数据仓库服务(DWS)

AI Agentic 能力

文档版本	01
发布日期	2026-11-10



华为云计算技术有限公司

版权所有 © 华为云计算技术有限公司 2026。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 DWS AI Agentic 介绍.....1
2 DWS AI 插件.....3
2.1 概述.....3
2.2 hwc-dws-cli.....4
2.2.1 安装与部署.....4
2.2.2 SQL 执行.....6
2.2.3 集群管理.....7
2.2.4 API 调用.....10
2.2.5 数据导入导出.....11
2.2.6 运维监控.....12
2.2.7 元数据和语义.....14
2.2.8 审计.....16
2.2.9 调优规则.....16
2.2.10 物化视图.....17
2.3 官方 SKILL.....21
2.4 DWS MCP.....22
2.4.1 Autopilot MCP.....23
2.4.2 数据库内核 MCP.....26
2.5 对接 AI Agent.....27
2.5.1 果办 OfficeAce.....27
2.5.2 Codex.....29
2.5.3 Claude.....30
2.5.4 OpenCode.....30
3 DWS AI 智能诊断.....32
4 DWS AI Agentic 最佳实践.....33
4.1 DWS 资源诊断最佳实践.....33
4.2 DWS SQL 检查最佳实践.....36
4.3 DWS SQL 调优最佳实践.....43
4.4 DWS 知识问答最佳实践.....48
5 Data for AI 融合分析.....52
5.1 Data for AI 能力介绍.....52
5.2 基于 DWS MCP Server 搭建数据分析 Agent.....55

 版权所有 © 华为云计算技术有限公司

5.3 基于 DWS 构建 RAG 框架生成行业调研报告..... 63
5.4 基于 DWS 的向量计算功能实现简单的商品搜索推荐系统..... 69

 版权所有 © 华为云计算技术有限公司

1 DWS AI Agentic 介绍

背景与定位

DWS（Data Warehouse Service）是华为云提供的企业级数据仓库服务，具备高性能、高安全、弹性可扩展等核心能力，支撑大规模数据分析场景。随着以大模型为代表的AI Agent技术日趋成熟，如何使大模型智能体能够"理解、使用、管控"数-据仓库，已成为DWS面向智能化演进的关键课题。

DWS AI Agentic是DWS面向AI Agent构建的智能化能力体系，旨在以标准化、可编程、可治理的方式，将DWS的查询、管理、运维与调优能力对外开放给AI生态，使各类AI Agent能够以接近专业数据工程师的水平使用DWS。

关键概念

为便于理解本文档，先对涉及的几个核心概念做简要说明：

概念	说明
AI Agent（智能体）	以大语言模型为大脑、能够理解用户意图并自主调用外部工具完成任务的智能程序。本指南中指可对接DWS的各类智能体，如Claude、Codex、OpenCode等。
SKILL（技能）	面向特定场景封装好的“能力包”，包含触发词、执行流程与行为约束，AI Agent加载后即可通过自然语言触发对应能力，如SQL调优、资源诊断等。
MCP（Model Context Protocol，模型上下文协议）	由Anthropic于2024年11月提出的开放协议标准，用于标准化大模型与外部系统（如数据库、API）的交互方式，使大模型能动态理解并调用工具。
hwc-dws-cli	DWS面向AI Agent与开发者提供的一体化命令行工具，是DWS AI Agentic能力体系的基础底座。
Autopilot	DWS内置的监控运维平台，提供集群监控指标采集能力，是AI智能诊断的数据来源。

能力体系

DWS AI Agentic的能力体系由三大部分构成：

版权所有 © 华为云计算技术有限公司

组成部分	说明	参考文档
DWS AI 插件	• **hwc-dws-cli**：面向 AI Agent 与开发者的一体化命令行工具，覆盖 SQL 查询、元数据、集群管理、数据导入导出、运维视图、监控与 SQL 调优等能力。 • **DWS MCP**基于 MCP（Model Context Protocol）协议开放 DWS 工具能力。 • **SKILL**：提供资源分析、SQL调优、DWS知识问答等AGENT能力集。	**DWS AI 插件**
DWS AI智能诊断	基于AI Agent 能力，用户可以在华为云数据仓库服务控制台实现智能化诊断，包括**计算资源（内存、CPU、IO）、集群状态、SQL语句**的诊断。	**DWS AI智能诊断**
DWS AI 实践	基于以上能力的最佳实践与场景化指南，包括如何接入主流 AI Agent、典型使用流程与注意事项。	**DWS AI Agentic最佳实践**

三者的关系如下：

- **AI 插件**是能力底座，打通AI Agent到DWS物理环境的连接能力。
- **AI智能诊断**提供Web页面的卡片式诊断能力，面向控制台用户。
- **AI 实践**则回答“如何用好”的问题，帮助用户把能力落地到真实业务场景。

核心设计理念

DWS AI Agentic在设计上遵循以下原则：

- **AI 原生**：从接口设计、协议规范到安全模型，均以AI Agent的调用方式为首要考量，而非简单复用面向人类的操作界面。
- **安全可控**：写操作需显式授权，高危操作需二次确认；SQL执行前经过权限门控与分类检查，全链路审计留痕，在赋予 AI 数据库操作能力的同时确保风险可控。
- **标准化开放**：基于通用CLI与MCP标准协议，实现一次接入、多处可用，生态内主流AI Agent均可对接。
- **可治理**：全量操作可审计、可回放、可追溯，满足企业级数据安全与合规要求。

版权所有 © 华为云计算技术有限公司

2 DWS AI 插件

2.1 概述

hwc-dws-cli是DWS面向AI Agent与开发者提供的命令行插件，将DWS的数据库访问、集群管理、运维能力整合为统一的命令行入口，构成DWS AI Agentic能力体系的基础底座。

插件定位：

- **面向 AI Agent**：命令结果统一以结构化格式（JSON 等）输出，语义稳定，便于 AI Agent 直接解析与决策。
- **面向开发者**：同时提供表格、CSV 等易读格式，确保 DBA 与数据工程师获得一致的运维体验。
- **面向自动化场景**：支持脚本化调用、统一的退出码约定与错误码体系，便于纳入 CI/CD 流水线与日常巡检流程。

核心特性

hwc-dws-cli的核心特性如表2-1所示。

表 2-1 核心特性

特性	说明
SQL执行	支持 DQL/DML/DDL/DCL/TCL 全类型SQL，覆盖单条执行、SQL文件批量执行与参数化查询。
元数据管理	库 / Schema / 表 / 列 / 分区检索，表结构描述，DDL导出，语义元数据（semantic）构建与检索。
集群管理	基于DWS开放API的集群信息查询，覆盖集群、参数组、审计、数据源、负载、扩缩容、逻辑集群、升级、SQL防误操作、自动vacuum full、事件、告警、备份容灾等。
数据导入导出	支持 CSV / JSON / Parquet 文件上传下载，结合DWS COPY实现高性能加载与断点续传。

版权所有 © 华为云计算技术有限公司

特性	说明
运维视图	内置20余个预置只读运维查询，覆盖会话、锁、活跃查询、等待事件、死锁、内存、Top SQL、GUC 参数、资源池等。
集群监控	提供集群级指标、主机列表与告警信息查询。
SQL 调优	规则引擎驱动的 SQL审查与调优建议，支持自定义规则与成本护栏。
安全管控	默认只读 + 分级授权 + 危险操作二次确认，SQL分类识别与全链路审计。

2.2 hwc-dws-cli

2.2.1 安装与部署

环境要求

使用hwc-dws-cli前，请确认满足以下环境要求：

- Python 3.10 及以上。
- pip工具可用。
- 可访问华为云 DWS 服务（需获取 IAM 凭证 AK/SK）。
- DWS 已配置 EIP（用于 SQL 查询，配置项可选）。
- 已**下载hwc-dws-cli工具**。

安装方式

插件基于 Python 打包分发，提供两种安装方式：

```
# linux bash/git bash安装
sh ./install.sh
# Powershell安装
.\install.bat
```

安装完成后可通过以下命令校验：

```
hwc-dws-cli --version
```

若回显版本号，说明安装成功；若提示命令未找到，请检查环境变量是否已正确配置。

部署配置

插件运行在用户侧（命令行环境），通过配置连接华为云DWS。配置项如**表2-2**所示。

版权所有 © 华为云计算技术有限公司

表 2-2 部署配置项

配置项	说明
IAM 凭证	AK/SK、区域（Region）、项目 ID
数据库连接	主机、端口、用户、密码、数据库名、SSL 模式
运行模式	同步 / 异步（外部任务队列、WITH HOLD 游标）
输出格式	json（默认）/ table / csv / vertical

