MET CS 669 Database Design and Implementation for Business
Health Term Project: Multi-Specialty Ambulatory Health Center

Objective
Create an initial design for the database schema for BUAHC which is a Multi-Specialty Ambulatory Health Center. Map your conceptual design into your approved DBMS, and implement the database schema.

Health Default Project Description
Design a database for BUAHC which is a Multi-Specialty Ambulatory Health Center. Aspects of operations at BUAHC include the following: the facility has multiple buildings and multiple physicians, patients may see more than one physician, and the facility accepts multiple insurance plans. Each patient visit is with one physician. Each physician has a maximum capacity for the number of visits they can handle in a day. To maintain an orderly schedule for physicians, patients, and staff, the BUAHC regulates patient visits to 30 minute blocks; for every hour a physician works, they see at most two patients. Each physician works in one building.

Each physician has a waiting list of patients that have requested to be seen as soon as possible, before their next appointment. When an existing appointment is canceled, the first patient on the waiting list is scheduled for that appointment (at which time that patient is removed from the waiting list).

The database should track insurance plan membership for each patient. The history of physician visits, including the date and physician visited, should also be maintained.

Required Use Cases
Below are a total of 10 use cases. The use cases come in pairs, and each pair covers one aspect of the BUAHC. A complete schema design will allow all ten of these use cases to be addressed in full. For use cases where information is requested, *provide a single query to retrieve the information*. For use cases that require data modification in the database, *develop a stored procedure that performs the modifications, and invoke the stored procedure.* The stored procedures you create should be reusable, by making use of one or more parameters. For example, the stored procedure for use case #1a should use at least two parameters – one to specify the patient and another to specify the particular physician being visited.

1) Appointments
    a. A patient books an appointment with physician “Kathlin Jones” and a different appointment with physician “Scottie Marr”. Develop a parameterized stored procedure that supports a patient booking an appointment, then invoke the stored procedure two times (to satisfy the use case) for a patient of your choosing.
    b. Management requests the names of all physicians who work in the “Agnes” or “Palladuis” buildings who have attended appointments with a particular patient. Write a single query that retrieves this information for a patient of your choosing.
2) Waiting List
    a. A patient phones a physician’s administrative assistant asking for an appointment, and the assistant adds the patient to the waiting list so that the patient will be the next one to be scheduled for a canceled appointment. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a patient and physician of your choosing.

b. A physician has had three appointment cancellations today, so requests the names of the first three patients on their waiting list. Write a single query that retrieves this information for a physician of your choosing.

3) Insurance Plans
   a. A new patient comes into the BUAHC, and a nurse enters the patient's insurance plan enrollment into the system. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a patient and insurance plan of your choosing.
   b. Management requests the names of all current patients who are enrolled in the "Empire Plan" insurance plan. Management defines a "current" patient as one who has had an appointment in the past two years. Write a single query that retrieves this information for a patient of your choosing.
4) Physician Schedules
   a. A physician decides to work an extra hour each day, specifically by beginning work an hour earlier, in order to support two additional appointments per day. Develop a parameterized stored procedure to adjust the physician's schedule, then invoke the stored procedure for a physician of your choosing.
   b. A receptionist needs to know the names of all physicians that are booked for the next two days. Being booked means the physicians have no available appointments for either day. Write a single query that retrieves this information for the receptionist.
5) Your Own Aspect – Select an aspect of the BUAHC you are interested in not already covered in numbers 1-4.
   a. a. Define a use case which involves adding, updating, or deleting data. This use case should involve at least two tables, where the transaction manipulates data in both tables, or the transaction manipulates data in one and reads from the second. Develop a parameterized stored procedure which addresses the use case, then invoke it two times to demonstrate that it addresses the use case and is flexible enough to handle different input.
   b. Define a use case that requires *aggregation* of the data used in part a) to retrieve a meaningful result. Write a single query that retrieves and aggregates the information. The query should involve at least two tables.

