. Look for the type of information you will need to track.
3. Build a list of business needs, rules and assumptions based on your scenario, research, and objectives.
4. Develop a list of potential entities including their attributes, the attribute's optionality as well as a possible UID for each entity.
5. Create a preliminary Entity Relationship Diagram (ERD) that meets these needs and objectives. (Note : Create entities only, relationships will be added in Step 3)

Step 2 : Supertypes and subtypes (Section 4, Lesson 1)

As stated in the scenario we need to track the visits a patient makes to their doctor. Some patient visits are related to a new issue/illness, some are follow up visits to an existing diagnosis and some visits are routine "well patient" visits or checkups. We would like to be able to track which type of visit each instance is so we can keep specific information regarding the visit. For example :

1. For a new issue/illness visit we will store an initial diagnosis
2. For follow-up visits we need to keep track of the patient's status regarding the diagnosis
3. For routine checkups we need to track patient vital information such as current blood pressure, height and weight.

Modify the ERD using a supertype/subtype structure within the Visit entity.

Step 3 : Relationships (Section 5, Lesson 1,2,3)

While creating your entities you should have been thinking about what relationships the entities would have with each other. Create the relationships between your entities including the relationship's optionality and cardinality.

1. Write out the ERDish for each of the relationships.

2. Some relationships will be transferrable and some non-transferrable – be sure to illustrate this point on the ERD. For example once a prescription is written for a patient it cannot be transferred to another patient.
3. The possible relationship types are : 1-to-1, 1-to-many and many-to-many.
4. Any many-to-many relationships will need to be resolved. For example each doctor may be affiliated with many hospitals and each hospital may have many doctors affiliated with it. We need to make sure this many-to-many relationship is resolved so that we can track which doctors are affiliated with which hospitals.

Modify the ERD to include the relationships.

Step 4 : Normalization (Section 6, Lesson 2,3,4)

Ensure that each entity has been normalized to third normal form – this means:

1. 1st normal form states that all attributes have a single value - no multivalued attributes. For example : each patient can only have one primary doctor, each doctor can only have one specialty etc.
2. 2nd normal form says that all attributes must be dependent on the entire key of the entity. For example we need to know each drug's name, purpose and side effects but if we include this in the Prescription entity it will be dependent only on what drug is prescribed not who it's for or what doctor prescribed it – so it does not belong in the same entity as the prescription information itself.
3. 3rd normal form states that no non-UID attribute can be dependent on another non-UID attribute. For example : A patient's insurance ID number will determine what insurance company they are insured with. The ID number determines the insurance company's name.

Modify the ERD to incorporate all 3 stages of normalization.

Step 5 : Arcs (Section 7, Lesson 1)

Each prescription issued by a doctor must be refillable or non-refillable. It can't be both.

Modify the ERD to make this distinction using an arc. Re-fillable prescriptions will have information about the number and size of refills. All prescriptions will need information about the date, dosage and duration of the RX.

Step 6 : Recursive Relationships (Section 7, Lesson 2)

Some patients in the patient entity may be part of the same family and be covered by the same insurance – we would like to designate a field in the patient entity showing who is the insurance holder for each patient – this field would be the patient ID number of the person holding the insurance for the family.

Modify the ERD to include a recursive relationship on the Patient entity showing the insurance owners role.

Step 7 : Modeling Historical Data (Section 8, Lesson 1)

For use in analyzing providers (doctors) and their effectiveness – if a patient changes primary care doctors we would like to be able to keep track of these changes. This will also aid in patient care tracking throughout their life. We would like to be able to keep a record of each patient's charts and which doctors may have provided information on them.

Modify the ERD to include an entity showing a history of previous primary care doctors and the dates that the doctor was assigned to a particular patient.

Step 8 : Basic Mapping (Section 9 Lesson 1, 2,3,4)

Transform the HealthOne database entities into table diagrams – use suitable naming conventions. Transform relationships into foreign-key columns. Transform the Visit supertype entity using the single-table implementation. Using the following table diagrams, include as many rows as necessary :

Key Type (pk,fk,uk)	Optionality ("*" or "o")	Column Name

Step 9 : Presentation to class (client) (Section 11 Lesson 1,2,3,4)

Create a presentation for the HeathOne client, whose role will be played by your instructor and class. You will be given the opportunity to present the ERD as a communication tool, along with the business rules, to show the client that you understand their needs and that these needs are being met by your design.

. Organize your presentation, by including:

- Statement of the problem
- Information requirements of the business clearly stated
- Assumptions

A suggested order for the presentation is a follows:

1. Introduce the group members
2. State the business issue that you addressed
3. Present and explain the ERD (large enough for all to see)
4. Summarize how your solution will meet the client's needs
5. Present written documentation
6. State assumptions that you made in creating your solution
7. Thank the clients for their time
8. Exit gracefully