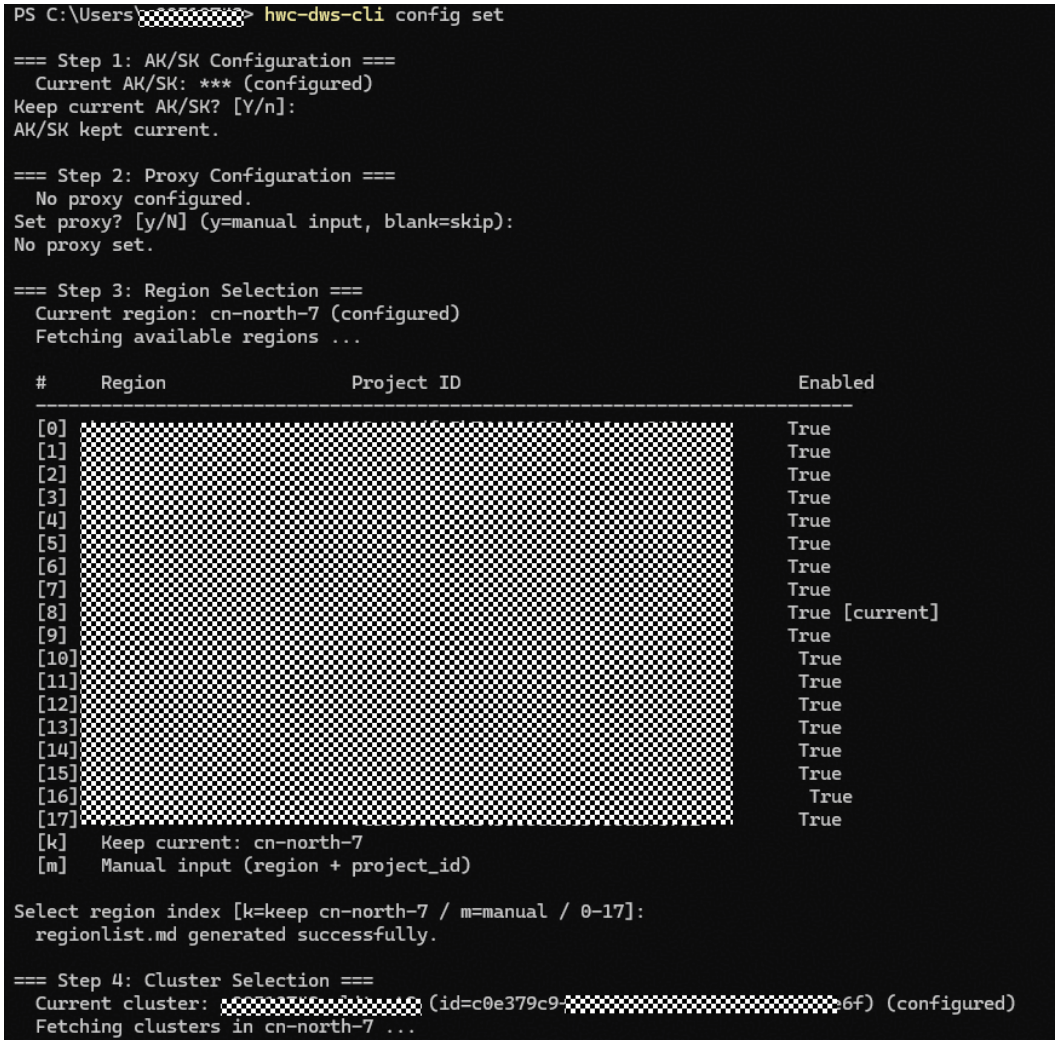
配置支持交互式向导（CLI 向导或 Web 向导）、配置文件与环境变量三种方式，环境变量优先级最高，便于在容器与自动化环境中注入凭证。配置中的敏感字段展示时自动脱敏。

安装完成后可通过以下命令配置：

```
hwc-dws-cli config set
```

以下为配置示例：

图 2-1 配置过程



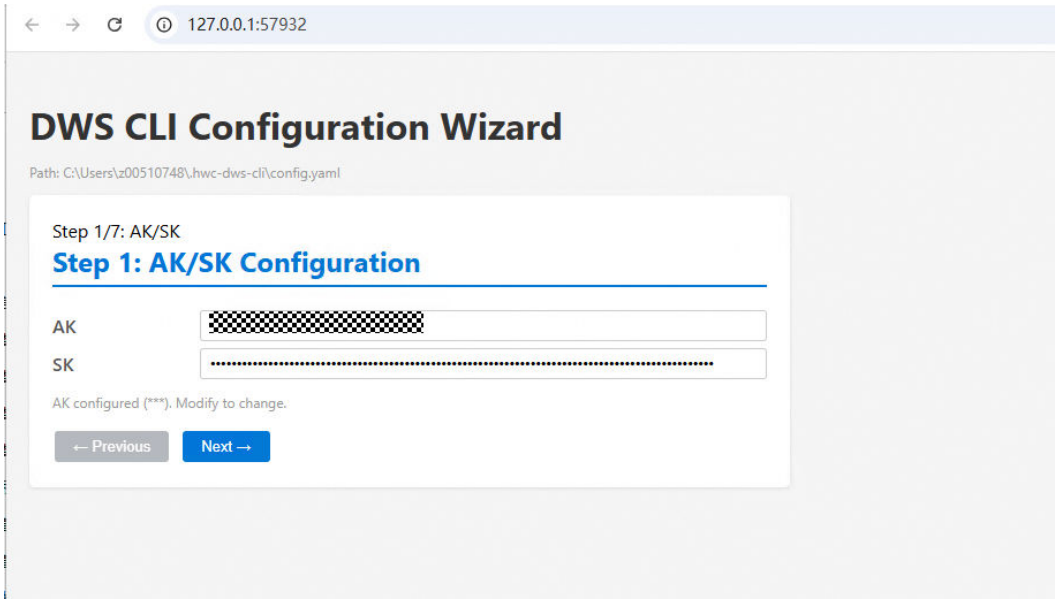
 版权所有 © 华为云计算技术有限公司

图 2-2 配置成功




```
hwc-dws-cli config set web
```

图 2-3 配置 web



2.2.2 SQL 执行

SQL执行是hwc-dws-cli的核心能力之一，支持AI Agent和开发者通过命令行直接向DWS集群提交SQL语句。默认以JSON结构化格式输出结果，便于AI Agent解析；同时支持table、csv等易读格式，满足人工查看需求。

安全模型： SQL执行默认只读（DQL），写操作（DML/DDL）需显式追加 --write，危险操作（DROP/TRUNCATE等）需追加 --write --confirm-dangerous，确保AI Agent不会意外修改或删除数据。

SQL执行命令参数说明如表2-3所示。

版权所有 © 华为云计算技术有限公司

表 2-3 SQL 执行参数

参数	说明
--exec <SQL>	待执行SQL语句（与 --file 二选一）。
--file <path>	执行SQL文件（与 --exec 二选一）。
--param <v>	SQL参数，可重复指定（参数化查询）。
--dry-run	仅生成执行计划（EXPLAIN VERBOSE），不实际执行。

支持 DQL / DML / DDL / DCL / TCL 全类型 SQL，高危操作/写操作需要单独指定参数。

```
# 执行查询（默认 JSON Envelope 输出）
hwc-dws-cli sql --exec "SELECT current_database()" --format table

# 执行 SQL 文件
hwc-dws-cli sql --file script.sql

# 参数化查询（--param 可重复）
hwc-dws-cli sql --exec "SELECT * FROM t WHERE id = $1 AND name = $2" --param 1 --param "alice"

# EXPLAIN VERBOSE 仅生成执行计划，不实际执行
hwc-dws-cli sql --exec "DELETE FROM t WHERE id=1" --dry-run

# 写操作（需 --write）
hwc-dws-cli sql --exec "INSERT INTO t VALUES (1)" --write

# 危险操作（需 --write --confirm-dangerous）
hwc-dws-cli sql --exec "DROP TABLE t" --write --confirm-dangerous
```

2.2.3 集群管理

集群管理提供基于DWS OpenAPI的集群信息只读查询能力，覆盖集群、参数组、审计、数据源、负载、扩缩容、逻辑集群、升级、SQL防误操作、事件、告警、备份容灾等15个类别共77条端点。

使用场景： AI Agent可通过集群管理命令获取集群运行状态、资源配置、审计日志等信息，用于智能诊断、运维巡检和容量规划。所有命令均为只读查询，不会修改集群配置。

集群 ID 自动从 config 注入，可用 --cluster-id 覆盖；--dry-run 输出 endpoint / method / path / params 预览而不发起调用。