Structural Business Rules

You will define business rules to describe the structure of the database. An example of this type of business rule not related to the BUAHC is "A car may be driven by many drivers; each driver drives one or more cars." Each business rule should describe the entities involved, the relationship between the entities, and the optionality and plurality constraints for each entity.

Entity-Relationship Diagrams

You will create a conceptual and logical Entity-Relationship diagram (ERD) for this database. You are not expected to produce a database schema that accommodates all the functions required for the operation of a health center; such a database would have hundreds or thousands of tables. Use the requirements above as a backdrop, then focus your design on the ten use cases. Do not include secondary subschemas such as payment processing, accounting, human resources, and electronic health records; they would unnecessarily bloat the size of your project. To reduce the complexity of the data requirements, you should also presume that BUAHC has not yet implemented a fully functional Electronic Health Record System. A general guideline is that to effectively create this design for this course, the logical ERD should have

between 8 and 20 entities. This range is not exact and will vary according to your specific implementation.

Your logical ERD will be mapped to a relational database schema through the use of SQL. The schema should contain tables, primary and foreign keys, and an index. The primary and foreign keys will help enforce the relationships indicated in the logical ERD, and help enforce referential integrity. Your tables should be normalized to BCNF, or accompanied with a justification as to why the table was not normalized to BCNF. The tables need to be filled with some fictional data. Make sure you integrate sample data from the use cases above. For example, buildings “Agnes” or “Palladuis” should exist in your database because of use case 1b. Many tables may just need a few rows. Barring something extraordinary, each table should need no more than 15 rows to effectively demonstrate the correctness of the queries and stored procedures. You may need to be creative when inserting the data so that the queries return reasonable results.

Index
Create and justify an index that is beneficial to at least one query in your implementation. Include screenshots illustrating the creation of the index, along with an explanation as to why the index is beneficial (be specific).

Deliverables

Your final submission will include two files -- a Word document and a zip file. The contents of each are explained below.

Word Document Contents
Your Word document should contain:

a. your structural business rules.
b. your conceptual ERD or EERD.
c. your logical ERD or EERD.
d. screenshots of the SQL addressing the 10 use cases (make sure to include screenshots of the query or stored procedure creation as well as the results of execution).
e. a screenshot illustrating the creation of the index, along with an explanation as to why the index is beneficial (be specific).

The Word document does *not* need to include screenshots of other SQL, such as your table creation and data inserts, unless you want to clarify something. All of your SQL will be included in the zip file, so your instructor or facilitator can refer to these scripts for additional details.

Keep in mind that part of the term project grade is exposition, so make sure to use sections and headers to keep the document organized, and to include explanations of any aspects of your project that warrant it.

Zip File Contents
Your zip file will contain SQL scripts that enable your instructor or facilitator to re-create your schema and execute your use case queries against the schema. A SQL script is nothing more than a text file that has a ".sql" extension in the filename. The following three scripts should be included, and you may include additional scripts if it makes sense for your implementation. Your zip file should contain:

a. your "create" script which contains the SQL DDL to create the tables, constraints, and stored procedures.
b. your "insert" script which inserts data into the tables.
c. your "use case" script which contains the queries and stored procedure invocations for each of the 10 use cases.

Grading Criteria

We will grade you according to the following percentage breakdown.

Percent of project grade	Item graded
20%	Exposition – How clearly and persuasively the ideas and designs are presented, and how well organized your Term Project submission is.
30%	The completeness and correctness of your design. The business rules, conceptual and logical ERDs, and database schema will be examined, including the entities, relationships, cardinalities, and primary and foreign keys. Your tables should be normalized to BCNF, or accompanied with a valid reason why the table was not normalized to BCNF.
5%	Submitting all five iterative term project deliverables on time. Each deliverable should represent a good faith attempt for an initial iteration of each section.
5%	Correct, working SQL scripts that are consistent with the other deliverables in the project. Using a database user with an empty schema, your facilitator should be able to execute your create script, your insert script, and your query script(s) without errors.
5%	Correct, working primary, foreign, and not null constraints.
5%	The index and explanation.
30%	SQL for the 10 use cases.