Data Lake Insight

Service Overview

Issue	01
Date	2024-08-13

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Infographics.. 1

2 What Is Data Lake Insight...3

3 Advantages.. 6

4 Application Scenarios... 9

5 Notes and Constraints..15

6 Permissions Management... 20

7 Quotas...27

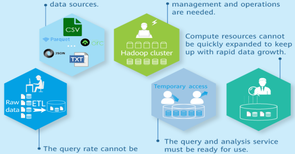
8 Related Services..29

9 Basic Concepts...32

1 Infographics

Huawei Cloud Data Lake Insight

01 Data Lake Challenges



It is difficult to query data from numerous data sources.

Professional analysts with expertise in cluster management and operations are needed.

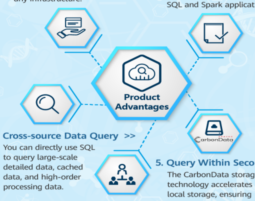
Compute resources cannot be quickly expanded to keep up with rapid data growth.

The query rate cannot be guaranteed for large amounts of data.

The query and analysis service must be ready for use.

02 Data Lake Insight Makes Cloud Data Processing and Analysis More Convenient

DLI is a fully hosted big data processing and analysis service that frees you from ETL. You can use SQL or Spark programs to gain insight into heterogeneous data on Huawei Cloud.



1. Ready-for-Use >>
You don't need to handle any infrastructure.

2. Easy-to-Use >>
DLI supports mainstream data formats and multiple access modes, allowing you to run SQL and Spark applications.

3. Cross-source Data Query >>
You can directly use SQL to query large-scale detailed data, cached data, and high-order processing data.

4. Enterprise Multi-Tenant >>
DLI helps you isolate resources and control data permission.

5. Query Within Seconds >>
The CarbonData storage technology accelerates local storage, ensuring queries are responded to in seconds.

03 Application Scenarios

Historical data analysis across sources

Scenario Description: Directly query a large amount of raw data across a wide time span in various query dimensions.
Typical Applications: transaction auditing, driving behavior analysis, and track playback

Joint analysis of heterogeneous data across sources

Scenario Description: Analyze data association in different storage systems with complex service logic.
Typical Applications: parallel comparison analysis of cold and hot data, and joint analysis of data on OBS, CloudTable, GaussDB(DWS), and RDS

Interactive analysis from multiple dimensions

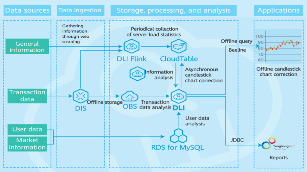
Scenario Description: Query data with low latency in fixed dimensions to support operational decision-making.
Typical Applications: BI analysis, user retention rate analysis, user classification preference analysis, and applause rate analysis

Large-scale log analysis

Scenario Description: Directly query a large amount of raw data across a wide time span in various query dimensions.
Typical Applications: driving behavior analysis, data analysis of ad operations, and financial market analysis

04 Success Story — Transaction Data Analysis

TB of transaction data is generated each day by over 3000 listed companies and approximately 120 million registered shareholders. This data is stored in various formats and devices and managed by different systems. **A financial company uses DLI to analyze and process data in different storage and formats, improving the analysis efficiency. The company reduced development costs by 70%.**



Update data from global platforms in real time

Dump data through DIS

1. Display data in real time and from multiple dimensions.
2. Provide information collected from the network.

Real-time stock transaction and precise data analysis

Customer Benefits :

- Reduced O&M costs
- Reduced development costs for heterogeneous data query and analysis and quicker monetization

Lower O&M costs

Reduced development costs

2 What Is Data Lake Insight

DLI Introduction

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with **Apache Spark**, **Trino**, and **Apache Flink** ecosystems. It frees you from managing any servers.

DLI supports standard SQL and is compatible with Spark SQL and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. You can use standard SQL or Spark and Flink applications to query mainstream data formats without data ETL. DLI supports SQL statements and Spark applications for heterogeneous data sources, including RDS, GaussDB(DWS), CSS, OBS, custom databases on ECSs, and offline databases.

Functions

You can query and analyze heterogeneous data sources such as CloudTable, RDS, and GaussDB(DWS) on the cloud using access methods, such as visualized interface, RESTful API, JDBC, and Beeline. The data format is compatible with five mainstream data formats: CSV, JSON, Parquet, and ORC.

- Basic functions
  - You can use standard SQL statements to query in SQL jobs. For details, see **Spark SQL Syntax Reference**.
  - Flink jobs support Flink SQL online analysis. Aggregation functions such as Window and Join, geographic functions, and CEP functions are supported. SQL is used to express service logic, facilitating service implementation. For details, see **SQL Syntax Constraints and Definitions**.
  - For spark jobs, fully-managed Spark computing can be performed. You can submit computing tasks through interactive sessions or in batch to analyze data in the fully managed Spark queues. For details, see **SQL Syntax Constraints and Definitions**.
- Federated analysis of heterogeneous data sources
  - Spark datasource connection: Data sources such as CloudTable, GaussDB(DWS), RDS, and CSS can be accessed through DLI. For details, see **Enhanced Datasource Connections**.

  - Interconnection with multiple cloud services is supported in Flink jobs to form a rich stream ecosystem. The DLI stream ecosystem consists of cloud service ecosystems and open source ecosystems.
    - Cloud service ecosystem: DLI can interconnect with other services in Flink SQL. You can directly use SQL to read and write data from cloud services, such as DIS, OBS, CloudTable, MRS, RDS, SMN and DCS.
    - Open-source ecosystem: By establishing network connections with other VPCs through enhanced datasource connections, you can access all Flink and Spark-supported data sources and output sources, such as Kafka, Hbase, Elasticsearch, in the tenant-authorized DLI queues.

