**Question: 1**

**You have an input dictionary given,**

**input\_dict = {"abc":{"def":{"ghi":{"jkl":{"mno":{"pqr":{"stu":{"vwx":{"yz":"you are finally here !!!"}}}}}}}}}**

**Task: You have to write a Python function that will take this input and print it like that,**

**output = {"abc":["def","ghi","jkl","mno","pqr","stu","vwx","yz"], "def":["ghi","jkl","mno","pqr","stu","vwx","yz"], "ghi":["jkl","mno","pqr","stu","vwx","yz"], "jkl":["mno","pqr","stu","vwx","yz"], "mno":["pqr","stu","vwx","yz"], "pqr":["stu","vwx","yz"], "stu":["vwx","yz"], "vwx":["yz"], "yz":["you are finally here !!!"]}**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Question: 2**

**Given an array of length ‘N’, where each element denotes the position of a stall. Now you have ‘N’ stalls and an integer ‘K’ which denotes the number of horses that are mad. To prevent the horses from hurting each other, you need to assign the horses to the stalls, such that the minimum distance between any two of them is as large as possible. Return the largest minimum distance.**

**array: 1,2,4,8,9 & k=3**

**O/P: 3**

**Explanation: 1st horse at stall 1, 2nd horse at stall 4 and 3rd horse at stall 8**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

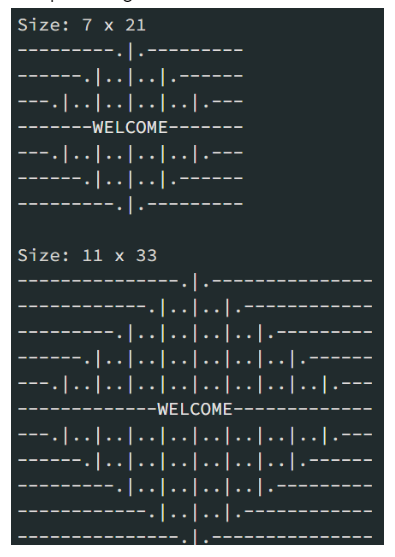
**Question 3 Mr. Karthiken works in a door mat manufacturing company. One day, he designed a new door mat with the following specifications:**

**a) Mat size must be N X M. (N is an odd natural number, and M is 3 times N.)**

**b) The design should have ‘WELCOME’ written in the center.**

**c) The design pattern should only use |, . and – characters.**

**Sample Design is given above image, Write a python code for this.**



Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Question 4**

**Given an array nums of n integers, return an array of all the unique quadruplets [nums[a], nums[b], nums[c], nums[d]] such that:**

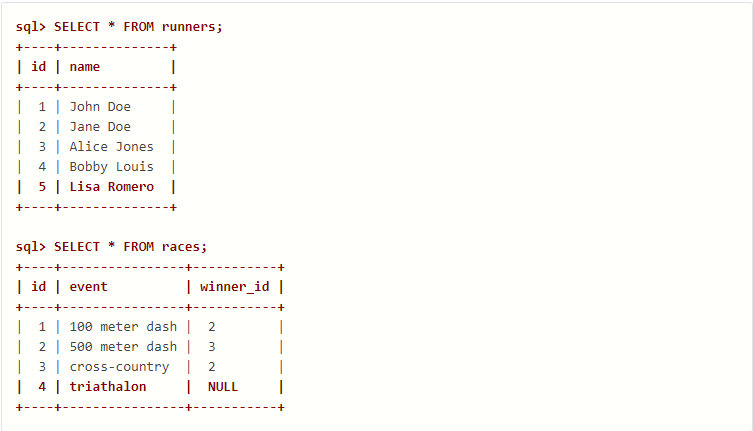
**a) 0 <= a, b, c, d < n b) a, b, c, and d are distinct. c) nums[a] + nums[b] + nums[c] + nums[d] == target**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**SQL Questions**

**Question 1**

**Given the following tables:**

****

**What will be the result of the query below?**

**SELECT \* FROM runners WHERE id NOT IN (SELECT winner\_id FROM races)**

**Explain your answer and also provide an alternative version of this query that will avoid the issue that it exposes.**

Solution: The provided SQL query selects all runners from the runner's table where their id is not present in the winner\_id column of the races table. This means it will return all runners who have never won any races.

