## Task 1 –

Scenario Choice (Bank Account Management System)

A regular database is the process of storing information in a way that it can be retrieved at any time, a relational database is one of these that presents the data within rows and columns. The term relational comes with the idea that all of the data within those rows are of the same type, the data can be related using things called common keys, this creates the ability to retrieve data based on its relation to other data withing the database.

A NoSQL database are commonly referred as non-relational databases. The systems doesn’t use a common table method as best practice to store and retrieve data but uses a hash method to convert each record’s key into a number and assign that to then a common Key space.

There are 4 Key types of NoSQL, Key-Value Pair DB which is where every record in the database only has 2 fields:

* Key-Value Pair DB which is where every record in the database only has 2 fields. E.G inventory would have product ID as key and product details as value.
* Document Orientated Databases are similar to Key-Value and work in the same way but instead stored the format as a document generally being JSON or XML files.
* Column Databases stores the data into tables, they have set rows which cannot change but have something called dynamic columns where values are stored together in sequence, although this sounds similar to a RDBS system it differs in use. A Column DB is optimised for fast retrieval of columns of data where as RDBS is optimised for storing rows of data.
* Graph Databases store the data in graph format consisting of nodes and edges, where nodes making up the entity and edges make up the relationship between them.

Another form of data management is Data Warehouses and Data Lakes. A Data Warehouse is designed to enable Business intelligence like analytics to be performed on the data and bring value out of old historical data the data for a warehouse needs to be readied and relational for this process. Opposing that, a Data Lake allows you to store vast amounts of structured and unstructured data and then run analytics as well. They are not as in depth as warehouse but can still bring guidance to better business decisions but are not always in format for business professionals to understand or gain value from.

A Data Warehouse might be used over a Data Lake in a scenario such as measuring the effectiveness of a marketing campaign, a warehouse can take all of the scattered data incoming to different parts of the business and create a single source of data for the marketing team to operate from making crucial input for decisions and keeping that time delay from receiving the data as minimal as possible, creating analytics data and making it understandable for anyone within the business. A Data Lake would be better used to gain insights on less direct areas like profit insights from items like, log files, audio/video streams and social media content to identify real time consumer behaviour without having to declare what the data is going to be used for and keeping it in its original format.

## Task 2 –

The scenario I have chosen for this task is a Bank Account Management system. This systems will be an application allowing customers to do basic transactions via the internet. It will allow the customer to; create an account, deposit/withdraw funds and view their account documents like statements and purchase history.

Diagram

Description automatically generatedThe data that will be required for this database will be integers and Strings, I will need to apply a level of data validation. I will require all data to be filled in as NOT NULL and will not accept blank as a valid input. Unless input in the other details columns for each of my entities.

Diagram

Description automatically generated

## Task 3 –

CREATE TABLES:

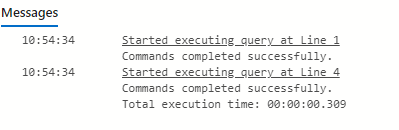
Text

Description automatically generatedThese first set of queries will create all of my different tables and populate them with relevant columns. For example my customer table should populate with customer name, email, password, address and other details. I also will declare their data type, whether that being NVARCHAR (a data type used for differences in sizes of the data, uses Unicode), INT (for whole numbers), DATE (Used for date format dd-mm-yyyy).

We also declare our primary keys and any foreign keys that will be used. For the primary keys I use the IDENITY(1,1) query which makes sure that the data is generated in a 1,2,3,4 pattern, also by declaring it is a primary key that makes it automatically validate that the new data is unique.

When declaring a foreign key I reference the table and column that date will relate to and that declares the relationship between those two tables. In addition, using the NOT NULL or NULL constraint means that where inputted the data can be left empty or not.

Text

Description automatically generated with medium confidenceWhen running these queries I am expecting a response of ‘commands completed successfilly and then I am expecting to be able to open the view of my Banking DB and there should be appropraite tables there, all created in order of the code.

INSERT INTO:

Table

Description automatically generatedText

Description automatically generatedMy next set of queries are to populate my tables with relevant data. I use the INSERT INTO statement to complete this process by declaring the table names that the data will be entering and then in order the column names. Thiis will then be followed by the appropriate data to be inserted into these columns. I can leave the other details columns in the tables empty or filled because I declared thm earlier on as NULL allowing the data to be left empty.

