

## Swinfy Solutions

March 24, 2025

### SQL Questions (10)

- SQL 1.** Write a SQL query to find the second highest salary from an employee table named `employees`.
- SQL 2.** Consider two tables: `students` (`student_id`, `name`, `department`) and `courses` (`course_id`, `student_id`, `course_name`, `grade`). Write a SQL query to find the names of students who have received an 'A' in at least three different courses.
- SQL 3.** Write a SQL query to calculate the running total of a column named `sales_amount` in a table called `daily_sales`, ordered by `sale_date`.
- SQL 4.** How would you identify duplicate records in a table named `customer_data` based on email and phone number columns?
- SQL 5.** Given a table `employees` with columns `emp_id`, `emp_name`, `dept_id`, and `salary`, write a query to find the department with the highest average salary.
- SQL 6.** Write a SQL query to find all employees who have a salary greater than their department's average salary. Use tables `employees` (`emp_id`, `name`, `dept_id`, `salary`) and `departments` (`dept_id`, `dept_name`).
- SQL 7.** Write a SQL query to create a pivot table showing the count of orders by customer and month from an `orders` table (`order_id`, `customer_id`, `order_date`, `amount`).
- SQL 8.** Write a SQL query to find the top 3 products with the highest revenue in each category using tables `products` (`product_id`, `product_name`, `category_id`, `price`) and `sales` (`sale_id`, `product_id`, `quantity`, `sale_date`).
- SQL 9.** Given a table `logs` with columns `user_id`, `activity`, and `timestamp`, write a query to find users who performed the same activity three or more consecutive times.
- SQL 10.** Write a SQL query to calculate the median salary for each department from the `employees` table.

### Python Questions (15)

- Python 1.** Write a Python function to find all pairs of elements in an array whose sum equals a specific target value.

## Array Manipulation

```
1 # Implement a Python function to rotate an array by k
  positions to the right.
2 # Example: rotate([1,2,3,4,5], 2) should return [4,5,1,2,3]
3 def rotate_array(nums, k):
4     # Your code here
5     pass
```

Python 2.

**Python 3.** Write a Python function to find the longest substring without repeating characters in a given string.

## String Manipulation

```
1 # Implement a Python function to check if two strings are
  anagrams.
2 def are_anagrams(s1, s2):
3     # Your code here
4     pass
```

Python 4.

**Python 5.** Write a function to find all valid IP addresses that can be obtained from a given string of digits.

## Pandas Challenge

```
1 # Given a pandas DataFrame 'df' with columns 'Date', 'Product
  ', and 'Sales',
2 # write code to find the product with the highest average
  monthly sales.
3 import pandas as pd
4
5 def highest_avg_monthly_sales(df):
6     # Your code here
7     pass
```

Python 6.

**Python 7.** Write a NumPy function to perform batch normalization on a 4D tensor (batch\_size, height, width, channels).

## Array Processing

```
1 # Implement a function to merge two sorted arrays into a
  single sorted array
2 # without using any built-in sorting functions.
3 def merge_sorted_arrays(arr1, arr2):
4     # Your code here
5     pass
```

Python 8.

**Python 9.** Write a Python function that uses NumPy to perform principal component analysis (PCA)

on a given dataset.

#### Pandas Data Cleaning

```
1 # Given a pandas DataFrame with missing values, duplicate rows
  # , and
2 # outliers, write a function to clean the data by:
3 # 1. Removing duplicates
4 # 2. Handling missing values appropriately
5 # 3. Removing outliers (values beyond 3 standard deviations)
6 import pandas as pd
7 import numpy as np
8
9 def clean_dataframe(df):
10     # Your code here
11     pass
```

Python 10.

**Python 11.** Write a function that uses NumPy to implement a sliding window algorithm for time series data.

#### String Pattern Matching

```
1 # Implement a function to find all occurrences of a pattern in
  # a string
2 # using the Knuth-Morris-Pratt (KMP) algorithm.
3 def kmp_search(text, pattern):
4     # Your code here
5     pass
```

Python 12.

**Python 13.** Write a Python function using Pandas to perform a grouped time-based operation that calculates the rolling mean of a metric across different categories.

#### NumPy Matrix Operations

```
1 # Implement a function that uses NumPy to solve a system of
  # linear
2 # equations Ax = b without using numpy.linalg.solve.
3 import numpy as np
4
5 def solve_linear_system(A, b):
6     # Your code here
7     pass
```

Python 14.

**Python 15.** Write a function to implement the Boyer-Moore string search algorithm for pattern matching in a text.

## Machine Learning Questions - Linear and Logistic Regression (10)

- ML 1.** Derive the normal equation for linear regression and explain how it can be used to find the optimal parameters without using gradient descent.

### Gradient Descent Implementation

```
1 # Implement the batch gradient descent algorithm for linear
  regression
2 # from scratch using NumPy. Include appropriate regularization
  .
3 import numpy as np
4
5 def linear_regression_gradient_descent(X, y, learning_rate
    =0.01,
6                                     iterations=1000,
    lambda_reg=0.1):
7     # Your code here
8     pass
```

**ML 2.**

- ML 3.** Explain the difference between L1 and L2 regularization in linear regression. When would you prefer one over the other?

### Logistic Regression Implementation

```
1 # Implement the sigmoid function and the cost function for
2 # logistic regression from scratch.
3 import numpy as np
4
5 def sigmoid(z):
6     # Your code here
7     pass
8
9 def logistic_regression_cost(X, y, theta, lambda_reg=0):
10     # Your code here
11     pass
```

**ML 4.**

- ML 5.** Compare and contrast linear regression and logistic regression in terms of their objective functions, assumptions, and use cases.
- ML 6.** Explain how you would evaluate the performance of a logistic regression model. What metrics would you use and why?

## Multi-class Logistic Regression

```
1 # Implement a one-vs-all (one-vs-rest) logistic regression
  classifier
2 # for multi-class classification using scikit-learn.
3 import numpy as np
4 from sklearn.linear_model import LogisticRegression
5
6 def one_vs_all_logistic_regression(X_train, y_train, X_test,
  num_classes):
7     # Your code here
8     pass
```

ML 7.

ML 8. Describe the assumptions of linear regression and explain how you would check if these assumptions are met in a given dataset.

ML 9. Write a detailed explanation of how to interpret the coefficients in both linear and logistic regression models. Include examples.

## Feature Selection for Regression

```
1 # Implement a function that performs feature selection for a
  regression
2 # model using recursive feature elimination (RFE) with cross-
  validation.
3 from sklearn.feature_selection import RFECV
4 from sklearn.linear_model import LinearRegression
5 import numpy as np
6
7 def select_features_for_regression(X, y, cv=5):
8     # Your code here
9     pass
```

ML 10.

— End of Exam —