ASSIGNMENT COVERSHEET

Prague College

Student Name: Ani Sanikidze	
Class: Relational and NoSQL Databases (FT)	
Assignment: HSP Database	
Lecturer: Mohamed Bettaz	Semester: 2004
Due Date: 25.01.2021	Actual Submission Date: 25.01.2021

Evidence Produced (List separate items)	**Location (Choose one)**	
	X	Uploaded to the Learning Center (Moodle)
		Submitted to reception
Note: Email submissions to the lecturer are not valid.		

Student Declaration	
I declare that the work contained in this assignment was researched and prepared by me, except where acknowledgement of sources is made. **I understand that the college can and will test any work submitted by me for plagiarism.** **Note:** The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement	
Date: 25.01.2021	Student Signature: Ani Sanikidze

A separate feedback sheet will be returned to you after your work has been graded.
Refer to your Student Manual for the Appeals Procedure if you have concerns about the grading decision.

Student Comment (Optional)
Was the task clear? If not, how could it be improved?
Was there sufficient time to complete the task? If not, how much time should be allowed?
Did you need additional assistance with the assignment?
Was the lecturer able to help you?
Were there sufficient resources available?
How could the assignment be improved?
For further comments, please use the reverse of this page.

HSP Database

Prague College

Relational and NoSQL Databases (FT)

Mohamed Bettaz

Ani Sanikidze

Technical Report

25 January 2021

Semester 2004

Word Count:2244

Contents

List of Figures

1 Introduction

The following technical report discusses and summarizes the analysis, design and implementation of the HSP database. Moreover, used analysis methodology, differences between SQL and NoSQL, conversion from SQL to NoSQL and MS SQL to MySQL is discussed below.

2 Database Development Life Cycle

In order the database to be considered as a good one, it should have at least the following desirable properties:

- Availability - ability to access the desired data using queries
- Integrity - accuracy and consistency of data
- Flexibility – adaptability to change
- Security – preserved data against attacks
- Efficiency – efficiency of storing, accessing, filtering and sharing processes
- Usability – data can be accessed and manipulated based on user requirements

Achieving these requirements can be a challenging process. However, the database development life cycle can be used to alleviate the complexity of developing an efficient database from the scratch. Database development life cycle is inherently associated with the more general software development life cycle. They both define the step-by-step process of requirements analysis.

In this project, it was decided to use a simple and straightforward waterfall methodology instead of scrum. The scrum can be considered as a team-based approach for projects. Waterfall methodology is a linear approach, where the output of each phase is the input of the next phase. The implementation of sequential phases is described throughout the technical report.

3 Data Analysis and Design

3.1 Requirements Gathering and Analysis

The requirements gathering and analysis is the very first step of the database development process. Based on the given HSP structure and entities' descriptions, the requirements were determined and applied to the EER diagram. The analysis process leads to the conclusion that the proposed database system's scope can be expanded by adding more attributes, entities and relationships. This eventually led to formulating a more sophisticated HSP database. All this is depicted using conceptual and logical models.

3.2 Models

Database models are used to show the logical structure of a database, including the entities, attributes, relationships and constraints that determine how data can be stored and accessed. Each database model has its unique notation and way of depicting the database structure. For the HSP database, general EERD and Relational Schemas were created. Since data has become a vital corporate resource, good data models can make a significant contribution to an organization's future success (Adelman, et al., 2005).

3.2.1 EER Diagram

The Entity Relationship Diagram (ERD) is a high-level data model used to illustrate the conceptual design of the system. Entities, attributes and relationships are depicted using special notations such as rectangles, diamond shapes and etc., correspondingly. The diagram also differentiates partial and total participations: total participation – each entity involved in the relationship (double line) and partial participation – not all entities are involved in the relationship (single line).