However, there is an issue with this query. In SQL, NULL values by default don't match with anything, including other NULL values. This means that if a runner's ID is NULL in the runners table but has a NULL value in the winner\_id of the races table, the runner's ID won't be considered a non-match. As a result, the runner might be excluded from the results even though they have yet to win any races.

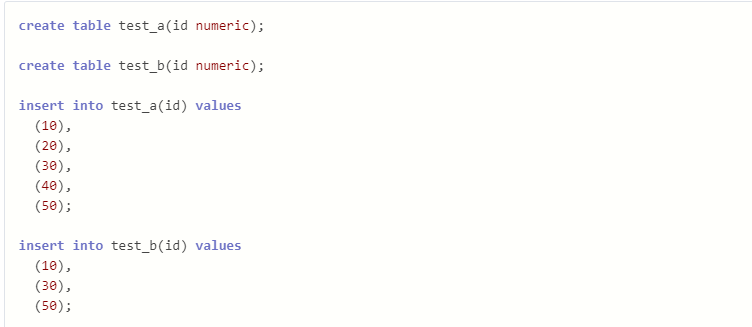
Here is an alternative version of the query that avoids this issue by using the IS NULL operator:

SELECT \* FROM runners WHERE id IS NULL OR id NOT IN (SELECT winner\_id FROM races);

This query explicitly checks for NULL values in the id column of the runner's table. It will return runners whose ID is NULL or not present in the winner\_id column of the races table.

**Question 2**

**Given two tables created as follows**

****

**Write a query to fetch values in table test\_a that are and not in test\_b without using the NOT keyword.**

Solution:

SELECT \*

FROM test\_a

EXCEPT

SELECT \*

FROM test\_b;

This query uses the EXCEPT operator, available in most relational databases. The EXCEPT operator returns all the rows from the first table (test\_a in this case) that are not present in the second table (test\_b).

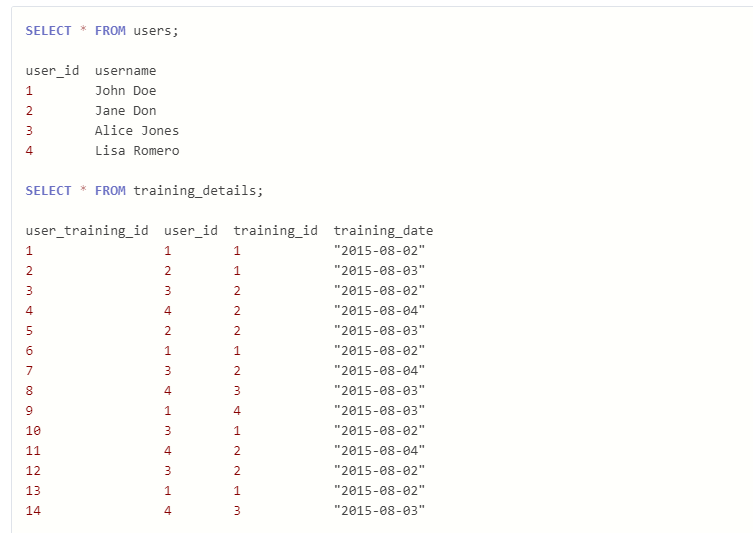
In simpler terms, it finds the difference between the two datasets. Let's break down the query:

SELECT \* FROM test\_a: This selects all columns and rows from the test\_a table. EXCEPT: This is the EXCEPT operator that performs the set difference operation. SELECT \* FROM test\_b: This selects all columns and rows from the test\_b table. The result of this query will be all the rows in test\_a that are not also in test\_b.

