1. THEORETICAL BACKGROUND AND METHODOLOGY

1.1 Theoretical Background:

PL/SQL (Procedural Language/Structured Query Language) is a powerful programming language extension for Oracle databases. It combines the procedural programming constructs of languages like C and Pascal with the SQL capabilities for manipulating and retrieving data from the database. In this response, I'll provide an overview of the theoretical background and methodology of PL/SQL.

Theoretical Background:

PL/SQL is designed to provide a means of writing complex database-driven applications that can be executed and stored within the Oracle database. It follows a block structure, where a block is a logical unit that contains one or more PL/SQL statements. The blocks can be nested and can include declarations, assignments, loops, conditional statements, and exception handling constructs.

Blocks: A block is the basic unit of PL/SQL programs. It consists of a declarative section, an executable section, and an optional exception handling section. The declarative section contains variable declarations and data definitions. The executable section contains the actual PL/SQL statements, and the exception handling section deals with errors and exceptions.

Variables: PL/SQL allows you to declare variables of different types, such as integers, strings, dates, and cursors, among others. Variables are used to store and manipulate data within the PL/SQL program.

Control Structures: PL/SQL provides control structures like loops and conditional statements. Loops such as FOR loops, WHILE loops, and LOOP statements allow for repeated execution of a block of code. Conditional statements like IF-THEN-ELSE statements enable the execution of different code blocks based on specified conditions.

Exception Handling: PL/SQL has robust error handling capabilities. It allows you to define exception handlers to catch and handle errors that might occur during the execution of the program. Exception handlers can be used to gracefully handle errors, log information, or take appropriate recovery actions.

1.2 Methodology:

When developing a PL/SQL program, you typically follow a methodology similar to other software development approaches. Here's an outline of the typical steps involved:

Problem Analysis: Understand the problem or requirement that the PL/SQL program needs to address. Identify the data manipulation or retrieval operations required.

Design: Design the structure of your PL/SQL program, including the blocks, variables, control structures, and exception handling mechanisms. Determine how the program will interact with the database and handle errors.

Development: Write the PL/SQL code using a suitable development environment or editor. Create the necessary database objects like tables, views, or stored procedures if needed.

Testing: Execute and test your PL/SQL program to ensure it performs the desired operations correctly. Verify the behavior of the program under different scenarios, including normal and exceptional cases.

Debugging: If issues or errors arise during testing, use debugging techniques like adding debug output or breakpoints to identify and fix the problems. Analyze the error messages or log information to locate the source of errors.

Optimization: Analyze the performance of your PL/SQL program and optimize it if necessary. Identify areas that may benefit from performance tuning, such as using appropriate indexing, minimizing database round-trips, or optimizing queries.

Deployment: Deploy the PL/SQL program in the production environment. Ensure that the necessary privileges and security measures are in place. Monitor the program's execution and handle any issues that arise in the live environment

2. INDUSTRY PROFILE & COMPANY PROFILE

2.1 Industry Profile :

PL/SQL (Procedural Language/Structured Query Language) is a widely used programming language in the industry, primarily in the context of Oracle databases. It is specifically designed for developing database-driven applications and has gained significant popularity among Oracle developers and organizations using Oracle as their database management system. Here is an industry profile of PL/SQL:

Oracle Ecosystem Dominance: PL/SQL is an integral part of the Oracle ecosystem, which includes Oracle Database, Oracle Application Express (APEX), Oracle Forms, Oracle Reports, and other Oracle products. As Oracle is one of the leading providers of enterprise-grade database solutions, PL/SQL is heavily used in industries that rely on Oracle technologies.

Enterprise Applications: PL/SQL is extensively used in the development of enterprise-level applications, especially in sectors such as finance, banking, telecommunications, healthcare, government, and manufacturing. These industries often handle large volumes of data and require robust, scalable, and secure database systems, making PL/SQL an ideal choice for building such applications.