    For details, see **Flink Jobs**.

- Storage-compute decoupling

  DLI is interconnected with OBS for data analysis. In this architecture where storage and compute are decoupled, resources of these two types are charged separately, helping you reduce costs and improving resource utilization.

  You can choose single-AZ or multi-AZ storage when you create an OBS bucket for storing redundant data on the DLI console. The differences between the two storage policies are as follows:

  - Multi-AZ storage means data is stored in multiple AZs, improving data reliability. If the multi-AZ storage is enabled for a bucket, data is stored in multiple AZs in the same region. If one AZ becomes unavailable, data can still be properly accessed from the other AZs. The multi-AZ storage is ideal for scenarios that demand high reliability. You are advised to use this policy.
  - Single-AZ storage means that data is stored in a single AZ, with lower costs.

- BI tool

  Interconnection with Yonghong BI for data analysis. For details, see **Preparing for Yonghong BI Interconnection**.

DLI Core Engine: Spark+Flink+Trino

- Spark is a unified analysis engine that is ideal for large-scale data processing. It focuses on query, compute, and analysis. DLI optimizes performance and reconstructs services based on open-source Spark. It is compatible with the Apache Spark ecosystem and interfaces, and improves performance by 2.5x when compared with open-source Spark. In this way, DLI enables you to perform query and analysis of EB's of data within hours.
- Flink is a distributed compute engine that is ideal for batch processing, that is, for processing static data sets and historical data sets. You can also use it for stream processing, that is, processing real-time data streams and generating data results in real time. DLI enhances features and security based on the open-source Flink and provides the Stream SQL feature required for data processing.
- Trino, previously known as PrestoSQL, is an open source SQL query engine that allows for interactive query and analysis. It excels in quickly and efficiently processing large-scale data queries and analyses with low latency.

Serverless Architecture

DLI is a serverless big data query and analysis service. It has the following advantages:

- Pay-per-use: You pay only for what you use (scanned data volume/CUH packages). When no jobs are running, you will not be billed.
- Auto scaling: DLI ensures you always have enough capacity on hand to deal with any traffic spikes.

Accessing DLI

A web-based service management platform is provided. You can access DLI using the management console or HTTPS-based APIs, or connect to the DLI server through the JDBC client.

- Using the management console

  You can submit SQL, Spark, or Flink jobs on the DLI management console.

  Log in to the management console. Choose **EI Enterprise Intelligence** > **Data Lake Insight** from the service list.
- Using APIs

  If you need to integrate DLI into a third-party system for secondary development, you can call DLI APIs to use the service.

  For details, see **Data Lake Insight API Reference**.
- JDBC

  You can use JDBC to connect to the server for data query. For details, see **Obtaining the Server Connection Address**.
- Spark-submit

  Jobs can be submitted using Spark-submit. For details, see **Using Spark-submit to Submit a Spark Jar Job**.

3 Advantages

Full SQL Compatibility

You do not need a background in big data to use DLI for data analysis. You only need to know SQL, and you are good to go. The SQL syntax is fully compatible with the standard ANSI SQL 2003.

Decoupled Storage and Compute

DLI compute and storage loads are decoupled. This architecture allows you to flexibly configure storage and compute resources on demand, improving resource utilization and reducing costs.

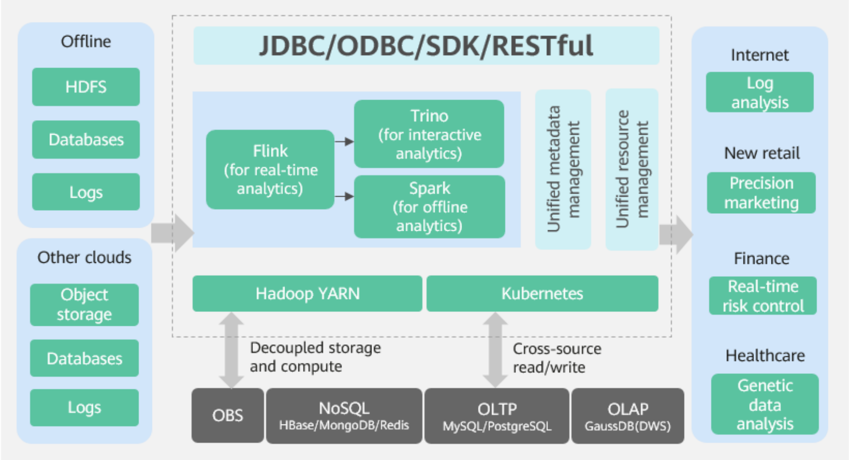
Enterprise Multi-Tenancy

You can manage compute or resource related permissions by project or user, and implement fine-grained control to isolate data for each task.

Serverless DLI

DLI is fully compatible with **Apache Spark** and **Apache Flink** ecosystems and APIs. It is a serverless big data computing and analysis service that integrates real-time, offline, and interactive analysis. Offline applications can be seamlessly migrated to the cloud, reducing the migration workload. DLI provides a highly-scalable framework integrating batch and stream processing, allowing you to handle data analysis requests with ease. With a deeply optimized kernel and architecture, DLI delivers 100-fold performance improvement compared with the MapReduce model. Your analysis is backed by an industry-vetted 99.95% SLA.

Figure 3-1 DLI serverless architecture



DLI has the following advantages over self-built Hadoop clusters:

Table 3-1 Advantages comparison

Advantage	Dimension	Data Lake Insight	Self-built Hadoop
Low cost	Capital cost	Billing is based on the actual amount of data scanned or used CUH. Saving up to 50% costs.	Long-term resource occupation, causing severe resource waste and high costs
	Elastic scalability	Container-based Kubernetes, intelligent elastic scaling	Not supported.
O&M free	O&M cost	Out-of-the-box, serverless architecture	Strong technical capabilities are required for configuration and O&M
	High availability	Cross-AZ DR	N/A
Easy to use	Learning cost	Low. The optimization parameters are standardized based on 10 years' experience in thousands of projects. In addition, DLI provides a GUI for intelligent optimization.	High. Hundreds of tuning parameters need to be learned.