Some databases use a different operator for set differences. For example, you can use the MINUS operator in Oracle instead of EXCEPT.

**Question: 3**

**Given the following tables:**

****

**Write a query to to get the list of users who took the a training lesson more than once in the same day, grouped by user and training lesson, each ordered from the most recent lesson date to oldest date.**

Solution:

SELECT u.user\_id, username, training\_id, training\_date, COUNT(\*) AS count

FROM users u

JOIN training\_details t ON t.user\_id = u.user\_id

GROUP BY u.user\_id, username, training\_id, training\_date

HAVING COUNT(\*) > 1

ORDER BY training\_date DESC;

1. SELECT u.user\_id, username, training\_id, training\_date, COUNT(\*) AS count: This clause selects the following columns from the tables involved:

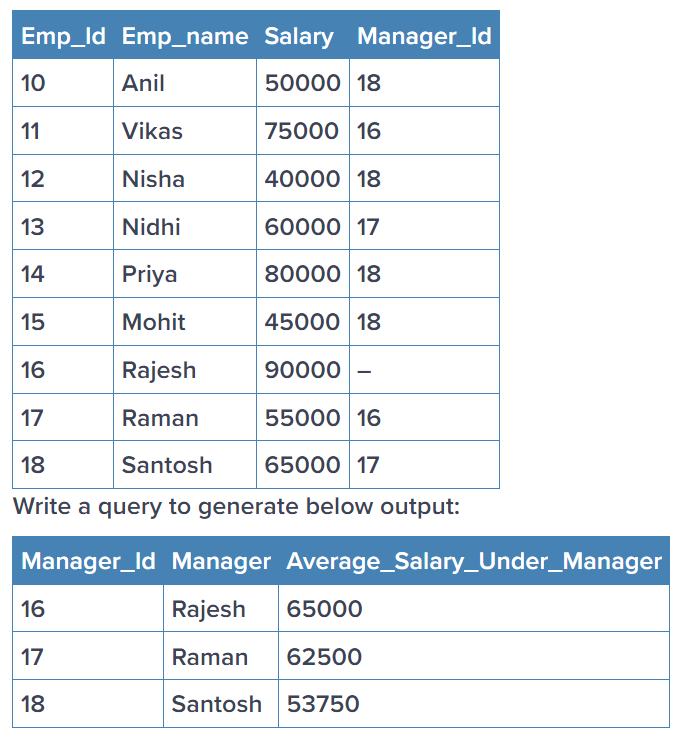
u.user\_id: The user ID from the user's table. username: The username from the user's table. training\_id: The training ID from the training\_details table. training\_date: The training date from the training\_details table. COUNT(\*) AS count: This counts the number of occurrences for each group (explained later). We alias the count as "count" for better readability. FROM users u JOIN training\_details t ON t.user\_id = u.user\_id: This clause joins the user's table (aliased as u) with the training\_details table (aliased as t) on the user\_id column. This ensures that we get data and details on users' training.

1. GROUP BY u.user\_id, username, training\_id, training\_date: This clause groups the rows by the user ID, username, training ID, and training date. This creates groups of entries where a user took the same training lesson on the same date.
2. HAVING COUNT(\*) > 1: This clause filters the grouped results. It only keeps the groups where the count (number of occurrences) is greater than 1. This means it will only return users who took a training lesson more than once on the same day.
3. ORDER BY training\_date DESC: This clause orders the results by the training\_date in descending order. This ensures that each user and training lesson's most recent lesson date is listed first.

This query should return the requested data based on the scenario you described.