Stored Procedures and Triggers: PL/SQL is commonly used to create stored procedures and triggers within the database. Stored procedures encapsulate complex business logic and data manipulation operations, providing a centralized and controlled approach for accessing and modifying data. Triggers are database objects that execute automatically in response to specific database events, enabling developers to enforce business rules and maintain data integrity.

Performance and Scalability: PL/SQL's tight integration with the Oracle database engine allows for optimized performance and scalability. PL/SQL programs can take advantage of Oracle's features, such as query optimization, parallel processing, and efficient memory management. This makes PL/SQL a preferred choice for applications that require high-performance data processing and handle concurrent user requests.

Legacy Systems and Maintenance: Many organizations have existing legacy systems built on Oracle databases and PL/SQL. As a result, there is a continued demand for PL/SQL developers to maintain, enhance, and migrate these systems. PL/SQL's stability and backward compatibility ensure the longevity of applications built using the language.

Integration with Other Technologies: While PL/SQL is primarily associated with Oracle databases, it can also be integrated with other technologies and frameworks. For example, PL/SQL can be used in conjunction with Oracle APEX to develop web applications, or it can be called from Java programs using JDBC (Java Database Connectivity).

Robust Error Handling and Security: PL/SQL provides robust error handling mechanisms and security features. It allows developers to handle exceptions, log errors, and gracefully recover from failures. Additionally, PL/SQL supports security measures such as role-based access control, encryption, and data masking, ensuring the integrity and confidentiality of sensitive data.

Community and Resources: PL/SQL has a dedicated community of developers, DBAs (Database Administrators), and enthusiasts who actively contribute to forums, blogs, and online resources. These resources provide assistance, best practices, and solutions to common challenges faced by PL/SQL developers.

2.2 Company Profile :

Rumango is a global solution provider in technology consulting and software services domain with an eternal intensifying focus on banking and financial services. Within the short duration since inception Rumango has quickly acquired clients across different geographies serving clients over 14 countries. Rumango with its strong team coming from various domain, IT specialization and consulting services have delivered projects within the stipulated time achieving all project objectives with precision.

Website

http://www.rumango.com

Industry

Information Technology & Services

Company size

51-200 employees

113 on LinkedIn Includes members with current employer listed as Rumango, including part-time roles.

Headquarters

Bangalore, Karnataka

Specialties

Banking, IT Consulting, Devops, Data warehouse, Analytics, Fraud Monitoring, Reconciliation, IT Staffing, Enterprise Service Bus, Core Banking Consulting, Oracle Flexcube, Oracle Digital Banking, and Data Migration

3. INTERNSHIP DETAILS / MY ROLE AS AN INTERN

As a Plsql developer, my role and responsibilities typically involve working with Snowflake's cloud- based data warehousing platform and utilizing Snow SQL to develop, manage, and optimize data solutions.

Data Modeling: Design and implement effective data models within Snowflake, including defining tables, schemas, and views. Ensure data integrity, optimization, and adherence to best practices.

ETL Development: Develop and maintain Extract, Transform, Load (ETL) processes to efficiently load data from various sources into Snowflake. This includes data ingestion, data transformation, and data quality checks.

Query Development: Write SQL queries and scripts using Snow SQL to retrieve, manipulate, and analyze data within Snowflake. Optimize queries for performance and efficiency, considering factors like query execution plans and indexing strategies.

Performance Tuning: Identify and resolve performance bottlenecks in Snowflake, such as optimizing query performance, fine-tuning virtual warehouse configurations, and implementing query caching techniques.

Learning PL/SQL: As an intern, you will begin by familiarizing yourself with the PL/SQL language, its syntax, and its features. You may receive training or guidance from experienced developers or mentors to understand the fundamentals of PL/SQL programming.

Assisting in Development: You will work alongside PL/SQL developers and assist them in various development tasks. This may include writing and modifying PL/SQL code, creating stored procedures, triggers, and functions, and optimizing SQL queries for performance.