表 2-4 集群管理命令

子命令组	命令
cluster list	clusters（--api-version v1|v2，默认 v2，--limit/--offset 分页）、statistics
cluster show	info、node-types、flavor、elbs、cns、instances、endpoints、volume、encrypt-info、actions <action_name>、instance <instance_id>、nodes、topo-rings、topology、resource-statistics

版权所有 © 华为云计算技术有限公司

```
# 集群列表（v2 API 默认）
hwc-dws-cli cluster list clusters --format table
hwc-dws-cli cluster list clusters --api-version v1 --limit 50 --offset 0
hwc-dws-cli cluster list statistics

# 集群详情 / 规格 / 节点
hwc-dws-cli cluster show info --cluster-id xxx-xxx-xxx
hwc-dws-cli cluster show node-types
hwc-dws-cli cluster show flavor
hwc-dws-cli cluster show nodes
hwc-dws-cli cluster show topology

# 含路径参数的查询
hwc-dws-cli cluster show actions restart
hwc-dws-cli cluster show instance 5f9c9f0e-xxxx-xxxx

# 预览 API 调用（不实际执行）
hwc-dws-cli cluster show info --dry-run
```

表 2-5 参数组、审计与数据源

子命令组	命令
cluster configurations	list、show <configuration_id>、audit-records、security <configuration_id>
cluster audit	log-records
cluster datasource	list

```
hwc-dws-cli cluster configurations list
hwc-dws-cli cluster configurations show 8c2b5f1e-xxxx
hwc-dws-cli cluster configurations audit-records
hwc-dws-cli cluster configurations security 8c2b5f1e-xxxx
hwc-dws-cli cluster audit log-records
hwc-dws-cli cluster datasource list
```

表 2-6 工作负载与资源池

子命令	命令
cluster workload	status、queue-jobs、rules
cluster workload plans	list、show <plan_id>、logs <plan_id>、stage <plan_id> <stage_id>
cluster workload pools	list、show <pool_id>（--pool-name 可用名称替代）、users <pool_name>、plans <pool_name>

```
# 工作负载
hwc-dws-cli cluster workload status
hwc-dws-cli cluster workload queue-jobs
hwc-dws-cli cluster workload rules

# 负载计划
hwc-dws-cli cluster workload plans list
```


版权所有 © 华为云计算技术有限公司

```
hwc-dws-cli cluster workload plans show my_plan
hwc-dws-cli cluster workload plans logs my_plan
hwc-dws-cli cluster workload plans stage my_plan stage_1

# 资源池
hwc-dws-cli cluster workload pools list
hwc-dws-cli cluster workload pools show my_pool
hwc-dws-cli cluster workload pools users my_pool
hwc-dws-cli cluster workload pools plans my_pool
```

表 2-7 扩缩容、逻辑集群、升级、事件告警

子命令组	命令
cluster scale-out	redistribution、redistribution-schemas
cluster scale-up	storage-expand-range
cluster vw	list、tasks、volumes、rings、plans
cluster upgrade	avail-versions、records
cluster sql-block	rules <database>
cluster auto-vacuumfull	list、detail <task_id>、configuration
cluster event	subs、specs、list
cluster alarm	subs、list、statistic、configs

```
# 扩容重分布
hwc-dws-cli cluster scale-out redistribution
hwc-dws-cli cluster scale-out redistribution-schemas

# 磁盘扩容范围
hwc-dws-cli cluster scale-up storage-expand-range

# 逻辑集群（vw）
hwc-dws-cli cluster vw list
hwc-dws-cli cluster vw tasks
hwc-dws-cli cluster vw volumes

# 升级
hwc-dws-cli cluster upgrade avail-versions
hwc-dws-cli cluster upgrade records

# SQL 防误操作拦截
hwc-dws-cli cluster sql-block rules postgres

# 自动 vacuum full
hwc-dws-cli cluster auto-vacuumfull list
hwc-dws-cli cluster auto-vacuumfull detail <task_id>
hwc-dws-cli cluster auto-vacuumfull configuration

# 事件 / 告警
hwc-dws-cli cluster event list
hwc-dws-cli cluster event subs
hwc-dws-cli cluster alarm list
hwc-dws-cli cluster alarm statistic
```


版权所有 © 华为云计算技术有限公司

表 2-8 备份容灾

子命令	命令
cluster backup-dr	snapshots、snapshot <snapshot_id>、snapshot-statistics、cluster-snapshots、snapshot-policies、snapshot-policy <policy_id>、snapshot-policy-plan、cross-region-snapshot-policies、check-name <name>、roach-configs、databases、database-schemas <database>、cross-regions、cross-region-policies、backup-files <snapshot_id>、disaster-recoveries、disaster-recovery <id>、disaster-recovery-clusters、restore-schemas <snapshot_id> <database>、disaster-recovery-progress <id>

```
# 快照
hwc-dws-cli cluster backup-dr snapshots

# 名称检查 / 数据库与 schema
hwc-dws-cli cluster backup-dr check-name my-snapshot
hwc-dws-cli cluster backup-dr databases
hwc-dws-cli cluster backup-dr database-schemas postgres

# 跨区域策略 / 容灾
hwc-dws-cli cluster backup-dr cross-region-snapshot-policies
hwc-dws-cli cluster backup-dr cross-regions
hwc-dws-cli cluster backup-dr cross-region-policies
hwc-dws-cli cluster backup-dr disaster-recoveries
hwc-dws-cli cluster backup-dr disaster-recovery <id>
hwc-dws-cli cluster backup-dr disaster-recovery-clusters
hwc-dws-cli cluster backup-dr disaster-recovery-progress <id>

# 按快照恢复 schema
hwc-dws-cli cluster backup-dr restore-schemas <snapshot_id> postgres
```

2.2.4 API 调用

通用API调用子命令说明如表2-9所示。

表 2-9 API 调用子命令

子命令	说明
api GET <path>	通用 GET 请求任意 DWS REST API 路径，占位符 {project_id} / {cluster_id} 自动从配置填充。
api list-quotas	列出 DWS 配额。
api list-projects [--domain-id]	列出 IAM 项目（可指定 IAM domain）。

所有请求经 APICaller 以 HMAC-SHA256 签名，端点域名按 config.active.region 自动构造（https://dws.<region>.myhuaweicloud.com；list-projects 走 IAM 端点）。

```
# 通用 GET 通道（占位符自动填充）
hwc-dws-cli api GET /v1.0/{project_id}/clusters
```


版权所有 © 华为云计算技术有限公司

```
hwc-dws-cli api GET /v1.0/{project_id}/clusters/{cluster_id}/snapshots
hwc-dws-cli api GET /v1.0/{project_id}/clusters/{cluster_id}/configurations

# 配额 / IAM 项目
hwc-dws-cli api list-quotas
hwc-dws-cli api list-projects
hwc-dws-cli api list-projects --domain-id d00xxxxxxx
```

⚠ 注意

api GET 仅支持 {project_id} 与 {cluster_id} 两个占位符；其他占位符会报 PATH.PLACEHOLDER.UNRESOLVED。

2.2.5 数据导入导出

数据导入导出是hwc-dws-cli的常用能力，支持将本地文件（CSV/JSON/Parquet）批量上传到DWS表、从DWS表或查询结果导出数据到本地文件，以及通过外表（OBS/HDFS）实现跨数据源导入导出。适用于数据初始化、批量迁移、数据采样等场景。

数据导入导出子命令说明如表2-10所示。

表 2-10 数据导入导出子命令

子命令	说明
data upload --file <f> --table <t> [--schema] [--format csv|json|parquet]	上传数据文件到表。支持批量 INSERT 与 DWS COPY FROM STDIN（--copy，高吞吐）两种模式；多线程（--threads）与块大小（--block-size）调优。
data download --file <f> [--table | --query] [--format csv|json]	导出表数据或查询结果到 CSV / JSON 文件，支持行数 / 大小上限控制。
data show --table <t> [--limit N]	查看表样例数据。
data purge --table <t> [--schema]	清空表（TRUNCATE，**需 --confirm-dangerous**）。
data resume --task-id <id>	恢复暂停的数据传输会话（断点续传）。
data foreign create	创建外表（OBS / HDFS），自动生成 DISTRIBUTE BY，预校验 format×mode 约束。
data foreign import	从外表导入数据到本地表（INSERT INTO ... SELECT）
data foreign export	将本地表数据导出到外表。

