Part 1

1. You have been hired by a business that specializes in manufacturing and selling notebooks to classroom supply stores. They would like you to create a database that helps them keep track of their orders. They only sell a few different models of notebooks and do not plan to add many new types of products over time. Their customer base is also fairly static. They would prefer that their database be based on either **SQL** or **JSON**. Which would you choose for them, and why?

   I believe using a relational database using SQL would be most beneficial for their business. For people who may be less technically inclined, such as business analysis or sales people, it is easier to learn. This is relevant because a database that anyone at a company can easily learn to manage may be useful to the business. The data is primarily structured and it is transaction oriented, so I think SQL is the better choice. SQL also aligns with ACID compliance, which is likely important in this scenario. This is because ACID compliance is important for financial interactions, keeping the database from falling out of sync and making sure the transactions are valid even when errors take place.

2. You have been hired by the alumni division of a University in order to develop a new database of alumni contact information. You know that the database will expand a lot every year, since thousands of students graduate. You also know that you only need a few types of information about each person—their first name, last name, telephone number, email address, street address, city, and zip code. Your boss tells you that you must choose between a **relational database**, a **document database**, or a **graph database**. Which would you choose, and why?

   I think using a relational database may be useful in this situation because the alumni department wants to store a large and expanding number of contacts with only a few pieces of information about each contact. This format will not change, so we already know what the schema will be. We also know that with several students graduating each year and becoming alumni, we need a database that is equipped to handle that growth. If they asked me for a specific software, I would suggest something like Salesforce.

3. You have switched to the public relations division of the same University, and they have asked you to create a database for the new social media platform that they are developing. This platform is designed to allow students to share their experiences at the University with potential new students. They expect the application to evolve rapidly over time, perhaps incorporating new types of users and expanding to new uses. They want the platform to be able to store a lot of different types of information and file

types—images, audio, video, etc. Would you suggest a **relational** or **next generation database** for this work? Why?

I think using a next generation database will be useful in this scenario. Working in the PR department means there will be a lot of social media involved, especially since I'm being asked to create a new social media platform for my first project. Post relational databases are well suited for social media, and allow you to work with interconnected data and more complex relationships. The data in post relational databases can take many different forms, such as the images, audio and video the PR division is looking for.

4. You are an anthropologist that is trying to map all of the family relationships between the members of several indigenous communities in Colombia. You want to create a database that allows you to quickly look up the relationship between any two people living in the communities. Would you select a **hierarchical database**, a **relational database**, a **document database**, or a **graph database**? Why?

   I would use a graph database for this specific scenario. A graph database is very useful when you're interested in exploring relationships and storing data that is interconnected. A document database would be more useful for storing documents, and relational and hierarchical databases also didn't seem like the right fit for this particular project.

5. You are working for a concierge service, and you need to create a database to store information about your clients. You have a fairly stable list of clients, although it slowly expands every year. You know that there is standard information that you store about each client—their name, email address, and phone number. However, the other information that you need to store on each client can vary quite dramatically. Some clients, for instance, use you mostly for restaurant reservations, and you need to store information about their favorite restaurants, their favorite types of food, what types of tables they like in a restaurant, etc. Other clients might need you to book flights for them, and you need to know their frequent flier numbers, their favorite type of seat, etc. And other clients may use you for still other unique types of requests. Would you select a **relational database**, a **document database**, or a **graph database** for this task? Why?

   I would suggest a relational database. As the concierge service said, they have a fairly stable list of clients, but it expands slowly every year. A relational database would work well for this sort of client trajectory. This type of database can also track historical changes. I think this may be useful for a concierge service because they can track changes in dining or food preferences, transportation inquiries, or other common requests and relevant patterns. For example, when helping a client who returns every winter and always asks where the best vegetarian places are, the service would benefit

from a history or record that shows this information under the client's name. They will always be prepared with a list of vegetarian places because they know this client asks about this, and they can be ready with the list before the client even asks. This type of attention to detail and accommodation will translate to exemplary customer service in the client's mind, and the client will be excited to return again.

6. You are working for a technology firm, and you know that all of the programmers there are strong object-oriented programming advocates. You've been asked to create a database for a new project, and you know that you could reasonably choose either a **relational** or **nonrelational** database model. However, being a brand new employee, you want to make your colleagues as happy as possible. Which database model do you think they would prefer? Why?

   My first thought was that developers who enjoy working with object oriented programming may prefer to use a nonrelational database model. Object oriented programming helps developers model complex data, and works well in conjunction with a nonrelational database model. However, I think that object oriented databases are susceptible to the poor adoption among the community phenomenon. It is less common for individuals to understand object databases, and it's possible that this is true at this company as well. Because of this, developers may prefer to use the relational database model, which is more widely known. Keeping this in mind and knowing that both relational and nonrelational database models are operationally feasible, my recommendation is to use a relational database model. I would like to make everyone's life easy in the workplace, especially if I'm trying to make a good impression as a new employee.

7. You are interviewing for a position with a company that focuses on big data analytics. You know that they deal with databases that contain massive amounts of data. During the interview, should you primarily emphasize your experience with **relational** or **nonrelational** database models? What specific technologies might you talk about? Why?