Testing and Debugging: You will be involved in testing PL/SQL code to ensure it functions as expected. This may involve writing test cases, executing tests, and analyzing the results. You may also be responsible for debugging code to identify and resolve any issues or errors.

Documentation: Documentation is an important part of software development. You may be required to document the PL/SQL code you work on, including writing comments within the code to explain its purpose and functionality. You may also contribute to the creation of user manuals or technical documentation related to the PL/SQL components.

Collaborating with the Team: As an intern, you will work closely with the PL/SQL development team, collaborating on projects and tasks. You may participate in meetings, discussions, and code reviews, where you can learn from experienced developers and contribute your ideas and insights.

Learning Database Concepts: PL/SQL is tightly integrated with Oracle databases, so it's essential to have a good understanding of database concepts. You will learn about data modeling, relational database design, database querying, and database administration principles.

Data Integration: Integrate Snowflake with other data systems and tools, such as data pipelines, data lakes, BI tools, and analytics platforms. Ensure smooth data flows and seamless integration between Snowflake and other components of the data ecosystem.

Security and Access Control: Implement and enforce security measures within Snowflake, including user access controls, role-based permissions, data encryption, and compliance with regulatory requirements.

Monitoring and Troubleshooting: Monitor and troubleshoot issues related to Snowflake's performance, data loads, query execution, and overall system health. Diagnose and resolve data-related issues, such as data integrity, data quality, or data consistency problems.

Collaboration and Documentation: Collaborate with other team members, stakeholders, and data professionals to understand requirements, provide technical guidance, and contribute to the overall data

solution architecture. Document processes, data flows, and technical specifications to ensure knowledge sharing and maintainable solutions.

Stay Updated: Keep abreast of the latest features, updates, and best practices in Snowflake and the broader data industry. Continuously learn and explore new technologies, tools, and techniques to enhance your skills as a Snowflake developer.

4. SKILLS ACQUIRED AND EXPERIENCE GAINED

During my working period as a plsql developer at Rumango, I learned variety of skills, few of them are listed below.

Snowflake Architecture: Understanding the architecture and components of Snowflake, including virtual warehouses, storage layers, and data sharing concepts.

Snow SQL: Proficiency in writing SQL queries and scripts using Snow SQL to retrieve, manipulate, and analyze data within Snowflake. Knowledge of Snowflake-specific SQL syntax and functions.

Data Modeling: Ability to design and implement effective data models within Snowflake, including defining tables, schemas, views, and managing data integrity.

ETL Development: Experience in developing and maintaining ETL processes to efficiently load data from various sources into Snowflake. Knowledge of data ingestion techniques, data transformation, and data quality checks.

Performance Optimization: Skills in optimizing query performance, including understanding query execution plans, indexing strategies, and utilizing Snowflake-specific features like query caching and materialized views.

Data Integration: Proficiency in integrating Snowflake with other data systems and tools, such as data pipelines, data lakes, BI tools, and analytics platforms. Understanding of data integration patterns and technologies.

Security and Access Control: Knowledge of implementing security measures within Snowflake, including user access controls, role-based permissions, data encryption, and compliance with regulatory requirements.

PL/SQL Programming: You will gain a strong foundation in PL/SQL programming. You will learn the language syntax, control structures, data types, and how to write efficient and effective PL/SQL code to manipulate data and implement business logic.

SQL Querying and Optimization: Working with PL/SQL involves interacting with databases, and you will learn how to write complex SQL queries to retrieve and manipulate data. You will also gain knowledge in query optimization techniques to improve the performance of your SQL queries.

Database Design and Data Modeling: Understanding the principles of database design and data modeling is crucial for efficient and scalable applications. Through your internship, you will learn about entity-relationship (ER) modeling, normalization, and how to design database schemas that meet the requirements of applications.