**Question: 4**

**Consider the Employee table below.**

****

Solution:

SELECT Manager\_Id, Manager\_name, AVG(Salary) AS Average\_Salary\_Under\_Manager

FROM emp

GROUP BY Manager\_Id, Manager\_name

ORDER BY Manager\_Id;

This query assumes the following about the tables in your database:

There is a table named emp that contains the following columns: Emp\_id (integer) Emp\_name (varchar) Salary (integer) Manager\_Id (integer) The query performs the following steps:

1. SELECT Manager\_Id, Manager\_name, AVG(Salary) AS Average\_Salary\_Under\_Manager: This clause selects the following columns:

Manager\_Id: The manager ID from the emp table. Manager\_name: The manager name from the emp table. AVG(Salary) AS Average\_Salary\_Under\_Manager: This calculates the average salary of employees under each manager and assigns an alias of Average\_Salary\_Under\_Manager for better readability.

1. FROM emp: This clause specifies that the data will be retrieved from the emp table.
2. GROUP BY Manager\_Id, Manager\_name: This clause groups the rows by the Manager\_Id and Manager\_name columns. This ensures that the average salary is calculated for each manager.
3. ORDER BY Manager\_Id: This clause sorts the results by the Manager\_Id in ascending order.

This query should return a table with the manager ID, manager name, and the average salary of employees reporting to that manager.

**Statistics:**

**Question: 1**

**What is the meaning of Six Sigma in statistics? Give proper example**

Solution: Six Sigma is a statistical concept and methodology that aims to improve the quality of process outputs by identifying and reducing the causes of defects and minimizing process variation. It is widely used in various industries, including manufacturing, finance, healthcare, and telecommunications.

"Six Sigma" refers to a statistical measure of process performance. It represents a process capability of producing fewer than 3.4 defects per million opportunities (DPMO). In other words, a Six Sigma process is highly efficient and produces deficient defects or errors.

Organizations typically employ the DMAIC (Define, Measure, Analyze, Improve, Control) methodology to achieve Six Sigma quality. This methodology provides a structured approach to problem-solving and process improvement.

Here's a brief explanation of each phase of the DMAIC methodology:

1. Define: Define the problem and the project goals, as well as the customer requirements.
2. Measure: Measure the current process performance and collect relevant data to identify areas for improvement.
3. Analyze: Analyze the data to identify the root causes of defects and process variation.
4. Improve: Implement solutions to address the identified root causes and improve the process.
5. Control: Establish control measures to sustain the improvements and prevent the recurrence of defects.

Example: Suppose a manufacturing company produces electronic components and wants to improve the quality of a particular process that involves soldering. The company identifies soldering defects, component misalignment, and insufficient soldering as significant product quality issues.

Using the Six Sigma DMAIC methodology, the company can:

1. Define: Define the problem, set project goals (e.g., reduce soldering defects by 50%), and understand customer requirements.
2. Measure: Measure the current process performance, collect data on defects, and identify areas for improvement.
3. Analyze: Analyze the data to identify root causes of defects, such as improper soldering techniques or equipment issues.
4. Improve: Implement solutions to address root causes, such as training employees on proper soldering techniques or upgrading soldering equipment.
5. Control: Establish control measures to monitor the process, such as regular quality checks and ongoing training programs, to ensure sustained improvement and prevent defects from occurring in the future.

By following the DMAIC methodology and achieving Six Sigma quality, the company can significantly improve the quality of its soldering process, reduce defects, and enhance customer satisfaction.

**Question: 2**

**What type of data does not have a log-normal distribution or a Gaussian distribution? Give proper example**

Solution: Solution: Many types of data do not follow a log-normal distribution or a Gaussian (normal) distribution. Here are a few examples:

1. Binary data: Data that can only take two possible values, such as yes/no, true/false, or success/failure, typically do not follow a Gaussian or log-normal distribution. Examples include whether a student passes or fails an exam, whether a customer makes a purchase, or whether a coin lands heads or tails.
2. Count data: Data representing counts of events or occurrences, such as the number of emails received per day, the number of defects in a production batch, or the number of customers in a queue, often do not follow a Gaussian or log-normal distribution. Instead, count data often follow a discrete distribution, such as the Poisson or negative binomial distribution.
3. Categorical data: Data that represent categories or groups, such as colors (red, blue, green), marital status (single, married, divorced), or vehicle types (car, truck, motorcycle), do not follow a Gaussian or log-normal distribution. Categorical data are typically analyzed using contingency tables, chi-square tests, or logistic regression methods.
4. Ordinal data: Data that represent ordered categories, such as Likert scale responses (e.g., strongly agree, agree, neutral, disagree, strongly disagree), education levels (e.g., high school, bachelor's degree, master's degree, PhD), or income brackets (e.g., low, medium, high), do not follow a Gaussian or log-normal distribution. Ordinal data often require non-parametric statistical methods for analysis.
5. Skewed data: Data that are heavily skewed or asymmetric, such as income data, housing prices, or stock returns, may not follow a Gaussian or log-normal distribution. Instead, skewed data may follow distributions such as the gamma, beta, or skewed versions of the normal distribution.

These are just a few data types that do not follow Gaussian or log-normal distributions. In practice, it's essential to assess the distribution of your data using descriptive statistics, visualizations, and statistical tests before choosing appropriate analytical methods.

**Question: 3**

**What is the meaning of the five-number summary in Statistics? Give proper example**

Solution: The five-number summary is a set of descriptive statistics that provides a concise summary of a dataset's distribution. It consists of five values: the minimum, the first quartile (Q1), the median (Q2), the third quartile (Q3), and the maximum.

Here's what each value represents:

1. Minimum: The smallest value in the dataset.
2. Q1 (First Quartile): The value below 25% of the data falls. It is also known as the 25th percentile.
3. Median (Q2): The dataset's middle value when ordered from smallest to largest. It is also known as the 50th percentile.
4. Q3 (Third Quartile): The value below 75% of the data falls. It is also known as the 75th percentile.
5. Maximum: The most significant value in the dataset.

The five-number summary is often used to visualize a dataset's spread and central tendency, particularly when creating box plots.

Here's an example of how to calculate the five-number summary for a dataset:

Consider the following exam score dataset: 65, 70, 75, 80, 85, 90, 95, 100.

1. Minimum: The smallest value is 65.
2. Q1 (First Quartile): There are 8 data points, so the median of the first four (N/2) sorted values is the 2nd value, which is 70.
3. Median (Q2): The median of the entire dataset is the 4th value, which is 80.
4. Q3 (Third Quartile): The median of the last four sorted values is the 6th value, which is 90.
5. Maximum: The most enormous value is 100.

So, the five-number summary for this dataset is Minimum = 65, Q1 = 70, Median = 80, Q3 = 90, Maximum = 100.

**Question: 4**

**What is correlation? Give an example with a dataset & graphical representation on Jupiter Notebook.**

Solution: Correlation is a statistical measure that describes the relationship between two variables. It indicates the extent to which changes in one variable are associated with changes in another variable. Correlation values range from -1 to 1, where:

1 indicates a perfect positive correlation: As one variable increases, the other also increases proportionally. -1 indicates a perfect negative correlation: As one variable increases, the other decreases proportionally. 0 indicates no correlation: There is no apparent relationship between the variables. Here's an example dataset and a graphical representation of correlation using a scatter plot in Google Colab Notebook

<https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Machine learning:**

**Question: 1**

**Imagine you have a dataset where you have different Instagram features like u sername , Caption , Hashtag , Followers , Time\_Since\_posted , and likes , now your task is to predict the number of likes and Time Since posted and the rest of the features are your input features. Now you have to build a model which can predict the number of likes and Time Since posted. Dataset This is the Dataset You can use this dataset for this question.**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Question: 2**

**Train an SVM regressor on : Bengaluru housing dataset**

**Must include in details:**

* **EDA**
* **Feature engineering**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Question: 3**

**Train and fine tune a decision tree using the wine dataset by following the following steps:-**

1. **Use load\_wine() to generate wine dataset**
2. **Split the dataset into train and test dataset**
3. **Use random search CV to hyperparameter tune the Decision Tree**
4. **Try to achieve an accuracy of at least 85%**