When executing this SQL code, it shoukd return in the message box that 5x7 rows were affected, that meaning 5 rows were added for each table I have (being 7). And the final result should be that I can run the querey (SELECT TOP 100) of each table and it will return all of the tables with the according data I have just inserted.

Table

Description automatically generated

Picture of DB Tables being populated (01/02/2022)

## Task 4 -

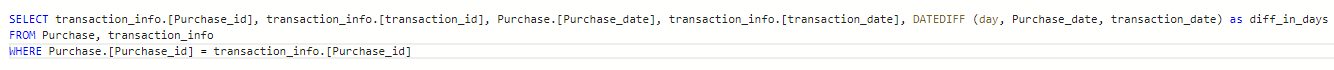
Text

Description automatically generated with low confidenceGraphical user interface

Description automatically generatedMy first scenario for my Banking DB system is that it will be used to find all the customers and when they opened their bank accounts. I will do this by using the (SELECT) query and display the customer table and also the account table so I can grasp which ID of which customer opened their accounts on specific dates and then able to use the relationship between ID and tables to find out the full name of that customer.

As you can see from the screenshots on the right. I Selected the two tables I need to find the appropriate information, This gave the ruselt and I could look for the date that Adam Davis opened his bank account. I saw his customer\_ID is 5 so I went to the Account table and saw that customer ID 5 opened their bank account on 09/02/2015. And can repeat that for all other customers.

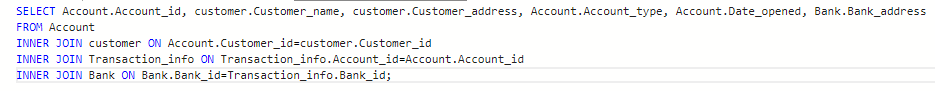
My Second scenario is to be able to find the difference between when customers made a purchase and the date of the actual transaction goung through and being compelted. I will do this by using the DATEDIFF function to find the difference between the data in the table purchases and transaction\_info this will allow me to compare those two dates and find out the delay time between them.

Table

Description automatically generatedAs you can see from the screenshots, to achieve the desired results I would SELECT the relative columns. However I ran into an issue, because Purchase\_id was in both the transaction table and purchase table it became ambiguous so I had to delcare which table I wanted to take it from, whcib led to the query being “transaction\_info.[purchase\_id]”. I then used the DATEDIFF function and returned it as a new colum called diff\_in\_days. Another issue was that it was repeating this DATEDIFF for every possible match of dates so I was getting 25 results (5x5) so to fix that I needed to identify the unique identifiers which were the ID’s so I used a WHERE statement to make sure it only got the result if both Purchase\_ID’s were matching in both tables.

Table

Description automatically generatedFor my third scenario, I will be using an example of a bank wanting to look at the full account details, like customers full name, address, date opened etc and it will retrieve those from 3 different tables (Customers, Account, and Bank). I will do this by using the INNER JOIN query. To join the unique identifiers within these tables and match them up with the relevant data.

As you can see from the screenshots the results I achieved were the table, with all of the required details.

Graphical user interface, text, application

Description automatically generatedFor my fourth scenario, this is showing that the user can collect the data together and report on it with statistics, for example being able to find the average quantity of products people have bought. I did this by using the SELECT AVG function which allows you to take all of the data and average it. (sum it all together and then divide by number of rows) it will then round it down the nearest whole number as I used the ROUND as you cannot buty .5 of a product.

When it comes to access, this database will need to be restricted as it will contain confidential customer information. SO to start of I have incorporated an Azure firewall to help increase security for outside traffic. Also I have made the Database only accessible with the correct username and password. I can also change this to be made using client IP only.so I can restrict access to only specific Ip addresses to make it more secure and/or use Azure active directory so only the people who are added to my Azure subscription with policies can access the database. This will help prevent access to unwanted users. In addition, in Azure data studio there are many security features, like always encrypt. So your data is unreadable to the outside world> finally there is alos the use of eclave Attestation URL meaning that the database is more likely to only be accessed by the inteded users only.