   I would emphasize my skills in nonrelational database models given that the company places such a focus on big data. I would talk about the large scale data storage that nonrelational database models offer and their potential for data processing across multiple servers. I would talk about how beneficial my background in nonrelational database models has been in previous jobs, in departments that focused on social applications, business intelligence, archiving data and advertisement. This will hopefully demonstrate my value as a candidate, and show the company that my experience and background makes me well suited for the role. In terms of specific technologies, I would

mention Amazon’s Dynamo and AWS, Hbase, Google File System, MapReduce, and BigTable, and XML and JSON.

Part 2

Scenario:

1. You are a social scientist that is studying climate change politics, and you want to track information about the scientists that have been attending recent international climate change conferences. Create a normalized database that keeps track of the following information:

- The dates of all the conferences
- The names of all conferences
- The street address, city, and country in which the conference took place
- Which scientists attended each conference
- The names of all scientists that have attended conferences
- The phone numbers and email addresses of each scientist
- The type of science that each scientist is trained to do (e.g. biology, climatology, etc.)
- The name of the institute from which each scientist comes
- The country from which each scientist comes
- The name of the primary scientific institute of every country
- The phone number of these primary scientific institutes
- The primary email address of these primary scientific institutes

You may need to create and store additional information in order to produce your database – for instance, Conference IDs, Country IDs, or Scientist IDs. Create these items as you see fit.

- Each country will have only one primary scientific institute and the scientists belong only to their respective country's primary institute.

Produce a proposed Database schema for this database, and submit this as a file upload for this assignment.

Define Entities

- The dates of all the conferences
- The names of all conferences
- The street address, city, and country in which the conference took place
- Which scientists attended each conference
- The names of all scientists that have attended conferences
- The phone numbers and email addresses of each scientist
- The type of science that each scientist is trained to do (e.g. biology, climatology, etc.)
- The name of the institute from which each scientist comes
- The country from which each scientist comes
- The name of the primary scientific institute of every country
- The phone number of these primary scientific institutes
- The primary email address of these primary scientific institutes

Define Attributes

- The dates of all the conferences
- The names of all conferences
- The street address, city, and country in which the conference took place
- Which scientists attended each conference
- The names of all scientists that have attended conferences
- The phone numbers and email addresses of each scientist
- The type of science that each scientist is trained to do (e.g. biology, climatology, etc.)
- The name of the institute from which each scientist comes
- The country from which each scientist comes
- The name of the primary scientific institute of every country
- The phone number of these primary scientific institutes
- The primary email address of these primary scientific institutes

Organize them into schema

Conference
Name
Date
Conference Address (street address, city and country)
Scientist

Scientist
Name
Phone Number
Email Address
Type of Science
Country
Primary Scientific Institute

Primary Scientific Institute
Name
Phone Number
Email Address

Define Primary Key

Conference

ConferenceID (Pk)
Name
Date
Conference Address (street address, city and country)
Scientist

Scientist

ScientistID (Pk)
Name
Phone Number
Email Address
Type of Science
Country
Primary Scientific Institute

Primary Scientific Institute

Primary Scientific InstituteID (Pk)
Name
Phone Number
Email Address

Define Foreign Key and Define Relationship

Conference

ConferenceID (Pk)
Name
Date
Conference Address (street address, city and country)
Scientist (Fk)

Scientist

ScientistID (Pk)
Name
Phone Number
Email Address
Type of Science
Country
Primary Scientific Institute (Fk)

Primary Scientific Institute

Primary Scientific InstituteID (Pk)
Name
Phone Number
Email Address

First, Second and Third Normal Form

Conference

ConferenceID (Pk)
Name
Date
Street address City Country
Scientist (Fk)

Scientist

ScientistID (Pk)
Name
Phone Number
Email Address
Type of Science
Country
Primary Scientific Institute (Fk)

Primary Scientific Institute

Primary Scientific InstituteID (Pk)
Name
Phone Number
Email Address

Define Data Types

Conference

ConferenceID (Pk)	**Number (3)**
Name	Varchar (30)
Date	Date
Street address	Varchar (30)
City	Varchar (30)
Country	Varchar (30)
Scientist (Fk)	Number (3)

Scientist

ScientistID (Pk)	**Number (3)**
Name	Varchar (30)
Phone Number	Varchar (30)
Email Address	Varchar (30)
Type of Science	Varchar (30)
Country	Varchar (30)
Primary Scientific Institute (Fk)	Number (3)

Primary Scientific Institute

Primary Scientific InstituteID (Pk)	**Number (3)**
Name	Varchar (30)
Phone Number	Varchar (30)
Email Address	Varchar (30)
Country	Varchar (30)

Part 2 Final Answer

Conference

ConferenceID (Pk)	**Number (3)**
Name	Varchar (30)
Date	Date
Street address	Varchar (30)
City	Varchar (30)
Country	Varchar (30)
Scientist (Fk)	Number (3)

Scientist

ScientistID (Pk)	**Number (3)**
Name	Varchar (30)
Phone Number	Varchar (30)
Email Address	Varchar (30)
Type of Science	Varchar (30)
Country	Varchar (30)
Primary Scientific Institute (Fk)	Number (3)

Primary Scientific Institute

Primary Scientific InstituteID (Pk)	**Number (3)**
Name	Varchar (30)
Phone Number	Varchar (30)
Email Address	Varchar (30)
Country	Varchar (30)