上传模式说明：

- **批量INSERT模式（默认）：** 逐行拼接INSERT语句提交，适合中小数据量。通过 --threads 指定并发线程数提升吞吐。

版权所有 © 华为云计算技术有限公司

- **COPY高性能模式（--copy）：** 使用DWS原生COPY FROM STDIN协议，批量流式写入，吞吐量显著高于INSERT模式，适合大数据量导入。可通过 --copy-options 传递delimiter、null、encoding等COPY选项。
- **断点续传：** 大文件传输中断后，可使用 data resume --task-id <id> 恢复传输会话，避免从头重传。

```
# 上传（默认 csv，batch INSERT）
hwc-dws-cli data upload --file data.csv --table fact_sales --schema dwd

# 上传（parquet 格式，多线程）
hwc-dws-cli data upload --file data.parquet --table fact_sales --format parquet --threads 8

# 上传（COPY 高性能模式，自定义选项）
hwc-dws-cli data upload --file big.csv --table fact --copy --copy-options "delimiter=|,null=\\N"

# 下载（按表 / 按查询）
hwc-dws-cli data download --table fact_sales --file out.csv
hwc-dws-cli data download --query "SELECT * FROM fact_sales WHERE day='2026-10-01'" --file out.json --format json
hwc-dws-cli data download --table fact_sales --file head.csv --max-record 1000 --max-size 10485760
```

DWS外表操作

```
# 创建外表（OBS，自动 DISTRIBUTE BY）
hwc-dws-cli data foreign create --table ext_sales --server obs_server \
  --type obs --columns "id:bigint,name:varchar(100),amt:numeric(10,2)" \
  --format csv --foldername "obs://bucket/data" --schema dwd

# 预览外表 DDL（不执行）
hwc-dws-cli data foreign create --table ext_sales --server obs_server --type obs \
  --columns "id:bigint" --format orc --foldername "obs://bucket/data" --dry-run

# 外表导入 / 导出
hwc-dws-cli data foreign import --foreign-table ext_sales --local-table fact_sales --schema dwd
hwc-dws-cli data foreign export --local-table fact_sales --foreign-table ext_sales --schema dwd
```

2.2.6 运维监控

运维分两段展示：SQL 直查（集群实时状态，18 个只读命令）与 Autopilot API（DMS 监控）。

表 2-11 SQL 直查

编号	子命令	说明
1	session	活动会话（非 idle），--node/--limit 过滤。
2	lock	当前锁。
3	active	活动查询。
4	conn	连接汇总（按 database / user / addr / application）。
5	pool	连接池汇总。
6	queue	工作负载队列状态。
7	user	数据库用户与角色。

版权所有 © 华为云计算技术有限公司

编号	子命令	说明
8	detail [--pid]	会话详情（可按 PID 过滤）。
9	perf	数据库性能统计。
10	wait	等待事件会话。
11	wait-node	按节点等待事件。
12	respool	资源池。
13	deadlock	潜在死锁（未授予事务锁）。
14	tablelock	表级锁。
15	lwlock	轻量锁等待。
16	memory	按节点内存使用。
17	topsql	Top SQL（按总耗时）。
18	guc [--name]	GUC 参数（pg_settings，可按名称 LIKE 过滤）。

```
# 会话 / 连接
hwc-dws-cli ops session --format table
hwc-dws-cli ops session --node cn --limit 50
hwc-dws-cli ops conn --format table
hwc-dws-cli ops pool

# 锁与死锁
hwc-dws-cli ops lock --format table
hwc-dws-cli ops tablelock
hwc-dws-cli ops lwlock
hwc-dws-cli ops deadlock

# 等待事件 / 详情
hwc-dws-cli ops wait
hwc-dws-cli ops wait-node
hwc-dws-cli ops detail --pid 12345

# 资源与性能
hwc-dws-cli ops respool
hwc-dws-cli ops queue
hwc-dws-cli ops user
hwc-dws-cli ops perf
hwc-dws-cli ops memory --format table

# Top SQL / GUC
hwc-dws-cli ops topsql --limit 20
hwc-dws-cli ops guc --name work_mem
```

表 2-12 Autopilot 监控（DMS API）

命令	说明
ops metric <name> [--from] [--to] [--limit] [--offset]	集群指标数据（DMS API，时间戳为毫秒，默认近 1 小时）。

版权所有 © 华为云计算技术有限公司

命令	说明
ops metric-list [--keyword] [--limit]	可用 DWS 指标名列表（支持客户端关键词过滤）。
ops host-list [--instance-name] [--limit]	集群主机概览。
ops alarm show [--from] [--to] [--time-zone] [--limit]	告警列表（DBSEvent API，默认近 6 天）。
ops alarm search [--keyword] [--cluster] [--severity] [--status]	告警搜索（客户端过滤：关键词 / 集群 / 级别 / 状态）。

```
# 指标数据（RFC3339 时间，默认近 1 小时）
hwc-dws-cli ops metric cpu_usage --format table
hwc-dws-cli ops metric disk_usage --from 2026-10-14T00:00:00Z --to 2026-10-14T12:00:00Z

# 指标名列表（可按关键词过滤）
hwc-dws-cli ops metric-list --keyword cpu

# 主机概览
hwc-dws-cli ops host-list --format table
hwc-dws-cli ops host-list --instance-name dn_1

# 告警（show 与 search）
hwc-dws-cli ops alarm show --format table
hwc-dws-cli ops alarm search --severity urgent --format table
hwc-dws-cli ops alarm search --keyword "connection" --cluster <cluster_id> --status 0
```

2.2.7 元数据和语义

表 2-13 元数据查询

子命令	说明
meta list-schemas	列出所有 schema。
meta list-tables [--schema <s>]	列出表（可按 schema 过滤）。
meta list-databases	列出所有数据库。
meta describe <table> [--schema <s>]	表结构富信息（8 段：基本信息 / 时间戳 / 存储 / 分布 / 分区 / 列 / 索引 / 扩展信息），支持 schema.table 输入与 cache-first 描述。
meta search-tables <keyword>	按名称关键词搜索表（ILIKE）。
meta search-columns <keyword>	按名称关键词搜索列（ILIKE）。
meta list-partitions <table> [--schema <s>]	列出表分区。
meta export-ddl <table> [--schema <s>]	导出表 DDL（CREATE TABLE）。

版权所有 © 华为云计算技术有限公司

子命令	说明
meta export-meta <table> [--schema <s>]	导出单表元数据（类型、估算行数、页数）。

```
# 列出 schema / 表 / 数据库
hwc-dws-cli meta list-schemas
hwc-dws-cli meta list-tables --schema public
hwc-dws-cli meta list-databases --format table

# 表结构富信息（两种等价写法）
hwc-dws-cli meta describe ads_live.ads_anchor_ranking --format table
hwc-dws-cli meta describe ads_anchor_ranking --schema ads_live

# 搜索 / 分区 / 导出
hwc-dws-cli meta search-tables order
hwc-dws-cli meta search-columns user_id
hwc-dws-cli meta list-partitions fact_sales --schema dwd
hwc-dws-cli meta export-ddl fact_sales --schema dwd
hwc-dws-cli meta export-meta fact_sales --schema dwd
```

标注：meta semantic 系列命令为**体验功能**，接口与行为可能随版本演进调整，不建议在正式自动化流程中依赖。

表 2-14 语义元数据

子命令	说明
meta semantic build [--schema <s>] [--db-name <n>] [--no-progress]	触发 8 阶段语义构建 pipeline。
meta semantic get <table> [--schema <s>]	获取表的语义元数据。
meta semantic set <table> [--description] [--use-cases] [--schema <s>]	存储表的语义描述。
meta semantic list-missing [--schema <s>]	列出尚无语义元数据的表。
meta semantic search <pattern>	按模式搜索语义元数据（LIKE / FTS5）。
meta semantic delete <table> [--schema <s>]	删除表的语义元数据。
meta semantic show [<table>] [--schema <s>] [--scope overview|table|joins|udfs]	语义概览或表详情。

```
# 触发 8 阶段语义构建（可限定 schema）
hwc-dws-cli meta semantic build --schema public

# 查看语义概览 / 表详情
hwc-dws-cli meta semantic show --scope overview
```


版权所有 © 华为云计算技术有限公司

```
hwc-dws-cli meta semantic show demo --schema public --scope table
hwc-dws-cli meta semantic show --schema public --scope joins

# 读写单表语义
hwc-dws-cli meta semantic set demo --description "订单明细表" --use-cases "电商订单分析"
hwc-dws-cli meta semantic get demo --schema public

# 检索与维护
hwc-dws-cli meta semantic search "订单"
hwc-dws-cli meta semantic list-missing --schema public
hwc-dws-cli meta semantic delete demo --schema public
```

2.2.8 审计

表 2-15 审计子命令

子命令	说明
audit show [--limit N] [--action <s>]	查看最近审计条目（默认 20 条，按 seq 降序）。
audit search [--action] [--actor] [--from] [--to] [--limit]	多维度过滤审计日志（默认 100 条）。
audit clean [--days N] --yes	清理过期审计条目（默认保留 30 天，**需 --yes**）。