Oracle Database Concepts: You will acquire a solid understanding of Oracle database concepts, including table structures, views, indexes, constraints, and transactions. You will learn how to work with the Oracle Database Management System and utilize its features and capabilities.

Stored Procedures, Triggers, and Functions: PL/SQL is often used to create stored procedures, triggers, and functions within the database. You will gain hands-on experience in designing and implementing these database objects, which encapsulate business logic and ensure data integrity.

Monitoring and Troubleshooting: Ability to monitor and troubleshoot issues related to Snowflake's performance, data loads, query execution, and system health. Skills in diagnosing and resolving data-related problems.

Collaboration and Documentation: Experience in collaborating with team members, stakeholders, and data professionals to understand requirements, provide technical guidance, and contribute to the overall data solution architecture. Documentation of processes, data flows, and technical specifications.

Data Governance and Compliance: Understanding of data governance principles and compliance requirements in the context of Snowflake, including data privacy regulations, data classification, and data lifecycle management.

Continuous Learning: Demonstrated ability to stay updated with the latest features, updates, and best practices in Snowflake and the broader data industry. Willingness to learn and explore new technologies, tools, and techniques.

Snowflake environment :

- Login to snowflake –

```
>snowsql –a account name –u username
```

- To create warehouse –

```
>CREATE WAREHOUSE WAREHOUSENAME
```

- To create database –

```
>CREATE DATABASE dbaname
```

```
>USE WAREHOUSE warehousename
>USE DATABASE databasename
```

Uploading CSV file in database :

- Create a file format –

```
CREATE FILE FORMAT my_file_format
TYPE = CSV
FIELD_DELIMITER = ','
RECORD_DELIMITER = '\n'

SKIP_HEADER = 1
NULL_IF = ('NULL', 'N/A')
TIMESTAMP_FORMAT = 'YYYY-MM-DD HH24:MI:SS'
DATE_FORMAT = 'YYYY-MM-DD'
```

```
TIME_FORMAT = 'HH24:MI:SS';
```

- Create a csv stage :

```
CREATE OR REPLACE STAGE my_csv_stage
URL = 's3://my-bucket/csv-files/'
FILE_FORMAT = my_file_format;
```

- Uploading files into the stage :

```
PUT file:///path/to/local/file.csv @my_csv_stage auto_compress = True;
```

- Copy the file from stage to desired table :

```
COPY INTO my_table FROM @my_csv_stage/file.csv
FILE_FORMAT = (format_name = format Name) on_error = ‘Skip_file’;
```

Uploading json file in database :

- Create a file format –

```
CREATE FILE FORMAT my_file_format
TYPE = ‘json’
FIELD_DELIMITER = ','
RECORD_DELIMITER = '\n'
SKIP_HEADER = 1
NULL_IF = ('NULL', 'N/A')
TIMESTAMP_FORMAT = 'YYYY-MM-DD HH24:MI:SS'
DATE_FORMAT = 'YYYY-MM-DD'
```

```
TIME_FORMAT = 'HH24:MI:SS';
```

- Create a csv stage :

```
CREATE OR REPLACE STAGE my_json_stage
URL = 's3://my-bucket/json-files/'
FILE_FORMAT = my_file_format;
```

- Uploading files into the stage :

```
PUT file:///path/to/local/file.csv @my_json_stage auto_compress = True;
```

- Copy the file from stage to desired table :

```
COPY INTO my_table FROM @my_json_stage/file.json
FILE_FORMAT = (format_name = format Name) on_error = ‘Skip_file’;
```

Types of tables in snowsql :

Temporary table - temporary tables are a type of table that are created and used for a specific session or user. They are temporary in nature and are automatically dropped at the end of the session or when the user

explicitly drops them. Temporary tables are useful for temporary data storage, intermediate results, or session-specific data manipulation.

Transient table - transient tables are a type of table that is similar to normal tables but with some performance optimizations for specific use cases. Transient tables are ideal for temporary or intermediate data that doesn't need to be persisted for a long time and can be re-created or reloaded if necessary.

