**CST4724 Final Project Spring 2020**

**Part 1: Relational Databases – Use ORACLE**

1. Normalize the data in table 1 attached – show the EER diagram for your final results

Books (book\_code, book\_title)

Publisher (publisher\_id, publisher\_name)

Author (author\_id, author\_name)

Table1 (book\_id, book\_code, publisher\_id, author\_id)

For table Books, it us related to table T1 through its primary key book\_code in which it is a foreign key in table T1. This is how these two are related to each other. For the table Publisher it has publisher\_id as its primary key and is related to table T1 by having its primary key as a foreign key in it. Table Author also has a primary key author\_id where it is a foreign key in the table T1. The table T1, has a primary key book\_id, but also has the foreign keys book\_code, publisher\_id, and author\_id from the other three tables Books, Publisher, and Author.

1. For each of the following steps, include screen shots of your code and results
2. Use SQL code to create the tables

A screen shot of a person

Description automatically generated A picture containing holding, sitting, man, green

Description automatically generated

A screen shot of a person

Description automatically generated

A picture containing sitting, clock

Description automatically generated

1. Use SQL code to show that each of the tables has been created

A picture containing sitting, green, black, table

Description automatically generated

1. Use SQL code to enter the information displayed in table 1 into the tables you have created

A picture containing sitting, table, green, phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

A picture containing bottle, screen, holding, sitting

Description automatically generated A close up of a screen

Description automatically generated

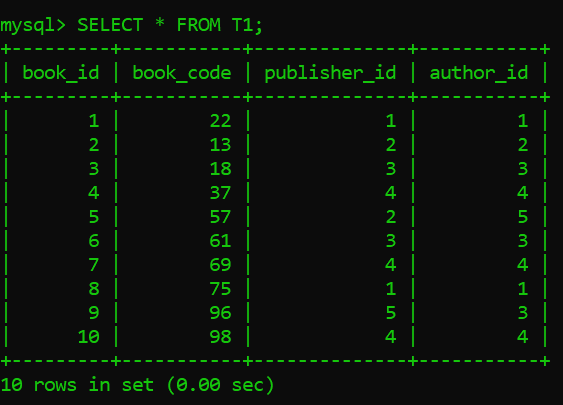
1. Use SQL code to show that the information has been entered into the appropriate tables

A screenshot of a video game

Description automatically generated A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated 

1. Run the following queries:
   1. For each author name, show the book title. Your output should include author name and book title. Order the output by author name in descending order

A close up of text on a screen

Description automatically generated

* 1. For each publisher, show the books they have published. Your output should include publisher name and book title. Order the output by publisher name in ascending order.

A close up of text on a black background

Description automatically generated

* 1. For each author, show the publisher who has published his work. Your output should include the author’s name and the publisher’s name. Order the output by author’s name in ascending order.

A close up of text on a black background

Description automatically generated

* 1. List the title of all the books which Salinger has written.

A screenshot of a cell phone

Description automatically generated

* 1. List the title of all the books published by publisher Vintage

A screenshot of a cell phone

Description automatically generated

* 1. List the title of all the books and the publisher’s name for all books published by Vintage, LB Books or Plume (use IN)

A screenshot of a cell phone

Description automatically generated

* 1. List the title of the books and the publisher name for all books published by Scribner or Plume (use OR) **\*\*\*NOTE: you should only have four lines of output**

**A screenshot of a cell phone

Description automatically generated**

* 1. List the title of the book and the publisher name for all books published by Penguin and written by Morrison (use AND)

A screenshot of a cell phone

Description automatically generated

**Part 2: Document Stores – Use MongoDB**

1. Access to MongoDB:
   1. Use MongoDB in the classroom
   2. Online Terminal:
      1. <https://docs.mongodb.com/manual/tutorial/query-documents/>
2. Create the data in table 1 in MongoDB

A picture containing object, clock, meter

Description automatically generated

1. For each of the following steps, include screen shots of your code and results
2. Use MongoDB to enter the information displayed in table 1 into the collection (s) you have created

A close up of a green screen

Description automatically generated

1. Use MongoDB to show that the information has been entered into the collection(s)

A close up of a green screen

Description automatically generated

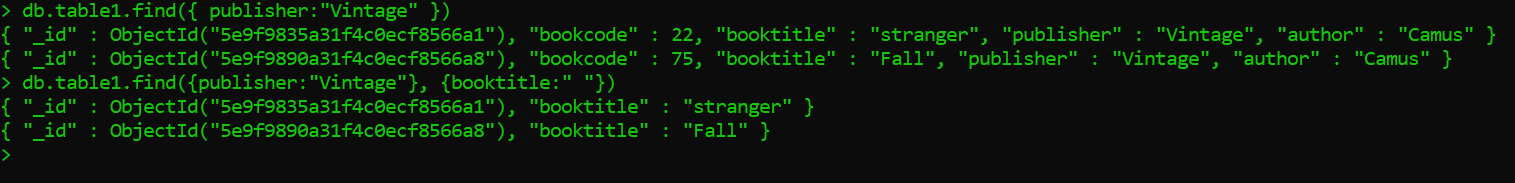
1. Use MongoDB to run the following queries:
   1. Show all books written by author Salinger

Here I used two different ways to get the output!