Even though the ERD conveys the general requirements of the database, it is possible to improve its representational capabilities using Enhanced Entity Relationship (EER) notation. EERD is an extension of ERD that introduces concepts such as generalization/specialization, hierarchy/lattice, subclass/superclass and disjoint/overlapping constraints (Elmasri, et al., 2011). In the EERD of the HSP, the superclass/subclass concept was used to represent the inheritance relation-ship between superclass employee and subclasses – doctor, nurse, researcher, and technician. The same approach was used for representing patient, inpatient and outpatient entities.

Note: The detailed description of the logic behind the HSP EER diagram is below the EERD in the EERD.pdf file.

3.2.2 Relational Schema

Based on the EER model it is possible to create a more sophisticated representation of the database – relational schema. The Relational schema illustrates how data is stored in Relational Databases. A relational database stores data in the form of relations (tables) and contains features such as junction tables, attribute domain and foreign keys. In order to validate the relational model, certain requirements should be met. There are 3 main categories of Relational Integrity constraints that should be applied to the relational database.

Domain constraint implies the assignment of the specific data type to attributes, i.e., restricting the kind of values a column or relation can hold in the database table. Domain constraint was applied to all the attributes of the proposed HSP database. For example, the values of the primary keys are defined as integers, first names as var-chars and etc.

Key constraint implies the existence of an attribute or set of attributes that uniquely identify each tuple of the relation. In the HSP schema, all the tables have primary keys, junction tables have composite primary keys.

Referential integrity constraint implies that a foreign key must have a matching primary key or it must be null. This concept maintains the correspondence between the tables.

The relational schema for HSP database was created by using MySQL Workbench. MySQL Workbench provides convenient environment for generating schemas and offers extremely useful feature - Forward Engineering, which transforms generated schema into tables.

Note: The detailed description of EER to Relational Schema conversion process is below the schema in Schema.pdf file.

Normalization Normalization is the process of decomposing and organizing data in database. Normalization ensures database efficiency by avoiding data redundancy, insertion, update and deletion anomalies (Date, 2019).

- Insertion anomaly implies the inability to insert data in the database.
- Update anomaly occurs due to the data inconsistency and partial update
- Delete anomaly implies the loss of unintended data in the process of deleting intended data

In order to overcome these kinds of anomalies, the database is normalized to normal forms. All the relations of the HSP schema have unique attributes - primary keys, with only atomic values. The domain constraint rule is also satisfied. Furthermore, there are no partial and transitive dependencies. All these point out that the model is normalized. The hospital database is normalized to BCNF normal form.

Note: The detailed description of normalization is in the schema.pdf.

4 SQL vs NoSQL

4.1 Structural Difference

SQL databases also called Relational Databases (RDBMS) store and manipulate data based on structure query language (SQL). SQL requires the usage of predefined schemas to determine the structure of the data. Moreover, all the data must have the same structure. SQL databases are table-based.

A NoSQL database uses dynamic schemas for storing unstructured data. The stored data can be document-oriented, column-oriented, graph-based or organized as key-value pairs. In contrast with SQL databases, NoSQL provides more flexibility, in the sense that documents can be created without having a predefined structure. Besides, each document can have its own unique structure.

4.2 Advantages and limitations of SQL and NoSQL DBs

The main advantage of SQL databases is fast query processing. Retrieval of large amounts of data can be done quickly and efficiently. Moreover, SQL is a worldwide uniform platform based on long-established and well-defined standards. Lastly, managing a database using SQL is relatively easy and convenient.

Due to the requirement of having the same data format, the changing of structures can be difficult and disruptive to the whole system.

The NoSQL provides flexibility and no predefined data structures are required. Compared to SQL DB, the addition of a new attribute is a less expensive operation and does not cause risks of maintaining the DB.

NoSQL databases don't support the reliability functions, i.e., it doesn't support ACID. Besides, NoSQL is not compatible with SQL. Therefore, the manual query language is required, making things slower and more complex.