Permanent table - permanent tables are regular tables that store data persistently. They are the standard tables used to store structured data for long-term storage and retrieval. Unlike temporary or transient tables, permanent tables are not automatically dropped or deleted by the system and retain data until explicitly removed.

External table - an external table is a table that is created to reference data stored externally, such as in cloud object storage (e.g., Amazon S3, Azure Blob Storage, Google Cloud Storage) or on-premises storage. External tables allow you to query and analyze data without physically loading it into Snowflake, providing a convenient way to work with data that is stored outside the Snowflake environment.

Stream :

A stream is an object that captures and stores changes made to a table. It acts as an ordered, append-only log of data modifications, including inserts, updates, and deletes, made to a specific table. Streams provide a reliable and scalable mechanism for capturing and processing real-time data changes.

- CREATE STREAM stream_name on table_name (this will track all kind of DML commands and changes made to the table)
- CREATE OR REPLACE STREAM stream_name on table_name appends_only = true (It records only the insert changes in the table)
- SHOW STREAM (it displays all the created streams)

Consuming data from a stream –

- CREATE TABLE emp_stream_consume(id number, name varchar, sal number)
- INSERT INTO emp_stream_consume select * from emp_stream
- begin transaction :

  INSERT INTO emp_stream_consume select id, name , sal , from delta_stream where METADATA$ACTION =’INSERT’ and METADATA$ISUPDATE = ‘FALSE’ ;

  Commit;

Task :

A task is an object used to schedule and automate the execution of one or more SQL statements or stored procedures. Tasks enable you to schedule routine data operations, data transformations, or data integration processes within your Snowflake environment.

- Create a task –

```
CREATE OR REPLACE TASK emp_insert_task

warehouse = firstwarehouse schedule = ‘1 minute’ as
insert into emp_stream(id, name, sal) values
(id_seq.nextval, ‘F_name’, 80000)
```

- Start task –

```
ALTER TASK emp_insert_task resume;
```

- Check task status - `SHOW TASKS LIKE 'my_task';`

CONCLUSION

During my internship with Plsql, I had the privilege to delve into the world of modern data management and analytics. Working with Snowflake's powerful cloud data platform and its command-line interface, SnowSQL, has been an enriching experience that has broadened my knowledge and skills in this domain.

Throughout my internship, I engaged in a range of activities and gained valuable insights:

Data Transformation and Analysis: I acquired hands-on experience with SnowSQL, enabling me to load, transform, and analyze data efficiently. I became proficient in executing SQL queries, filtering, aggregating, and manipulating data to derive meaningful insights.

File Format Management: I learned how to define file formats in SnowSQL, understanding the nuances of different formats and their impact on data ingestion and extraction. This knowledge equipped me with the ability to handle various file formats effectively.

Table Management and Optimization: I became adept at creating and managing tables in Snowflake, including normal tables, variant tables, and external tables. I gained insights into optimizing table structures and leveraging Snowflake's capabilities to enhance query performance.

Automation and Scheduling: I had the opportunity to work with Snowflake tasks, automating routine data operations and scheduling their execution. This experience enhanced my understanding of task management and the importance of efficient scheduling.

Real-world Data Challenges: Through practical assignments, I encountered real-world data challenges and learned how to tackle them using Snowflake and SnowSQL. These challenges exposed me to the intricacies of working with large datasets, data integration, and data quality.

Overall, my internship with plsql provided me with a solid foundation in data management and analytics. I gained valuable technical skills, hands-on experience, and exposure to industry-leading technologies in a fast-growing field.

The knowledge and expertise I acquired during this internship will undoubtedly prove instrumental in my future endeavors. I am excited to apply what I have learned to contribute to data-driven solutions and tackle complex business challenges. I am grateful for the opportunity to intern with Snowflake and look forward to leveraging this experience to make a positive impact in the world of data management and analytics.