A screen shot of a clock

Description automatically generated

* 1. Show all books published by Vintage



* 1. Show all books that are published by Vintage, LB Books or Plume (use IN)

A picture containing green, clock

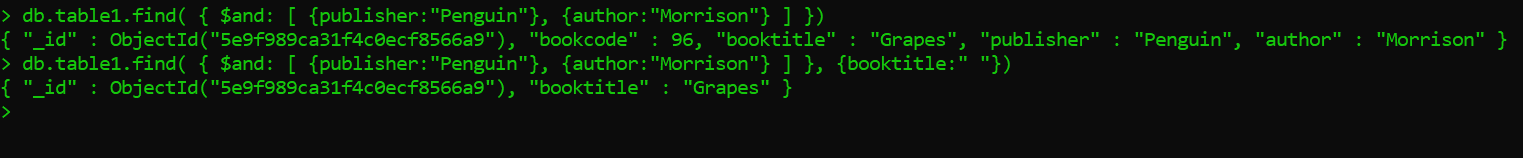
Description automatically generated

* 1. Show all books published by Scribner or Plume (use OR)

A close up of a green screen

Description automatically generated

* 1. Show all books published by Penguin and written by Morrison (use AND)



**Part 3: Graph Store – Use Neo4j**

1. Access Neo4j: <https://neo4j.com/sandbox-v2/>
2. For each of the following steps, include screen shots of your code and results
3. Use the data in Table 1

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

1. Graphically represent the relationship between author and book(s) – use word or any other graphic tool

Camus ----------wrote-------->Stranger

King--------------wrote-------->Dreamcatcher

Morrison-------wrote-------->Beloved

Salinger--------wrote--------->Nine

Heller--------wrote----------->Catch22

Morrison-----wrote---------->Jazz

Salinger--------wrote-------->Franny

Camus---------wrote--------->Fall

Morrison------wrote-------->Grapes

Salinger-------wrote--------->Catcher

1. Use Neo4j code to create the relationships between author and books

This is the code I inserted:

A screenshot of text

Description automatically generated

1. Show all the nodes and relationships

Result:

A screenshot of a cell phone

Description automatically generated

1. Graphically represent the relationship between publisher and book(s) – use word or any other graphic tool

Vintage-------publishes------->Stranger

Scribner-------publishes------->Dreamcatcher

Plume--------publishes--------->Beloved

LB Books-------publishes------>Nine

Scribner-------publishes------->Catch22

Plume-------publishes---------->Jazz

LB Books-------publishes------>Franny

Vintage-------publishes-------->Fall

Penguin-------publishes------->Grapes

LB Books-------publishes------>Catcher

1. Use Neo4j code to create the relationships between publisher and book(s)

Code inserted:

A close up of text on a white background

Description automatically generated

1. Show all the nodes and relationships

Result:

A screenshot of a cell phone

Description automatically generated

1. Graphically represent the relationship between publisher and author – use word or any other graphic tool

Vintage-----contracts---->Camus

Scribner------contracts--->King

Plume------contracts---->Morrison

LB Books----contracts---->Salinger

Scribner-----contracts---->Heller

Penguin----contracts---->Morrison

1. Use Neo4j code to create the relationships between publisher and author

Code inserted:

A picture containing indoor, table, photo, paper

Description automatically generated

1. Show all the nodes and relationships

Result:

A screenshot of a cell phone

Description automatically generated

1. Sort the names in descending order

A screenshot of a cell phone

Description automatically generated

1. Which publisher employs the author morrison

A screenshot of a cell phone

Description automatically generated

**Part 4: Experience recap**

Now that you have had practice implementing the same data in a relational database and two NoSQL stores,

1. What can you say about your experience using Oracle, MongoDB and Neo4j in this exercise?

My experience using Oracle, MongoDB, and Neo4j is that each have their uses for what you may need to do with your data. I can see now after using a relational database and two NoSQL stores how they differ and what is meant by relational, storing documents in MongoDB, and creating relationships using the graph store neo4j. This experience allowed me to have a clear picture/understanding of how these software’s are used in the database world. In this exercise I realized the steps to take for implementing the data in each system. For Oracle I needed to normalize the table and break it up into pieces of tables, so I will not have any duplication of data and having the different tables to come together using primary keys and foreign keys. In MongoDB it was more about taking the data from the table and storing it into the system. Therefore, it’s a document data store because there is no need to normalize the table, just take in the data from what’s in the table and store its information into the system. For the graph store neo4j, I had to identify the relationships in my table, then make the nodes and relationships into neo4j.