5 Implementation

According to ICA requirements and advantages of SQL, MySQL was used for implementing the database. The first step of the implementation process is the creation of tables. The tables for the HSP database were generated by using the Forward Engineering feature in MySQL Workbench.

Afterward, the data population took place. In order to make the database look closest to the real-world hospital database, the appropriate data was collected and inserted in tables. The values of attributes that don't have significant roles in the database manipulation process were stored with meaningless text – 'lorem ipsum'.The data for patient examination, surgeries, prescriptions and conducted researches is in the time frame of 2020-12-01 to 2020-12-31. Hence, all the stored dates are in a given time frame. This way,it's easier to check the correct functionality of the stored procedures. The referential integrity was achieved by applying constraints on foreign keys.

5.1 Indexes

For enhancing the performance of CRUD operations in HSP database, the indexes were created. Indexes ensure the lookups in a logarithmic time by organizing records in a Binary Search Tree (BST), with smaller values at one side and larger values at the other side. The indexes were created based on the stored procedures. For example, in order to speed up look ups of patients and employees, the composite indexes for sorting by last and first names were created.

5.2 Stored Procedures

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name. Stored procedures are stored in a database; therefore, it is reusable and transparent to any application which wants to use it.

The stored procedure has both advantages and disadvantages. Both of them are discussed below.

Advantages:

- Increases performance of application
- Reusability

- It runs faster than un-compiled SQL commands
- Reduces the traffic between application and database server
- Instead of sending an un-compiled code - SQL commands statements, application only has to send the stored procedure name and get the data back to manipulate it further
- Stored procedures are available in other applications.
- Stored procedure is secured

Disadvantages:

- Takes memory space
- Depending on the database technology debugging stored procedures will either not be possible at all or extremely clunky.
- Writing and maintaining stored procedures is a demanding process

6 SQL to NoSQL

The migration of SQL database into the NoSQL can be done by exporting tables from MySQL Workbench using – Table Data Export Wizard feature and importing it in Mongodb. In this case, the patient, doctor and department tables were manually converted in NoSQL using Mongodb. The tables – doctor, department and patient were transformed to json son formats.

```
doctor
 {"doct_id":1,
  "dept_no":1}
```

Figure 1: Doctor Document

```
department
 {"dept_id":1,
  "dept_name" : "Cardiology"}
```

Figure 2: Department Document

The documents doctor and department are linked to each other by department id foreign key. This represents the relationship - “doctor works in the department”. The junction table “doctor-manages-dept” identifies the department directors by referencing doctor and department ids.

```
Doctor_manages_dept
        {"dept_id" : 1,
         "doct_id" :1)
```

Figure 3: Department Director Document

```
Patient
    {"pat_id":1,
        "f_name":"Oliver",
        "l_name":"Sanchez",
        "addmission_date":"2020-12-25",
        "date_of_birth":"1980-07-26",
        "sex":"m",
        "street":"Oxford 40",
        "city":"London",
        "diagnosis":"HYPERTENSION"
         Examination: [
                Doct_id : '1'
                Date : '2020-12-12'
                Lab_id: '1'
                Conducted test: "lorem ipsum"]}
```

Figure 4: Patient Document

The patient collection contains all the attributes of the patient table. Additionally, examination data is embedded in the patient. By using embedding, the application can retrieve complete patient information by one query. Hence, it improves readability and scalability. The examination array will store all the examinations done on a patient. The potential problem of embedding is that it can cause the creation of large documents. However, based on the real-life scenario, patients undergo one or multiple examinations. Thus, in this case, embedding can be considered as a rational approach.