Advantage	Dimension	Data Lake Insight	Self-built Hadoop
	Supported data sources	● Cloud: OBS, RDS, GaussDB(DWS), CSS, MongoDB, and Redis ● On-premises: self-built databases, MongoDB, and Redis	● Cloud: OBS ● On-premises: HDFS
	Ecosystem compatibility	DLV, Yonghong BI, and Fanruan BI	Big data ecosystem tool
	Custom image	Supported. Dependencies can be added as required to meet service diversity requirements.	Not supported.
	Workflow scheduling	Scheduling between Data Lake Factory (DLF) and DataArts Studio	Self-built scheduling tools, such as Airflow
	Multiple enterprise-level tenants	Table-based permission management, providing column level permission granularity.	File-based permission management
High performance	Performance	Higher performance with in-depth software and hardware optimization	Performance is the same as that of Hadoop open-source versions

Cross-Source Analysis

Analyze your data across databases. No migration required. A unified view of your data gives you a comprehensive understanding of your data and helps you innovate faster. There are no restrictions on data formats, cloud data sources, or whether the database is created online or off.

4 Application Scenarios

DLI is applicable to large-scale log analysis, federated analysis of heterogeneous data sources, and big data ETL processing.

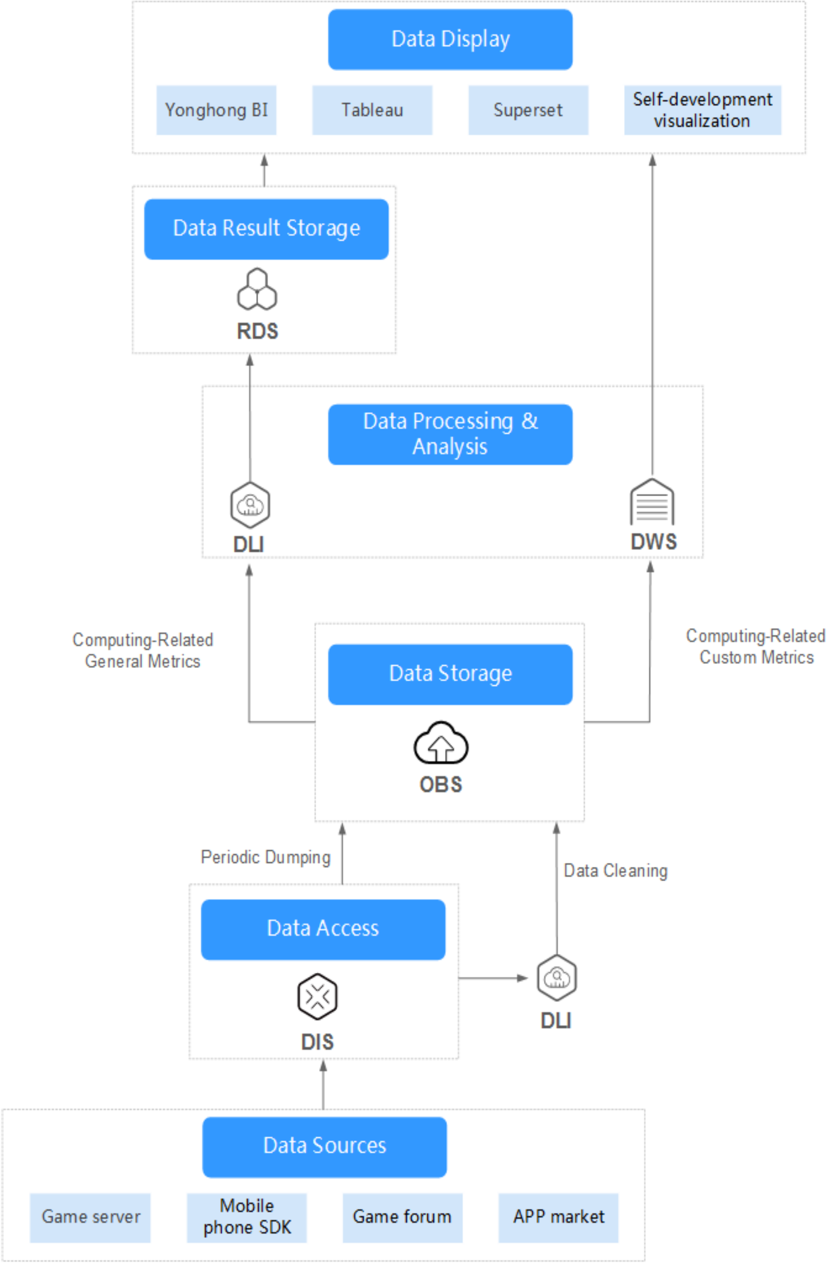
Large-scale Log Analysis

- Gaming operations data analysis

  Different departments of a game company analyze daily new logs via the game data analysis platform to obtain required metrics and make decision based on the obtained metric data. For example, the operation department obtains required metric data, such as new players, active players, retention rate, churn rate, and payment rate, to learn the current game status and determine follow-up actions. The placement department obtains the channel sources of new players and active players to determine the platforms for placement in the next cycle.
- Advantages
  - Efficient Spark programming model: DLI directly ingests data from DIS and performs preprocessing such as data cleaning. You only need to edit the processing logic, without paying attention to the multi-thread model.
  - Ease of use: You can use standard SQL statements to compile metric analysis logic without paying attention to the complex distributed computing platform.
  - Pay-per-use: Log analysis is scheduled periodically based on time-critical requirements. There is a long idle period between every two scheduling operations. DLI adopts the pay-per-use billing mode, which saves the cost by more than 50% compared with the dedicated queue mode. DLI only bills you for the resources used for scheduling.
- It is recommended that you use the following related services:

  OBS, DIS, GaussDB(DWS), and RDS

Figure 4-1 Gaming operations data analysis



Federated Analysis of Heterogeneous Data Sources

- Digital service transformation for car companies

  In the face of new competition pressures and changes in travel services, car companies build the IoV cloud platform and IVI OS to streamline Internet applications and vehicle use scenarios, completing digital service transformation for car companies. This delivers better travel experience for vehicle owners, increases the competitiveness of car companies, and promotes sales growth. For example, DLI can be used to collect and analyze daily vehicle metric data (such as batteries, engines, tire pressure, and airbags), and give maintenance suggestions to vehicle owners in time.

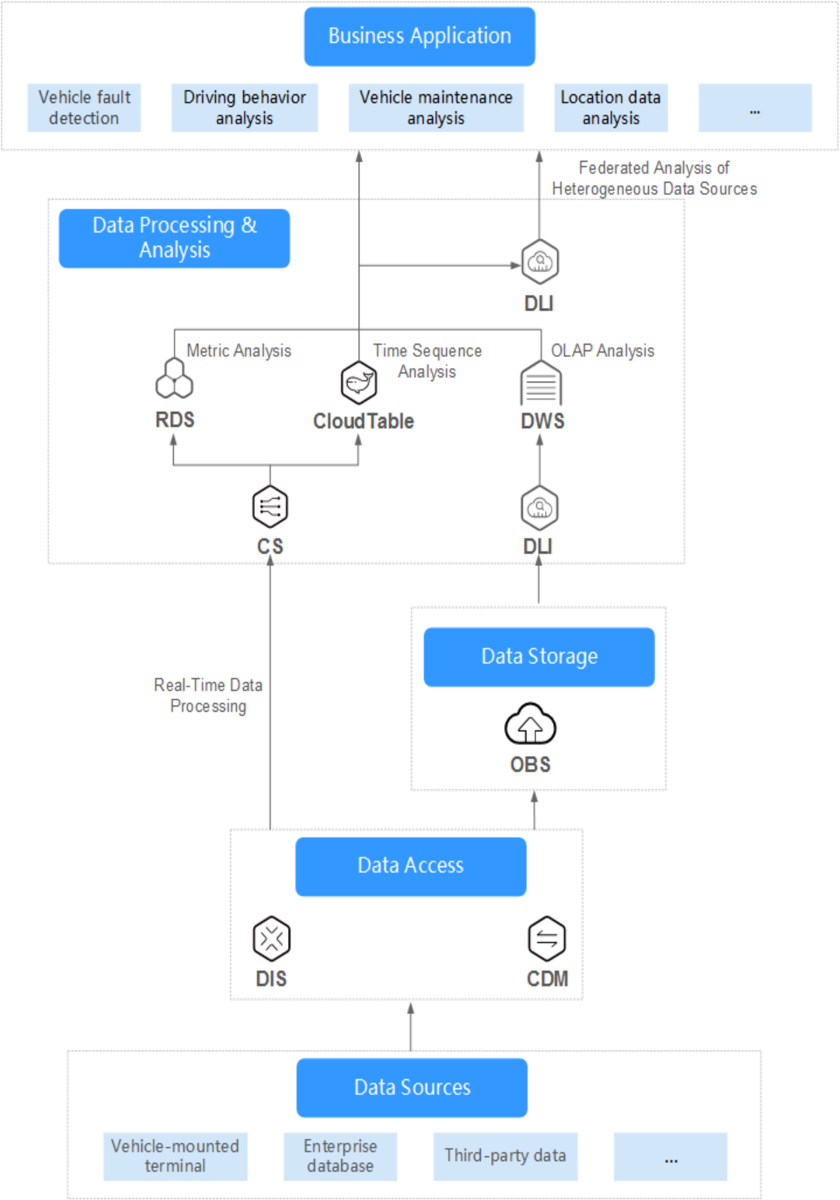
- Advantages

- No need for migration in multi-source data analysis: RDS stores the basic information about vehicles and vehicle owners, table store CloudTable saves real-time vehicle location and health status, and GaussDB(DWS) stores periodic metric statistics. DLI allows federated analysis on data from multiple sources without data migration.
- Tiered data storage: Car companies need to retain all historical data to support auditing and other services that require infrequent data access. Warm and cold data is stored in OBS and frequently accessed data is stored in CloudTable and GaussDB(DWS), reducing the overall storage cost.
- Rapid and agile alarm triggering: There are no special requirements for the CPU, memory, hard disk space, and bandwidth.

- It is recommended that you use the following related services:

  DIS, CDM, OBS, GaussDB(DWS), RDS, and CloudTable

Figure 4-2 Digital service transformation for car companies



Big Data ETL Processing

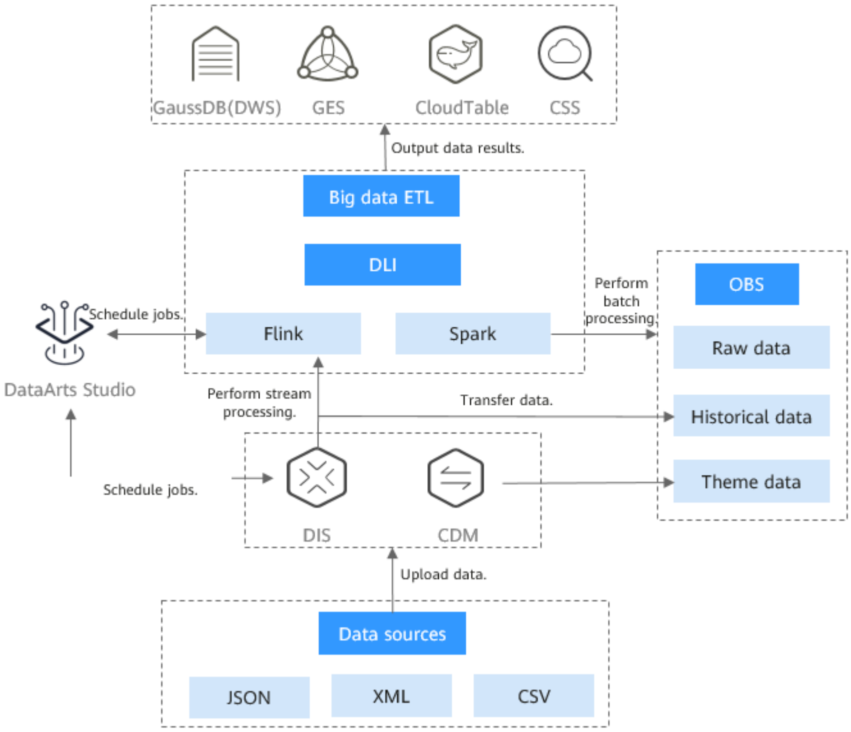
- Carrier big data analysis

  Carriers typically require petabytes, or even exabytes of data storage, for both structured (base station details) and unstructured (messages and communications) data. They need to be able to access the data with extremely low data latency. It is a major challenge to extract value from this data efficiently. DLI provides multi-mode engines such as batch processing and stream processing to break down data silos and perform unified data analysis.

- Advantages
  - Big data ETL: You can enjoy TB to EB-level data governance capabilities to quickly perform ETL processing on massive carrier data. Distributed datasets are provided for batch processing.
  - High Throughput, Low Latency: DLI uses the Dataflow model of Apache Flink, a real-time computing framework. High-performance computing resources are provided to consume data from your created Kafka, DMS Kafka, and MRS Kafka clusters. A single CU processes 1,000 to 20,000 messages per second.
  - Fine-grained permissions management: Your company may have numerous departments, where data needs to be shared and isolated. Using DLI, you can apply for resource queues by tenant to isolate computing resources (CPUs and memory), ensuring job SLA. DLI supports table- or column-level data permission control, allowing for secure access for different departments.
- It is recommended that you use the following related services:

  OBS, DIS, and DataArts Studio

Figure 4-3 Carrier big data analysis



Geographic Big Data Analysis

- Geographic Big Data Analysis

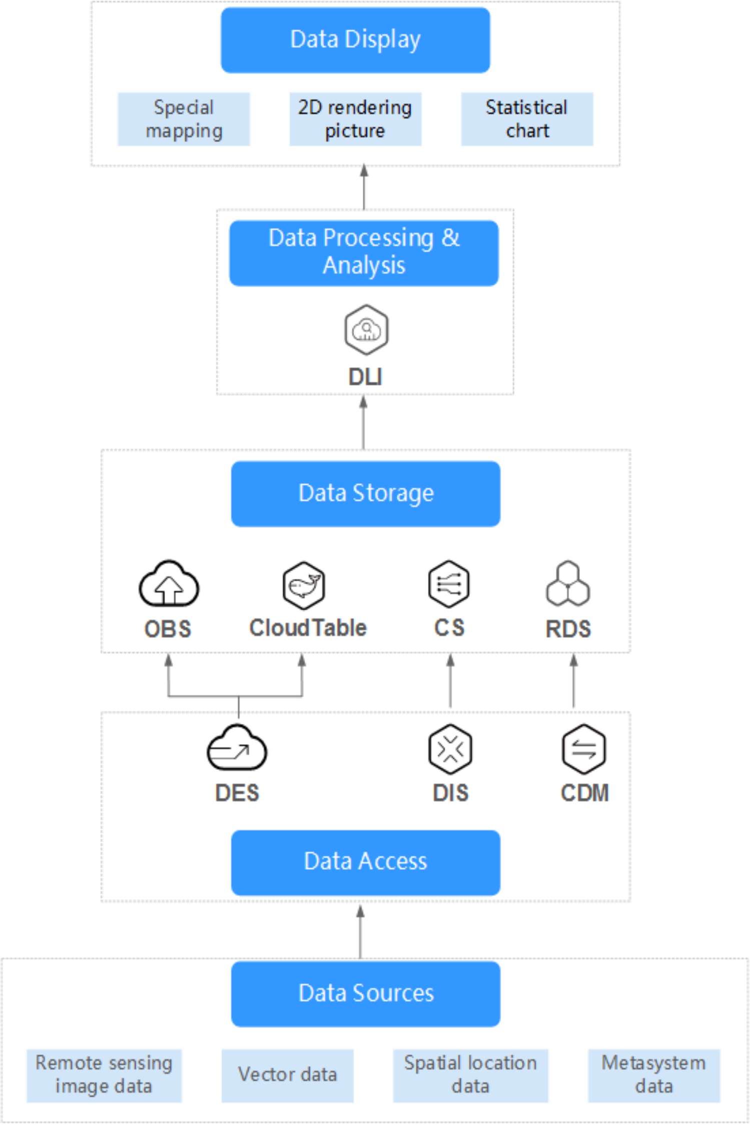
  Geographic big data usually has a large data volume. For example, global satellite remote sensing images might take up to petabytes of data. Besides, there are various types of data, including structured remote sensing image grid data, vector data, unstructured spatial location data, and 3D modeling data. For this scenario, efficient mining tools or methods are essential.

