**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 July 2020 | **Name:** | Gaganashree P |
| **Course:** | Coursera | **USN:** | 4AL15EC024 |
| **Topic:** | Network Security & Database Vulnerabilities | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | Gaganashree-P |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\User\Pictures\Screenshots\Screenshot (384).png C:\Users\User\Pictures\Screenshots\Screenshot (384).png  **C:\Users\User\Pictures\Screenshots\Screenshot (384).png** What is Data? In simple words, data can be facts related to any object in consideration. For example, your name, age, height, weight, etc. are some data related to you. A picture, image, file, pdf, etc. can also be considered data. What is a Database? A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.  Let us discuss a few examples: An online telephone directory uses a database to store data of people, phone numbers, other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.  Let us also consider Facebook. It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can provide a countless number of examples for the usage of databases. What is SQL? **SQL** is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases. SQL stands for Structured Query language, pronounced as "S-Q-L" or sometimes as "See-Quel"... Relational databases like MySQL Database, Oracle, MS SQL Server, Sybase, etc. use ANSI SQL. What is a Database Management System (DBMS)? **Database Management System (DBMS)** is a collection of programs which enables its users to access database, manipulate data, reporting and representation of data. It also helps to control access to the database. Database Management Systems are not a new concept and as such, had been first implemented in the 1960s.  Charles Bachman's Integrated Data Store (IDS) is said to be the first DBMS in history. With time database technologies evolved a lot while usage and expected functionalities of databases have been increased immensely. Types of DBMS Let us see how the DBMS family got evolved with time. The following diagram shows the evolution of DBMS categories.  [What is Database? What is SQL?](https://www.guru99.com/images/typesofdbms.png)  There are 4 major types of DBMS. Let us investigate them in detail.   * **Hierarchical**: This type of DBMS employs the "parent-child" relationship of storing data. This type of DBMS is rarely used nowadays. Its structure is like a tree with nodes representing records and branches representing fields. The windows registry used in Windows XP is an example of a hierarchical database. Configuration settings are stored as tree structures with nodes. * **Network DBMS:** This type of DBMS supports many-to-many relations. It usually results in complex database structures. RDM Server is an example of a database management system that implements the network model. * **Relational DBMS**: This type of DBMS defines database relationships in the form of tables, also known as relations. Unlike network DBMS, RDBMS does not support many to many relationships. Relational DBMS usually have pre-defined data types that they can support. It is the most popular DBMS type in the market. Examples of relational database management systems include MySQL, Oracle, and Microsoft SQL Server database. * **Object-Oriented Relation DBMS**: This type supports the storage of new data types. The data to be stored is in the form of objects. The objects to be stored in the database have attributes (i.e., gender, ager) and methods that define what to do with the data. PostgreSQL is an example of an object-oriented relational DBMS.  What is NoSQL? **NoSQL** is an upcoming category of Database Management Systems. Its main characteristic is its non-adherence to Relational Database Concepts. NoSQL means "Not only SQL". The concept of NoSQL databases grew with internet giants such as Google, Facebook, Amazon etc. who deal with gigantic volumes of data.  When you use a relational database for massive volumes of data, the system starts getting slow in terms of response time. To overcome this, we could "scale up" our systems by upgrading our existing hardware. The alternative to the above problem would be to distribute our database load on multiple hosts as the load increases. This is known as "scaling out".  NoSQL database is **non-relational databases** that scale-out better than relational databases and are designed with web applications in mind. They do not use SQL to query the data and do not follow strict schemas like relational models. With NoSQL, ACID (Atomicity, Consistency, Isolation, Durability) features are not guaranteed always. Why it makes sense to learn SQL after NoSQL? With the advantages of NoSQL databases outlined which scale-out better than relational models, you might be thinking **why one would still want to learn about the SQL database?** Well, **NoSQL databases** are highly specialized systems and have their special usage and limitations. NoSQL suit more for those who handle huge volumes of data. The vast majority use relational databases and associated tools.  Relational databases have the following advantages over NoSQL databases.   * SQL (relational) databases have a mature data storage and management model. This is crucial for enterprise users. * SQL databases support the notion of views that allow users to only see data that they are authorized to view. The data that they are not authorized to see is kept hidden from them. * SQL databases support stored procedure SQL which allows database developers to implement a part of the business logic into the database. * SQL databases have better security models compared to NoSQL databases.   The world has not deviated from the use of relational databases. There is a **growing** demand for professionals who can handle relational databases. Thus, learning databases and SQL still holds merit. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 July 2020 | **Name:** | Gaganashree P |
| **Course:** | Sales force | **USN:** | 4AL15EC024 |
| **Topic:** | Salesforce Platform Basics | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | Gaganashree-P |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| C:\Users\User\Pictures\Screenshots\Screenshot (387).pngC:\Users\User\Pictures\Screenshots\Screenshot (388).pngC:\Users\User\Pictures\Screenshots\Screenshot (389).pngLearning Objectives After completing this unit, you’ll be able to:   * Define key terms related to the Salesforce architecture. * Find information related to trust. * Explain at least one use case for Salesforce APIs.  What Is the Salesforce Architecture? By now you know that you can use Salesforce to deliver a highly customized experience to your customers, employees, and partners. You can do it without writing much (or any) code, and you can do it fast.  What’s so special about Salesforce? It all starts with our architecture.  Before you close out this window in a frantic attempt to avoid learning about what seems like a really boring subject, sit tight. Learning about Salesforce architecture is quite interesting, and understanding it makes working with the platform a whole lot easier.  When you think about the Salesforce architecture, imagine a series of layers that sit on top of each other. Sometimes it helps to think of it as a cake because cake is delicious, and it makes everything better. Navigate SetupLearning Objectives After completing this unit, you’ll be able to:   * Locate Setup and identify its key elements. * Identify important menus for customizing your org. * Use Quick Find to access menu items.  Setup: Your New Work Home Earlier, we mentioned that you’ll spend a lot of time in Setup during your time as a Salesforce administrator. And we weren’t kidding. Setup is your one-stop-shop for customizing, configuring, and supporting your org.  Since there’s so much you can do in the Setup area, it’s important to get comfortable with navigating it. There are a few ways to approach it. As you learn what’s available to you, you’ll get more comfortable finding the things you need.  You can get to Setup from any page in your Salesforce org. From the gear menu at the top of the screen ( The gear icon to open Setup.), click **Setup**. Let’s get familiar with the Setup area. Power Up with AppExchangeLearning Objectives After completing this unit, you’ll be able to:   * Develop your own AppExchange strategy. * Install an app from AppExchange.  What Is AppExchange? You’re probably comfortable with the idea of app stores. Whether you’re downloading apps on your phone, tablet, computer, or other device, you have to download and install apps to make the most of your technology.  Salesforce is the same way. Earlier, we mentioned the enterprise ecosystem. Salesforce has a community of partners that use the flexibility of the Salesforce platform to build amazing apps and other solutions that anyone can use. These offerings are available (some for free, some at a cost) for installation on AppExchange. | | | |