7 MS SQL to MySQL

The migration of the MS SQL database to MySQL can be done using the Workbench migration wizard. This process can be divided into the following steps:

Step1: In order to start the migration process, ODBC drivers must be present for connecting to the source Microsoft SQL Server database. Besides, both destination MySQL server database, and source MS SQL Server database should be connected with appropriate privileges that are required for migrating the data across. Step 2: In this step, source RDBMS Connection Parameters should

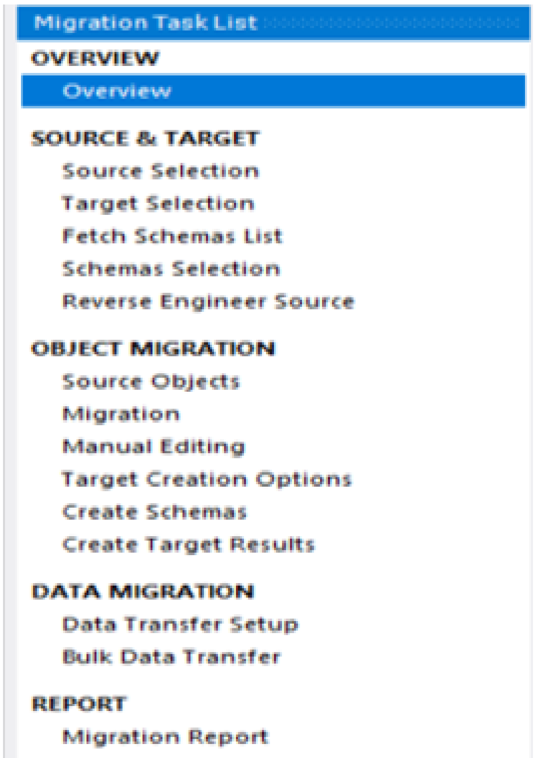

Figure 5: Migration Task List

be defined by selecting DSN and specifying the username. After this, target RDBMS should be configured by specifying hostname or IP address, port and username.

Step 3: After defining source and destinations and selecting the targeted database, the migration process should be successful. The migration Wizard provides the possibility to migrate the schemas, however, the stored procedures require manual conversion.

8 OLAP Cube

Businesses use online transaction processing (OLTP) databases for storing all the transactions and records. These databases usually contain numerous amounts of records that are used by strategists to make decisions for the business. The retrieval of these records can be a costly process. This problem is solved by OLAP which allows efficient retrieval of business intelligence information.

OLAP cube also called hypercube is a data structure that allows fast and flexible multidimensional data analysis for business intelligence (BI). By providing fast data analysis and business intelligence capabilities to end-users, the OLAP cubes overcome the limitations of relational databases. Cubes display and sum large amounts of data while also providing users with access to any data points. This way, the data can be rolled-up, drilled-down, sliced, diced, pivoted (rotate) as needed to handle the widest variety of questions that are relevant to a user's area of interest.

In addition, the data in the OLAP cube is aggregated and detailed. The cube can return a wide variety of information for user queries almost instantaneously. The data within the cube is arranged strictly hierarchically, which means data is organized in a certain way. For example, year is divided into months, months into days and etc.

The important parts of the OLAP cube are dimensions. Dimensions reflect certain aspects of the company. Therefore, instead of manipulating tables, users deal with dimensions. For example, let's assume a user wants to retrieve information about the company's sales and customers data by searching for customer ID, state (location), year, month, day of the month. All these attributes are stored as dimensions. Dimensions also allow the filtering of data. For example, users can filter customers living in a specific geographic location and etc.

9 Conclusion

In conclusion, the information about HSP was analyzed and gathered as requirements. Based on this, EER diagram and relational schemas were created. The schema was normalized to BCNF. Moreover, the implementation details and key concepts such as indexes and stored procedures were explained. The report also introduced conversion from SQL to NoSQl and from MS SQL to MySQL.

References

Adelman, S., M. Abai, and L.T Moss (2005). *Data Strategy*. Boston: Addison Wesley.

Date, C. J. (2019). *Database Design and Relational Theory: Normal Forms and All That Jazz Management*. New York: Apres.

Elmasri, R. and S. Navathe (2011). *Fundamentals of Database Systems*. 6th ed. Boston: Addison Wesley.

List of Figures