审计日志为本地 HMAC 链式存储（~/.hwc-dws-cli/store.db 的 audit_log），SQL 以摘要形式留存，不记录明文 SQL。

```
# 查看最近审计（按 action 过滤）
hwc-dws-cli audit show --limit 50 --action SQL
hwc-dws-cli audit show --action API

# 多维搜索（按 action / 操作者 / 时间区间）
hwc-dws-cli audit search --action API --actor dbadmin --from 2026-10-01 --to 2026-10-13
hwc-dws-cli audit search --action SQL --limit 200

# 清理 7 天前的审计条目（需确认）
hwc-dws-cli audit clean --days 7 --yes
```

2.2.9 调优规则

CLI 不直接执行 SQL 诊断（诊断由 hwc-dws-sql-tuning Skill 完成），tune 负责规则配置管理：启用 / 禁用规则、增删自定义规则，Agent 在执行调优前读取配置。

表 2-16 调优规则子命令

子命令	说明
tune config show [--format table]	显示全部规则 + 启用 / 禁用状态 + 自定义规则（语法 46 + 执行计划 27 + PERFORMANCE 扩展 11 + 对象 34 = 118 条内置规则）。

版权所有 © 华为云计算技术有限公司

子命令	说明
tune config set <key> <value>	设置配置项（如 global.tuning.disabled_rules "UPR-001,UOR-15"）。
tune config add-rule [--file] [--id] [--name] ...	添加自定义规则（YAML 文件或逐项参数；集群类规则需只读 diagnostic_sql）。
tune config remove-rule <rule_id>	删除自定义规则。

```
# 查看全部规则与状态
hwc-dws-cli tune config show
hwc-dws-cli tune config show --format table

# 禁用规则（逗号分隔）
hwc-dws-cli tune config set global.tuning.disabled_rules "UPR-001,UOR-15"

# 自定义规则（YAML 文件 / 逐项参数）
hwc-dws-cli tune config add-rule --file custom_rule.yaml
hwc-dws-cli tune config add-rule --id CUSTOM-001 --name "禁止全表扫描" \
  --category plan --level error --description "扫描行数超过阈值时告警"

# 删除自定义规则
hwc-dws-cli tune config remove-rule "CUSTOM-001"
```

2.2.10 物化视图

物化视图（Materialized View）是DWS中预计算并物化的查询结果，可加速复杂聚合查询。hwc-dws-cli提供物化视图子命令组，支持查看物化视图列表、依赖树、DDL定义、刷新管理、调度修复、级联重建等功能，帮助DBA全面掌握物化视图的运行状况。

表 2-17 全局参数

参数	说明
--dry-run	预览SQL，不执行；仅写命令支持。
--verbose	同步打印执行的SQL。
--format <json|\table|\csv|\vertical>	结果的输出格式。

表 2-18 物化视图只读子命令

子命令	说明
matview list [--schema <s>] [--owner <o>]	列出所有物化视图，含依赖层级、分区标记、属主、大小。
matview tree <view>	展示物化视图 / 基表的级联依赖树（DAG），含每个节点的直接依赖。

版权所有 © 华为云计算技术有限公司

子命令	说明
matview define <view> [--output <f>]	查询物化视图及其级联依赖的 DDL 定义语句，支持导出到文件。
matview grants <view> [--output <f>]	查询物化视图及其级联依赖的 GRANT 语句（ALTER OWNER + GRANT），支持导出到文件。
matview partition <view>	查询物化视图的分区刷新状态，以及基表与物化视图的分区映射。
matview duration [<view>] [--schema <s>] [--owner <o>] [--name <n>] [--partition] [--history] [--limit <N>] [--broken-only] [--stale-only]	查询物化视图刷新耗时，省略视图名（<view>）时扫描全库。 • [--partition] 查询物化视图各分区的刷新耗时。 • [--history] 查询最近10条刷新历史记录，仅适用于单视图模式（指定<view>参数），不可与 [--partition] 同时使用。 • [--name <n>] 支持使用通配符按名称模糊匹配， % 匹配任意长度字符串，_ 匹配单个字符。

<view> 参数为物化视图或基表的名称，需带上所属schema前缀，格式为schema.name 。

```
# 列出所有物化视图（按 schema / owner 过滤）
hwc-dws-cli matview list --format table
hwc-dws-cli matview list --schema public --owner dbadmin

# 查看基表 / 物化视图的级联依赖树
hwc-dws-cli matview tree public.mview

# 查询物化视图的 DDL 定义（可写入文件）
hwc-dws-cli matview define public.mview --output mv_ddl.sql

# 查询物化视图的 GRANT 语句（可写入文件）
hwc-dws-cli matview grants public.mview --output mv_grant.sql

# 查询物化视图的分区刷新状态和映射关系
hwc-dws-cli matview partition public.mview

# 查询单个物化视图的刷新耗时
hwc-dws-cli matview duration public.mview
hwc-dws-cli matview duration public.mview --partition
hwc-dws-cli matview duration public.mview --history

# 全库扫描所有物化视图刷新耗时摘要（可按 schema / owner 过滤、 名称模糊匹配）
hwc-dws-cli matview duration --schema public --owner dbadmin --name mv% --limit 20
```


版权所有 © 华为云计算技术有限公司

表 2-19 物化视图写子命令

子命令	说明
matview refresh <view> [--partition <s>] [--resetrict] [--cascade <forward|backward|all>]	刷新物化视图数据。 ● [--partition] 指定分区名进行分区刷新，不可与 [--cascade] 同时使用。
matview alter <view> [--rename <s>] [--query-rewrite <enable|\disable>] [--refresh-start <s>] [--refresh-every <s>] [--refresh-restrict] [--refresh-on-demand] [--set <key=value>] [--reset <key>]	修改物化视图。 ● [--refresh-start] 刷新启动表达式，如 now() 或 '2026-01-01 16:00:00'::timestamp 。 ● [--refresh-every] 刷新间隔表达式，如 interval '30 second' 。 ● [--refresh-restrict] 不进行级联刷新，必须与 [--refresh-start] 或 [--refresh-every] 同时使用。
matview rebuild <view> [--sql <s>] [--sql-file <f>] [--extra-users <s>] [--grant <s>] [--output <f>] [--no-lock] [--kill-sessions <prompt|always|never>] [--lock-timeout <N>]	级联重建物化视图 / 普通视图 / 基表。 ● 重建DDL语句为必填，使用 [--sql] 直接输入或 [--sql-file] 从文件中读取，二者不能同时使用。 ● 支持以下 <view>类型 + DDL 组合 1. 物化视图 + CREATE MATERIALIZED VIEW：删除重建物化视图 2. 普通视图 + CREATE OR REPLACE VIEW：替换视图 3. 基表 + ALTER TABLE：级联重建基表 4. 基表 + CREATE TABLE <table>_new：重建新表，并将旧表数据备份至 <table>_old；DDL 中新表名必须带_new后缀，重建完成后会自动重命名至原名。 ● [--kill-sessions]：DROP锁冲突时kill策略， prompt - 交互确认（默认） / always - 直接 kill / never -不 kill，中止重建。 ● [--lock-timeout]：DROP/RENAME锁等待超时秒数，默认5s，超时触发被动响应查持锁会话。仅约束锁等待；DDL 后立即清零，不影响后续重建/数据填充。

版权所有 © 华为云计算技术有限公司

子命令	说明
matview rebuild-batch [--sql <s>] [--sql-file <f>] [--extra-users <s>] [--grant <s>] [--output <f>] [--no-lock] [--kill-sessions <prompt|always|never>] [--lock-timeout <N>]	批量级联重建多个同质目标，重复依赖只DROP/CREATE一次。无<view>参数，目标名从DDL解析。 ● 重建DDL语句为必填，使用 [--sql] 直接输入或 [--sql-file] 从文件中读取，可重复，但二者不能同时使用。DDL语句必须同质（例如全 ALTER TABLE、全 CREATE MV ……） ● 支持类型及参数与 rebuild 命令相同。
matview repair [<view>] [--schema <s>] [--all] [--stale-factor <N>] [--stall-intervals <N>]	修复 stale 和 broken 的物化视图刷新调度。 ● <view>、[--schema]、[--all] 三选一：<view> 修复单张指定物化视图；[-schema] 扫描指定schema下所有物化视图，进行批量修复，[--all] 扫描所有schema下物化视图，进行批量修复。 ● broken：后台调度任务因异常被标记中断，刷新任务不再自动执行。 ● [--stale-factor]：过期容忍因子，默认为1.0。age > interval * stale-factor 为过期，stale-factor 越小越严格。 ● [--stall-intervals]：停滞阈值，默认为3。过期interval最大可超过的值。 ● 同时满足以下三个条件的物化视图判定为stalled，需要修复 1. 已过期（age > interval * stale-factor） 2. 调度未激活或已停止（active=false） 3. 过期时间超过N个interval（停滞阈值）
matview drop <view> [--cascade] [--kill-sessions <prompt|always|never>] [--lock-timeout <N>]	删除物化视图。 ● [--cascade]：级联删除目标及所有上游依赖。