- Advantages
  - Spatial Data Analysis Operators: With full-stack Spark capabilities and rich Spark spatial data analysis Spatial Data Analysis Operators With full-stack Spark capabilities and rich Spark spatial data analysis algorithm operators, DLI delivers comprehensive support for real-time processing of dynamic streaming data with location attributes and offline batch processing. DLI can handle massive data, including structured remote sensing image data, unstructured 3D modeling, and laser point cloud data.
  - CEP SQL: DLI delivers geographical location analysis functions to analyze geospatial data in real time. You can fulfill yaw detection and geo-fencing through SQL statements.
  - Big Data Processing: DLI allows you to quickly migrate remote sensing image data at the TB to EB scale to the cloud and perform image data slicing to offer resilient distributed datasets (RDDs) for distributed batch computing.

- It is recommended that you use the following related services:

  DIS, CDM, DES, OBS, RDS, and CloudTable

Figure 4-4 Geographic Big Data Analysis



5 Notes and Constraints

On Jobs

- DLI supports the following types of jobs: Spark SQL, Spark Jar, Flink SQL, and Flink Jar.
- DLI supports the following Spark versions: Spark 3.3.1, Spark 3.1.1 (EOM), Spark 2.4.5 (EOM), and Spark 2.3 (EOS).
- DLI supports the following Flink versions: Flink Jar 1.15, Flink 1.12 (EOM), Flink 1.10 (EOS), and Flink 1.7 (EOS).
- SQL jobs support the Spark and Trino engines.
  - **Spark**: displays jobs whose execution engine is Spark.
  - **Trino**: displays jobs whose execution engine is Trino.
- SparkUI can only display the latest 100 jobs.
- A maximum of 1,000 job results can be displayed on the console. To view more or all jobs, export the job data to OBS.
- To export job run logs, you must have the permission to access OBS buckets. You need to configure a DLI job bucket on the **Global Configuration** > **Project** page in advance.
- The **View Log** button is not available for synchronization jobs and jobs running on the default queue.
- Only Spark jobs support custom images.
- An elastic resource pool supports a maximum of 32,000 CUs.
- Minimum CUs of a queue that can be created in an elastic resource pool:
  - General purpose queue: 4 CUs
  - SQL queue: Spark SQL queue: 8 CUs; Trino SQL queue: 16 CUs

For more notes and constraints on jobs, see **Job Management**.

On Queues

- A queue named **default** is preset in DLI for you to experience. Resources are allocated on demand. You are billed based on the amount of data scanned in each job (unit: GB).
- Queue types:

  - For SQL: Spark SQL jobs can be submitted to SQL queues.
  - For general purpose: The queue is used to run Spark programs, Flink SQL jobs, and Flink Jar jobs.

  The queue type cannot be changed. If you want to use another queue type, purchase a new queue.

- The billing mode of a queue cannot be changed.
- The region of a queue cannot be changed.
- Queues with 16 CUs do not support scale-out or scale-in.
- Queues with 64 CUs do not support scale-in.
- When creating a queue, you can only select cross-AZ active-active for yearly/monthly queues and pay-per-use dedicated queues. The price of a cross-AZ queue is twice that of a single-AZ queue.
- Newly created queues need to run jobs before they can be scaled in or out.
- DLI queues cannot access the Internet.

  For how to access the Internet from an elastic resource pool, see **Configuring the Connection Between a DLI Queue and a Data Source on the Internet**.

For more notes and constraints on using a DLI queue, see **Notes and Constraints on Using a Queue**.

On DLI Storage Resources

DLI can store databases and tables. DLI storage is billed based on the amount of stored data.

On Resources

- **Database**
  - **default** is the database built in DLI. You cannot create a database named **default**.
  - DLI supports a maximum of 50 databases.
- **Table**
  - DLI supports a maximum of 5,000 tables.
  - DLI supports the following table types:
    - **MANAGED**: Data is stored in a DLI table.
    - **EXTERNAL**: Data is stored in an OBS table.
    - **View**: A view can only be created using SQL statements.
    - Datasource table: The table type is also **EXTERNAL**.
  - You cannot specify a storage path when creating a DLI table.
- **Data import**
  - Only OBS data can be imported to DLI or OBS.
  - You can import data in CSV, Parquet, ORC, JSON, or Avro format from OBS to tables created on DLI.

  - To import data in CSV format to a partitioned table, place the partition column in the last column of the data source.
  - The encoding format of imported data can only be UTF-8.
- **Data export**
  - Data in DLI tables (whose table type is **MANAGED**) can only be exported to OBS buckets, and the export path must contain a folder.
  - The exported file is in JSON format, and the text format can only be UTF-8.
  - Data can be exported across accounts. That is, after account B authorizes account A, account A has the permission to read the metadata and permission information of account B's OBS bucket as well as the read and write permissions on the path. Account A can export data to the OBS path of account B.
- **Package**
  - A package can be deleted, but a package group cannot be deleted.
  - The following types of packages can be uploaded:
    - **JAR**: JAR file
    - **PyFile**: User Python file
    - **File**: User file
    - **ModelFile**: User AI model file

For more notes and constraints on resources, see **Data Management**.

On Enhanced Datasource Connections

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- Enhanced connections can only be created for yearly/monthly and pay-per-use queues.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.

  You can set these permissions by referring to **Service Authorization**.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the notes and constraints on IP addresses are:
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **192.168.*xx*.*xx*** or **192.168.*xx*.*xx*:8181**.

- When checking the connectivity of datasource connections, the notes and constraints on domain names are:
  - The domain name can contain 1 to 255 characters. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

For more notes and constraints on enhanced datasource connections, see **Enhanced Datasource Connection Overview**.