**Grow a random forest using the following steps:-**

1. **Continuing the previous question, create 10 subsets of the training dataset. You can use the ShuffleSplit class for it.**
2. **Train 1 decision tree on each subset, using the best hyperparameter values found in the previous question.**
3. **Evaluate all the trees on the test dataset. Are they performing better than the tree created in the previous question?**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Deep Learning :**

**Question: 1**

**(a) Explain how you can implement DL in a real-world application.**

**(b) What is the use of Activation function in Artificial Neural Networks? What would be the problem if we don't use it in ANN networks.**

Solution: (a) Implementing deep learning (DL) in a real-world application involves several key steps:

1. Problem Definition: Identify the problem you want to solve using DL. This could be anything from image classification, natural language processing, time series prediction, etc.
2. Data Collection and Preprocessing: Gather the necessary data for your problem domain. This data may need to be cleaned, preprocessed, and transformed to be suitable for training DL models. Preprocessing steps may include normalization, scaling, handling missing values, etc.
3. Model Selection: Choose an appropriate DL architecture for your problem. This could be a Convolutional Neural Network (CNN) for image-related tasks, a Recurrent Neural Network (RNN) for sequential data, or a combination of different architectures for more complex problems.
4. Training: Train your DL model on the prepared dataset. This involves feeding the data into the model, computing the loss function, and updating the model parameters using optimization algorithms like stochastic gradient descent (SGD) or its variants.
5. Evaluation: Evaluate the trained model on a separate validation dataset to assess its performance. Metrics such as accuracy, precision, recall, F1 score, etc., can be used depending on the nature of the problem.
6. Hyperparameter Tuning: Fine-tune the hyperparameters of the DL model to improve its performance. This may involve adjusting learning rates, batch sizes, network architecture, regularization techniques, etc.
7. Deployment: Deploy the trained DL model into production environment for real-world use. This may involve integrating the model into existing software systems, setting up APIs for inference, and monitoring the model's performance over time.
8. Monitoring and Maintenance: Continuously monitor the deployed model's performance and retrain/update it as necessary. This ensures that the model remains accurate and effective in solving the problem it was designed for.

(b) Activation functions are a critical component of artificial neural networks (ANNs) as they introduce non-linearity into the network, enabling it to learn complex patterns and relationships in the data. Without activation functions, ANNs would reduce to a linear transformation, making them unable to learn and represent non-linear mappings between inputs and outputs.

The primary use of activation functions in ANNs is to introduce non-linearity into the network, allowing it to learn complex patterns and relationships in the data. Activation functions transform the input signal into an output signal that is then passed to the next layer of the network. Commonly used activation functions include:

1. Sigmoid: Maps the input to a value between 0 and 1, making it suitable for binary classification problems.
2. Tanh (Hyperbolic tangent): Similar to the sigmoid function but maps the input to a value between -1 and 1, offering better symmetry around the origin.
3. ReLU (Rectified Linear Unit): Sets all negative values to zero and passes positive values unchanged. ReLU is widely used due to its simplicity and effectiveness in training deep neural networks.
4. Leaky ReLU: Similar to ReLU but allows a small, positive gradient for negative inputs to prevent dying ReLU problem.
5. Softmax: Used in the output layer of multi-class classification problems to produce a probability distribution over multiple classes.

If activation functions are not used in ANN networks, the network would essentially reduce to a linear model, regardless of the number of layers or parameters. This is because the composition of linear functions remains a linear function. Without non-linearity, ANNs would fail to capture complex patterns in the data and would be limited in their ability to learn and generalize from the training data.

**Question: 2**

**Train a Pure ANN with less than 10000 trainable parameters using the MNIST Dataset**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>

**Question: 3**

**Perform Regression Task using ANN**

**Note: You are feel free to use any Regression ML dataset**

Solution: <https://colab.research.google.com/drive/1_eRA2va7ePQcmfjElkYzrzsFX7CqMoli?usp=sharing>