执行写命令需要 --write 参数；执行 drop 命令需要 --write 和 --confirm-dangerous 参数。

版权所有 © 华为云计算技术有限公司

⚠ 注意

alter、drop、repair、rebuild、rebuild-batch 为事务性命令，失败或中断后自动回滚；refresh为非事务命令，提交后无法回滚。执行写命令前建议使用 --dry-run 进行预览。

```
# 刷新物化视图
hwc-dws-cli matview refresh public.mview --write
hwc_dws_cli matview refresh public.part_mview --partition p1 --write

# 修改物化视图
hwc-dws-cli matview alter public.mview --rename mview_new --query-rewrite enable --refresh-start
"now()" --refresh-every "interval '30s'" --write
hwc-dws-cli matview alter public.mview --set mv_refresh_guc=enable_nestloop=off --reset mv_pck_column
--write

# 删除物化视图
hwc-dws-cli matview drop public.mview --write --confirm-dangerous
hwc-dws-cli matview drop public mview drop public.mview --cascade --kill-sessions always -lock-timeout 30
--write --confirm-dangerous

# 修复物化视图
hwc-dws-cli matview repair public.mview --dry-run    # dry-run 预览是否需要修复
hwc-dws-cli matview repair public.mview --write
hwc-dws-cli matview repair --all --stale-factor 0.5 --stall-intervals 2 --write

# 重建物化视图
hwc-dws-cli matview rebuild public.mview --sql "CREATE MATERIALIZED VIEW public.mview AS SELECT *
FROM public.table WHERE a > 10" --write
hwc-dws-cli matview rebuild public.view --sql "CREATE OR REPLACE VIEW public.view AS SELECT * FROM
public.table WHERE a > 1" --write
hwc-dws-cli matview rebuild public.table --sql "ALTER TABLE public.table MODIFY c bigint" --kill-sessions
never --lock-timeout 10 --write
hwc-dws-cli matview rebuild public.table --sql "CREATE TABLE public.table_new AS SELECT * from
public.table" --extra-users user_read --grant SELECT --write

# 批量重建多个同质目标
hwc-dws-cli matview rebuild-batch --sql "CREATE MATERIALIZED VIEW public.mv1 AS SELECT * FROM
public.t WHERE a > 10"
  --sql "CREATE MATERIALIZED VIEW public.mv2 AS SELECT t1.a AS t1a, t1.b AS t1b, t2.a AS t2a, t2.b AS t2b
FROM public.table t1 LEFT JOIN public.table2 t2 ON t1.a = t2.a WHERE t2.b > 1"
  --kill-sessions prompt --lock-timeout 10 --no-lock --write
```

2.3 官方 SKILL

面向华为云DWS数据仓库的DBA日常运维、故障诊断与调优场景，DWS提供一组官方SKILL，将专业知识封装为AI Agent可直接调用的能力，覆盖知识问答、SQL调优和资源类故障根因诊断三大能力域。Agent加载对应SKILL后，用户通过自然语言即可触发专业诊断流程，无需手工编写复杂查询。

官方Skill清单如表2-20所示。

版权所有 © 华为云计算技术有限公司

表 2-20 Skill 总览

Skill 名称	能力域	功能	典型触发词
hwc-dws-knowledge-qa	知识问答	DWS产品文档离线知识库问答。	DWS集群、数据仓库、SQL语法、GAUSS-错误码、GDS、列存/行存、HStore、集群扩容、快照、资源池
hwc-dws-sql-tuning	SQL 调优	DWS SQL 统一调优诊断：语法/执行计划/对象多维度规则检查，输出结构化诊断报告。	SQL调优、SQL诊断、SQL优化、SQL性能、执行计划、SQL审查、SQL tuning
hwc-dws-cpu-diag	资源诊断	CPU 使用率过高/告警根因诊断。	CPU高、CPU告警、CPU诊断、CPU使用率过高、CPU满
hwc-dws-cn-session-diag	故障诊断	CN 并发数/活跃会话超阈值根因诊断。	CN并发数、活跃会话数量使用率、并发打满、already too many clients、无法连接数据库
hwc-dws-deadlock-diag	故障诊断	死锁数量超阈值根因诊断。	死锁、deadlock、锁冲突、锁等待、长事务、事务阻塞
hwc-dws-disk-diag	空间诊断	磁盘使用率高/磁盘满根因诊断。	磁盘使用率高、磁盘满、磁盘告警、只读、下盘、数据倾斜
hwc-dws-dymem-diag	资源诊断	内存使用率高/OOM 根因诊断。	内存高、内存告警、内存诊断、OOM、内存溢出、动态内存使用率超阈值
hwc-dws-io-diag	资源诊断	I/O 过载根因诊断（三阶段决策树）。	I/O高、I/O告警、IO过载、磁盘IO负载异常、util打满、await高

2.4 DWS MCP

版权所有 © 华为云计算技术有限公司

2.4.1 Autopilot MCP

简介

MCP（Model Context Protocol）是由 Anthropic 于 2024 年 11 月提出的开放协议标准，旨在解决大型语言模型与外部系统（如数据库、API）交互的碎片化问题。通过标准化接口，让 LLM 动态理解工具功能并执行操作，降低集成成本。

DWS Autopilot MCP Server 是 DWS AI 诊断体系中的关键组件，为 Skill 提供集群监控指标查询的工具接口。Skill 通过 MCP Server 获取集群的 CPU、内存、I/O 等实时监控数据，作为诊断的事实依据。

用户 → AI 对话窗口 → Skill（诊断逻辑）→ MCP Server（指标查询）→ DWS Autopilot API → 集群监控数据

单击**github地址**获取。

功能介绍

工具

DWS Autopilot MCP Server 提供以下工具：

1. **dws_autopilot_get_clusters** 查询 DWS 集群列表，无需参数。
2. **dws_autopilot_get_hosts** 查询 DWS 集群节点信息，支持过滤、分页、排序。
   - 必填：cluster_id（字符串）— 集群 ID
   - 可选过滤：filter（host_name / work_ip）、value、sub_filter、sub_value
   - 可选分页：page_size（默认 10，最大 2000）、page_num（默认 1）、offset（默认 0）、limit（默认 512）
   - 可选排序：order_by（cpu_usage、mem_usage、disk_usage_avg、disk_io、tcp_resend_rate、net_io）、sort_by（ASC / DESC，默认 DESC）
3. **dws_autopilot_get_metric** 查询 DWS 集群指标时序数据。
   - 必填：cluster_id（字符串）、metric_name（字符串）、from_ts（13 位 Unix 时间戳，毫秒）、to_ts（13 位 Unix 时间戳，毫秒）
   - 支持的指标：cpu_usage、mem_usage、disk_usage_avg、tcp_resend_rate、disk_io、net_io、cpu_io_diagnose_detail（返回包含 query_id 和 query 字段的逐查询详情）
   - 可选：offset（默认 0）、limit（最大 1000，默认 50）、order_by、sort_by（ASC / DESC）

认证方式

采用 AK/SK 签名认证：

- **AK/SK 签名模式**：使用华为云 Access Key（AK）和 Secret Key（SK）对每个 API 请求进行签名，签名后自动添加 Authorization 和 X-Sdk-Date 请求头完成认证。
- 如果配置了 project_id，会自动添加 X-Project-Id 请求头，用于多项目场景。

配置加密

敏感字段（AK、SK）支持 AES-256-GCM 加密存储：

版权所有 © 华为云计算技术有限公司

- 启动 MCP Server 时自动检测明文并加密
- 基于机器指纹的密钥保护，绑定当前机器环境
- 主密钥双重保护：crypter（存储在 yaml）+ cryptComponent（存储在 crypt.json）

安装与配置

环境准备

- Python 3.10 及以上
- pip 包管理器

安装

```
# 克隆源码
git clone <repo_url>
cd dws_autopilot_mcp

# 安装依赖
pip install .
```

配置说明

配置文件位于 conf/dws_config.yaml，首次使用需填写配置：

```
region_id: "cn-north-7"
ak: "your_access_key"
sk: "your_secret_key"
project_id: "your_project_id"
http_proxy: "http://proxy.example.com:8080"
https_proxy: "http://proxy.example.com:8080"
proxy_username: "your_proxy_user"
proxy_password: "your_proxy_password"
```