On Datasource Authentication

- Only Spark SQL and Flink OpenSource SQL 1.12 jobs support datasource authentication.
- Flink jobs can use datasource authentication only on queues created after May 1, 2023.
- DLI supports four types of datasource authentication. Select an authentication type specific to each data source.
  - CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled.
  - Kerberos: applies to MRS security clusters with Kerberos authentication enabled.
  - Kafka_SSL: applies to Kafka with SSL enabled.
  - Password: applies to GaussDB(DWS), RDS, DDS, and DCS.

For more notes and constraints on datasource authentication, see **Datasource Authentication Introduction**.

On SQL Syntax

- Constraints on the SQL syntax:
  - You are not allowed to specify a storage path when creating a DLI table using SQL statements.
- Constraints on the size of SQL statements:
  - Each SQL statement should contain less than 500,000 characters.
  - The size of each SQL statement must be less than 1 MB.

Other

- For quota notes and constraints, see **Quotas**.
- Recommended browsers for logging in to DLI:
  - Google Chrome 43.0 or later
  - Mozilla Firefox 38.0 or later
  - Internet Explorer 9.0 or later

For details about the compatibility list of more browsers, see **Which Browsers Are Supported?**

6 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your DLI resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your public cloud resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

If your account does not require individual IAM users for permissions management, you may skip over this section.

IAM can be used free of charge. You pay only for the resources in your account. For more information about IAM, see **IAM Service Overview**.

DLI Permissions

By default, new IAM users do not have permissions assigned. You need to add the users to one or more groups, and attach permissions policies or roles to these groups. The users then inherit permissions from the groups to which they are added. After authorization, the users can perform specified operations on DLI based on the permissions.

DLI is a project-level service deployed and accessed in specific physical regions. To assign ServiceStage permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing DLI, the users need to switch to a region where they have been authorized to use cloud services.

Type: There are roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. If one role has a dependency role required for accessing SA, assign both roles to the users. However, roles are not an ideal choice for fine-grained authorization and secure access control.

- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs. For the actions supported by DLI APIs, **Permissions Policies and Supported Actions**.

Table 6-1 DLI system permissions

Role/Policy Name	Description	Category	Dependency
DLI FullAccess	Full permissions for DLI.	System-defined policy	This role depends on other roles in the same project. ● Creating a datasource connection: **VPC ReadOnlyAccess** ● Creating yearly/ monthly resources: **BSS Administrator** ● Creating a tag: **TMS FullAccess** and **EPS EPS FullAccess** ● Using OBS for storage: **OBS OperateAccess** ● Creating an agency: **Security Administrator**
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy	None

Role/Policy Name	Description	Category	Dependency
Tenant Administrator	Tenant administrator ● Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. ● Scope: project-level service	System-defined role	None
DLI Service Administrator	DLI administrator. ● Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. ● Scope: project-level service	System-defined role	None

Table 6-2 lists the common operations supported by each system policy. You can choose required system policies according to this table.

For details about how to grant permissions using SQL syntax, see **Data Permissions List** in *Data Lake Insight SQL Syntax Reference*.

Table 6-2 Common operations supported by each system permission

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
Queue	DROP_QUEUE	Deleting a Queue	√	×	√	√
	SUBMIT_JOB	Submitting a job	√	×	√	√
	CANCEL_JOB	Terminating a Job	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	RESTART	Restarting a queue	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to a queue	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to a queue	√	×	√	√
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√
	CREATE_VIEW	Creating a view	√	×	√	√
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding a role	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying tables in a database	√	√	√	√
	DISPLAY_DATABASE	Displaying databases	√	√	√	√
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting tables	√	×	√	√
	SELECT	Querying tables	√	×	√	√
	INSERT_INTO_TABLE	Inserting table data	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Overwriting a table	√	×	√	√
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Administrator
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√
Enhanced datasource connection	BIND_QUEUE	Binding an enhanced datasource connection to a queue It is only used to grant permissions across projects.	×	×	×	×

You can create custom policies to supplement system-defined policies and implement more refined access control. For details about how to create a custom policy, see **Creating a Custom Policy**.

7 Quotas

What Is a Quota?

A quota limits the quantity of a resource available to users, thereby preventing spikes in the usage of the resource.

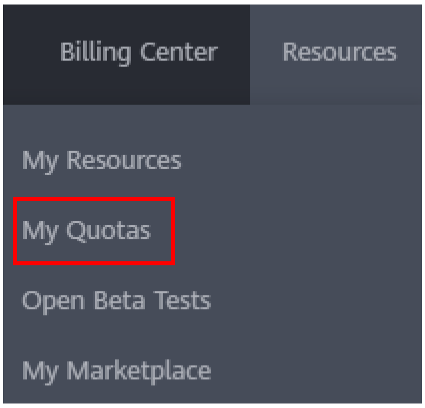
You can also request for an increased quota if your existing quota cannot meet your service requirements.

How Do I View My Quotas?

1. Log in to the management console.
2. Click 📍 in the upper left corner and select a region and a project.
3. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.

   Figure 7-1 My quotas

4. View the used and total quota of each type of resources on the displayed page.

   If a quota cannot meet service requirements, increase a quota.

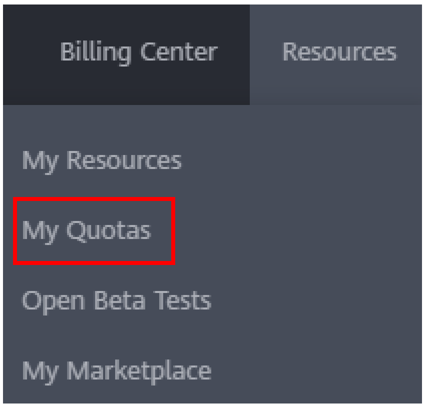
How Do I Apply for a Higher Quota?

1. Log in to the management console.

2. In the upper right corner of the page, choose **Resources** > **My Quotas**.

   The **Service Quota** page is displayed.