1. What can you say about the differences in use among the three systems (Oracle, MongoDB and Neo4j)

The difference in use among the three systems (Oracle, MongoDB, and Neo4j) is that if you want to be organized and have everything in order, you may want to consider using Oracle. This is because in Oracle you have a table organized with the certain subject of information and to find the information related to that you can see it using JOINs. If you have a lot of data and want to store it all, then you would want to use MongoDB. This is because MongoDB doesn’t require you to take the take to normalize your tables, you just take in all the information you have and store it in the system. If you just a way to see the relationships among different things, then it would be best to use neo4j. This is because neo4j is a graph store so you can store the data and create relationships among them and have it shown in a graphical representation among data elements.

1. Identify why you would use each of these systems:
   1. Oracle

You would want to use this when you want your data to be organized. This means organized in a way that there are no duplication and have tables for every category of selected data for. Doing this would make you data organized and good to manage. Doing this would create organized database to be able to select, update, and modify your data. It best for not too much data, just enough to be able to manage.

* 1. MongoDB

You would use MongoDB where you have lots of data that needs to be stored. If there is limited of time that you cannot keep up with spending detailed time in organizing each tables data, then its best to use this document store MongoDB. Doing so would allow you to store in all the information into MongoDB as you can create many documents to store the information. There would no need to have a schema or joins either. You can just look up the information and have the system output he results of what you are looking for.

* 1. Neo4j

Neo4j would be best used for where you need to see relationships among your data. Its good to have this type of system to graphically represent your data from the table and see the different relationships each data has and how they connect to each other. You are not only able to type commands to show you the selected data you wish to see, but also get a graph view to see the connections among everything. Its good to be able to see all this data being shown to you. That’s why if you want to see how all your data is connected and the relationships they have, then using neo4j would be the best choice.

1. What different view of the data do you get from using Oracle, MongoDB and Neo4j

The different view of data you get from using Oracle, MongoDB, and Neo4j is that Oracle shows you the tables that you have normalized into separate, more organized way. If you wan to see a specific table with the data that its supposed to contain then you can view that. If there is a table where you need to see where it connects to the other table to get its information, then you would need to join the table to see the records. The view you get from MongoDB is similar except that there is no joins of tables required. If your looking for a specific data, then you can command it in the system to find the information you want to see and it will show you. It is a document store type, of NoSQL, so it will have all the information in the system and there’s no need to normalize the table, just inserting the information from the tables to the system. In Neo4j, you get a graphical representation view. This allows you to see the relationships and connections of each object from your table as a node and see how everything is connected.

1. Describe how a company can benefit from using all three systems. Use what you have learned in class and by completing this exercise to validate your opinion.

A company can benefit from using all three systems by having all the data in the table and present them or see it for themselves how useful it can be to use each one. While I used these three systems, I could get a better understanding of the data in the table. So, when I used oracle, I had to normalize the table into organized separate tables. I had four table in total and each one had specific data that they contained. This was great to be able to see the different kinds of information on each table. Using oracle, I was able to make join queries that allowed me to see the information that had a connection to other tables information for the data I was looking for. When I used MongoDB, it was much faster to just store the data from the table to the system. From there I can just type a command to look for the data and it will give me the results. This would be useful for tables with many records that need to be inserted into the system. With neo4j, I had to first identify the relationships of the table from this exercise then create the nodes and relationships. I had to be careful not to repeat the same relationship for the same nodes, because just one is enough to see the connection between them unless it’s a different kind of relationship, then I would need to add it. Using neo4j allowed me to see the relationship and connections among my data. This is similar to using joins in oracle but instead of joins, I’m able to graphically represent the data and their relationships with each other. This makes it easier to view than tables in Oracle. If you want to organize your table then oracle is best, if its storing a lot in data from a table then use MongoDB, if you just to better see relationships from your table then use neo4j.Using all three of these, you can view your data differently to better understand your tables data. This is how a company can benefit from using all three systems.

**Table 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Book\_Code** | **Book\_Title** | **Publisher** | **Author** |
| 22 | Stranger | Vintage | Camus |
| 13 | Dreamcatcher | Scribner | King |
| 18 | Beloved | Plume | Morrison |
| 37 | Nine | LB Books | Salinger |
| 57 | Catch 22 | Scribner | Heller |
| 61 | Jazz | Plume | Morrison |
| 69 | Franny | LB Books | Salinger |
| 75 | Fall | Vintage | Camus |
| 96 | Grapes | Penguin | Morrison |
| 98 | Catcher | LB Books | Salinger |