表 2-21 参数说明

参数	说明	示例
region_id	华为云区域 ID，用于自动生成 API 端点	cn-north-7
ak	华为云 Access Key（明文输入，启动后自动加密）	—
sk	华为云 Secret Key（明文输入，启动后自动加密）	—
project_id	项目 ID，用于多项目场景的 X-Project-Id 请求头	—
http_proxy	HTTP 代理地址（可直接包含认证信息，也可通过 proxy_username/proxy_password 单独配置）	http://proxy.example.com:8080
https_proxy	HTTPS 代理地址	http://proxy.example.com:8080

版权所有 © 华为云计算技术有限公司

参数	说明	示例
proxy_username	代理认证用户名（自动注入代理 URL，密码自动 percent-encoding）	—
proxy_password	代理认证密码（自动 percent-encoding 并注入代理 URL）	—

region_id 会自动生成 DMS_MONITORING_BASE_URL（https://dws.{region_id}.myhuaweicloud.com），无需手动配置。

配置管理 CLI

安装后提供 dws-mcp-config 命令行工具，需在项目根目录（dws_autopilot_mcp/）下执行：

init - 初始化/更新配置

init 命令用于设置配置参数：

```
python -m dws_autopilot_mcp.config_cli init --region_id cn-north-7 --ak your_ak --sk your_sk --project_id your_project_id
```

init 命令支持部分更新，仅更新指定的参数，未指定的参数保持不变。例如只更新 AK：python -m dws_autopilot_mcp.config_cli init --ak new_ak。

更新配置后，重启 MCP Server 即可自动加密敏感字段（AK、SK）。

其他命令

```
# 手动触发加密（将明文敏感字段加密，启动 MCP Server 时也会自动执行）
python -m dws_autopilot_mcp.config_cli encrypt

# 查看配置状态（敏感信息以 ****** 显示，不会泄露）
python -m dws_autopilot_mcp.config_cli show

# 重置配置为空模板（清空所有参数，删除 crypt.json）
python -m dws_autopilot_mcp.config_cli reset
```

encrypt、show、reset 命令无需额外参数。

客户端配置

以 OpenCode 为例，在 MCP Servers 配置中添加：

```
{
  "dws_autopilot_mcp": {
    "type": "local",
    "command": [
      "/path/to/python.exe",
      "-m",
      "dws_autopilot_mcp.server"
    ],
    "environment": {
      "PYTHONPATH": "/path/to/dws_autopilot_mcp/src"
    }
  }
}
```


版权所有 © 华为云计算技术有限公司

将 /path/to/python.exe 替换为本机 Python 可执行文件路径，/path/to/dws_autopilot_mcp/src 替换为本项目 src 目录的绝对路径。Windows 下 PYTHONPATH 使用 ; 分隔，Linux/macOS 使用 : 分隔。

也可通过环境变量 DWS_MCP_CONFIG 指定配置文件路径。

开始体验

完成配置后，即可在客户端中通过自然语言与 DWS 集群交互，例如：

- "查询 XXXX 集群节点信息"
- "查看 XXXX 集群最近 1 小时的 CPU 使用率"
- "我的 DWS 集群 CPU 使用率很高，帮忙诊断一下"
- "检查这条 SQL 的语法：SELECT * FROM t1"

MCP Server 在 AI 诊断中的角色

在 DWS AI 诊断体系中，MCP Server 承担以下角色：

表 2-22 MCP Server 在 AI 诊断中的角色

角色	说明
数据通道	作为 Skill 获取集群监控指标的通道
参数补全	I/O 诊断中，当用户未提供 project_id/region_id 时，自动从 MCP Server 配置文件中读取
指标查询	提供 dws_autopilot_get_metric 工具，支持 CpuStat、MemStat、IOStat、InstanceMemory、cpu_io_diagnose_detail、memory_diagnose_detail 等指标查询

2.4.2 数据库内核 MCP

简介

在企业数据分析和运营决策过程中，业务人员常需从数据仓库提取数据并生成分析报告。传统模式下，业务人员需将数据需求提交给开发人员，由开发人员编写SQL语句查询数据，再将结果反馈给业务人员。这一流程通常耗时数天，频繁的沟通迭代进一步拖慢了决策效率。

随着大语言模型（LLM）技术的成熟，是否存在一种方式能让业务人员直接使用自然语言与数据库交互，无需编写SQL即可完成数据查询与分析?

DWS MCP Server正是为解决这一需求而生的方案。它基于**MCP（Model Context Protocol）协议**，将DWS数据库的元数据查询、SQL执行等能力以标准化工具接口暴露给大模型，使LLM能够自动理解数据库结构、生成SQL语句并执行查询，最终将查询结果进行业务解读与可视化呈现。

版权所有 © 华为云计算技术有限公司

功能介绍

DWS MCP Server目前支持元数据查询、语句执行、监控信息查询等基本功能，以MCP协议中的工具（Tools）和资源（Resource）两种形式向支持MCP协议的客户端暴露。相关工具和资源参见表2-23和表2-24。

表 2-23 数据库管理工具（Tools）

名称	描述
list_databases	列出所有数据库。
get_activity	从 pgxc_stat_activity视图获取最近的查询活动。
execute_query	执行 SQL 查询。
list_schemas	列出当前数据库中的所有模式。
list_tables	列出指定模式下的所有表。
list_views	列出指定模式下的所有视图。
get_table_info	获取表/视图的定义。
get_comment	获取模式/表的注释。

表 2-24 可用资源（Resources）

资源URI	功能说明
gaussdb:///{schema}/tables	列出指定模式下的所有表
gaussdb:///{schema}/views	列出指定模式下的所有视图
gaussdb:///{schema}/{table}/attributes	列出指定表/视图所有的列
system:///{system_path}	系统信息（例如 /version）

使用实践

DWS MCP Server使用流程请参见**基于DWS MCP Server搭建数据分析Agent**。

2.5 对接 AI Agent

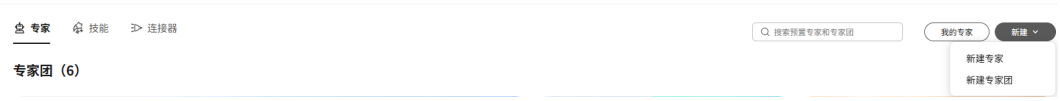
2.5.1 果办 OfficeAce

果办OfficeAce致力于将人工智能深度融入日常办公场景，成为用户身边的AI办公工作台。借助果办的专家能力，可以快速集成hwc-dws-cli。

版权所有 © 华为云计算技术有限公司

步骤1 下载果办OfficeAce与安装部署请参考**OfficeAce产品文档**。

步骤2 登录OfficeAce客户端界面， 在左侧导航栏，选择“专家·技能·连接器”，选择“专家”页签，在右上角单击“我的专家”，单击“新建 > 新建专家”。



1. 基础信息的名称、描述，由用户自定义。
2. 将AGENTS.md的内容复制粘贴到“创建专家”的灵魂配置中。



步骤3 添加SKILL。



步骤4 使用技能分析集群负载。



----结束

版权所有 © 华为云计算技术有限公司

2.5.2 Codex

前提条件

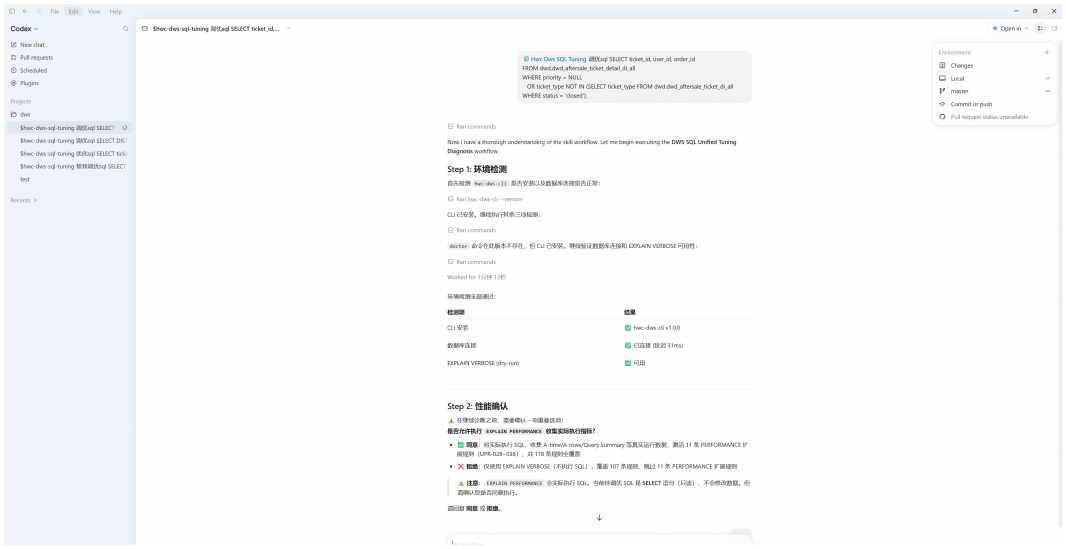
- 已安装hwc-dws-cli。
- 将skill配置到Codex。复制 skill 目录到 ~\.codex\skills\下。