   Figure 7-2 My quotas

3. Click **Increase Quota**.
4. On the **Create Service Ticket** page, configure parameters as required.

   In the **Problem Description** area, fill in the content and reason for adjustment.
5. Select the agreement and click **Submit**.

8 Related Services

OBS

OBS works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI provides an API for you to import data from corresponding OBS paths to DLI tables.

  For details about the API, see **Importing Data**.
- Data storage: You can create OBS tables on DLI. However, such tables only store metadata while data content is stored in corresponding OBS paths.
- Data backup: DLI provides an API for you to export the data in DLI to OBS for backup.

  For details about the API, see **Exporting Data**.
- Query result storage: DLI provides an API for you to save routine query result data on OBS.

  For details about the API, see **Exporting Query Result**.

IAM

Identity and Access Management (IAM) authenticates access to DLI.

For details about related operations, see **Creating an IAM User and Granting Permissions** and **Creating a Custom Policy**.

CTS

Cloud Trace Service (CTS) audits performed DLI operations.

For details about DLI operations that can be recorded by CTS, see **DLI Operations That Can Be Recorded by CTS**.

Cloud Eye

Cloud Eye helps monitor job metrics for DLI, delivering status information in a concise and efficient manner.

For details about the metrics, see **Viewing Monitoring Metrics**.

SMN

Simple Message Notification (SMN) can send notifications to users when a job running exception occurs on DLI.

If the system displays a message indicating that the SMN topic does not exist when you use an existing SMN topic, see **Why Is a Message Displayed Indicating That the SMN Topic Does Not Exist When I Use the SMN Topic in DLI?**

CDM

Cloud Data Migration (CDM) migrates OBS data to DLI.

For details, see **Importing Data to a DLI Table Using CDM**.

CloudTable

CloudTable works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import CloudTable data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to CloudTable tables.

For details about how to use a DLI datasource connection to access CloudTable data, see **Cross-Source Analysis Development Methods**.

RDS

Relational Database Service (RDS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import RDS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to RDS tables.

For details about how to use a DLI datasource connection to access RDS data, see **Cross-Source Analysis Development Methods**.

GaussDB(DWS)

GaussDB(DWS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import GaussDB(DWS) data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to GaussDB(DWS) tables.

For details about how to use a DLI datasource connection to access GaussDB(DWS) data, see **Cross-Source Analysis Development Methods**.

CSS

CSS works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import CSS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to CSS tables.

For details about how to use a DLI datasource connection to access GaussDB(DWS) data, see **Cross-Source Analysis Development Methods**.

DCS

Distributed Cache Service (DCS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import DCS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to DCS tables.

For details about how to use a DLI datasource connection to access GaussDB(DWS) data, see **Cross-Source Analysis Development Methods**.

DDS

Document Database Service (DDS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import DDS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to DDS tables.

For details about how to use a DLI datasource connection to access GaussDB(DWS) data, see **Cross-Source Analysis Development Methods**.

MRS

MapReduce Service (MRS) works as the data source and data storage system for DLI, and delivers the following capabilities:

- Data source: DLI allows you to import MRS data using DataFrame or SQL.
- Query result storage: DLI uses the SQL INSERT syntax to store query result data to MRS tables.

For details about how to use a DLI datasource connection to access GaussDB(DWS) data, see **Cross-Source Analysis Development Methods**.

9 Basic Concepts

Tenant

DLI allows multiple organizations, departments, or applications to share resources. A logical entity, also called a tenant, is provided to use diverse resources and services. A mode involving different tenants is called multi-tenant mode. A tenant corresponds to a company. Multiple sub-users can be created under a tenant and are assigned different permissions.

Project

A project is a collection of resources accessible to services. In a region, an account can create multiple projects and assign different permissions to different projects. Resources used for different projects are isolated from one another. A project can either be a department or a project team.

Database

A database is a warehouse where data is organized, stored, and managed based on the data structure. DLI management permissions are granted on a per database basis.

In DLI, tables and databases are metadata containers that define underlying data. The metadata in the table shows the location of the data and specifies the data structure, such as the column name, data type, and table name. A database is a collection of tables.

Metadata

Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.

Storage Resource

Storage resources in DLI are used to store data of databases and DLI tables. To import data to DLI, storage resources must be prepared. The storage resources reflect the volume of data you are allowed to store in DLI.

Use Elastic Resource Pool

Dedicated computing resources. They are isolated by resource pools and can only be shared by queues in the same elastic resource pool. You can set scaling policies of different priorities for these queues to adjust compute resources according to queue workload in different periods of a day.

SQL Job

SQL job refers to the SQL statement executed in the SQL job editor. It serves as the execution entity used for performing operations, such as importing and exporting data, in the SQL job editor.

Spark Job

Spark jobs are those submitted by users through visualized interfaces and RESTful APIs. Full-stack Spark jobs are allowed, such as Spark Core, DataSet, MLlib, and GraphX jobs.

CU

Compute unit (CU) is the pricing unit of queues. 1 CU consists of 1 vCPU and 4 GB memory. The computing capabilities of queues vary with queue specifications. The higher the specifications, the stronger the computing capability.

OBS Table, DLI Table, and CloudTable Table

The table type indicates the storage location of data.

- OBS table indicates that data is stored in the OBS bucket.
- DLI table indicates that data is stored in the internal table of DLI.
- CloudTable table indicates that data is stored in CloudTable.

You can create a table on DLI and associate the table with other services to achieve querying data from multiple data sources.

Constants and Variables

The differences between constants and variables are as follows:

- During the running of a program, the value of a constant cannot be changed.
- Variables are readable and writable, whereas constants are read-only. A variable is a memory address that contains a segment of data that can be changed during program running. For example, in **int a = 123**, **a** is an integer variable.