操作步骤

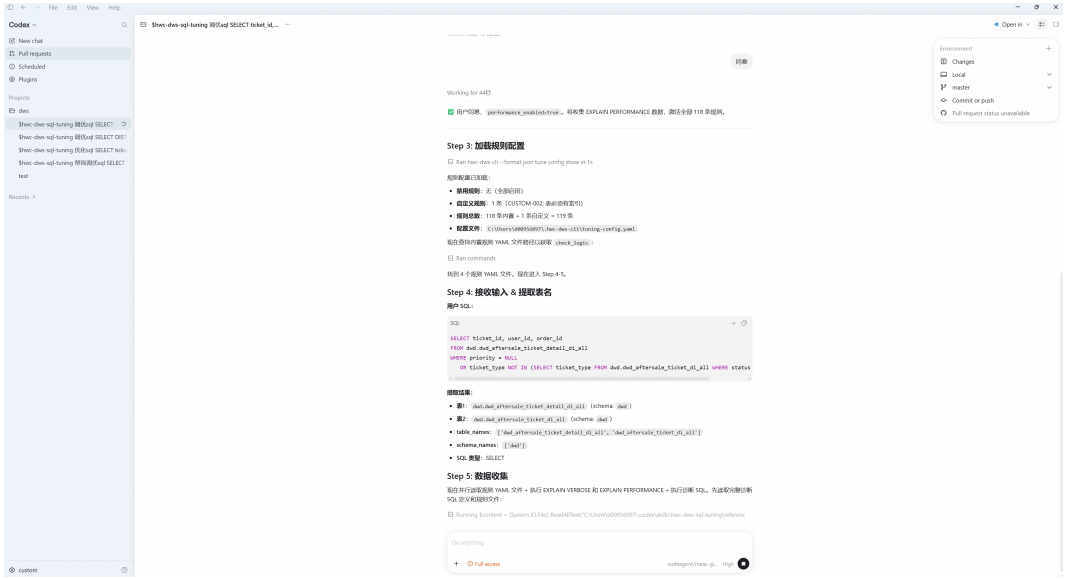
以SQL调优skill为例（本 skill 依赖hwc-dws-cli和集群连接。若未安装或未配置，skill 会直接中止，需修复后重新调用。）

步骤1 在Codex会话中输入触发词（"SQL调优" / "SQL诊断" 等）并附上待诊断的 SQL 文本。

步骤2 Codex询问是否允许执行EXPLAIN PERFORMANCE（会实际执行 SQL，DML 将修改数据；可选"允许"或"仅 EXPLAIN VERBOSE"）。



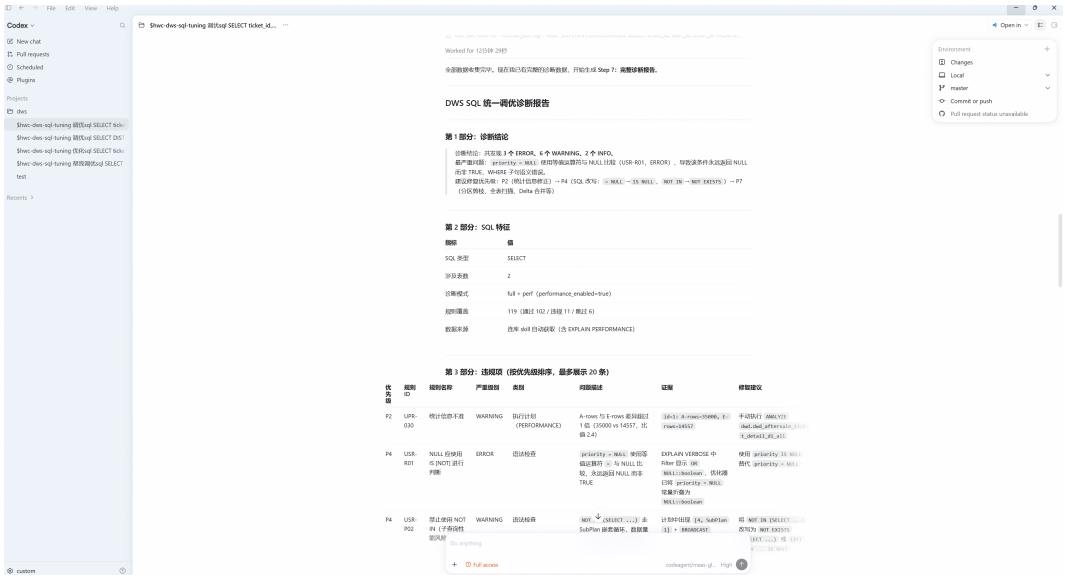
步骤3 等待自动诊断完成（环境检测 → 规则加载 → 数据收集 → 规则检查）。



版权所有 © 华为云计算技术有限公司

步骤4 查看诊断结果：

输出七部分报告（诊断结论 / SQL特征 / 违规 / 优化建议 / 优化后SQL+Hint / 性能验证 / 统计）。



----结束

2.5.3 Claude

前提条件

- 已安装Claude及配置API模型，请参考**Claude官方网站**。
- 已下载安装部署hwc-dws-cli。

操作步骤

步骤1 创建项目目录，例如：/home/myprj。

步骤2 复制AGENTS.md到/home/myprj并重命名为CLAUDE.md。

步骤3 复制hwc-dws-cli目录下的agent_skills目录中的全量目录到~/.claude/skills/。

步骤4 进入/home/myprj，并启动claude。

----结束

2.5.4 OpenCode

前提条件

- 已安装OpenCode及配置API模型，请参考**OpenCode官方网站**。
- 已下载安装部署hwc-dws-cli。

操作步骤

步骤1 创建项目目录，例如：/home/myprj，复制AGENTS.md到/home/myprj。

版权所有 © 华为云计算技术有限公司

步骤2 安装SKILL。

- 全局：复制hwc-dws-cli目录下的agent_skills目录中的全量skill到~/.config/opencode/skills/。
- 本项目：复制hwc-dws-cli目录下的agent_skills目录中的全量skill到/home/myprj/.opencode/skills/。

步骤3 进入/home/myprj，并启动opencode。

----结束

版权所有 © 华为云计算技术有限公司

3 DWS AI 智能诊断

随着企业数据仓库中数据量与业务复杂度持续增长，集群运维人员需要时刻关注计算资源使用情况、集群运行状态以及SQL执行效率等关键指标，以确保数据仓库稳定高效运行。

然而，当出现内存溢出、CPU使用率飙高、IO瓶颈、集群节点异常或慢SQL等问题时，传统的人工排查方式需要运维人员逐一查看多项监控指标、翻阅日志并凭借经验判断根因，不仅耗时耗力，且在问题复杂或多因素交织的情况下难以快速准确定位，严重影响业务连续性。

那么，能否让系统自动完成上述诊断分析，帮助用户快速定位问题根因并给出可执行的优化建议?

华为云数据仓库服务基于AI Agentic能力，在DWS控制台提供智能化诊断功能。用户可针对**计算资源（内存、CPU、IO）**、**集群状态**、**SQL语句**三大维度一键发起智能诊断，系统将自动采集并关联分析相关指标与日志，快速定位问题根因，并输出结构化诊断报告与优化建议，显著降低运维门槛，提升问题处置效率。

DWS AI智能诊断功能陆续在华为云各Region开通中，如需使用请联系技术支持。

版权所有 © 华为云计算技术有限公司

4 DWS AI Agentic 最佳实践

4.1 DWS 资源诊断最佳实践

当DWS集群出现CPU、内存或I/O等资源异常时，定位根因往往依赖运维经验，过程繁琐且耗时。DWS集群诊断面向该场景，提供资源异常的自动根因分析能力：用户只需在AI Agent对话窗口中用自然语言描述异常现象（如"CPU使用率过高"），系统即可自动采集监控指标、分析根因（区分客户侧与系统侧），并输出标准化诊断报告，帮助用户快速定位并解决问题。

功能概述

DWS集群诊断覆盖以下三类资源异常诊断，适用场景如表4-1所示。

表 4-1 诊断类型与适用场景

诊断类型	适用场景
CPU高诊断	CPU使用率过高、CPU负载异常。
内存高诊断	内存使用率过高、OOM（内存溢出）、动态内存超阈值。
I/O过载诊断	I/O使用率过高、磁盘I/O负载异常、磁盘利用率（util）打满、I/O等待时间（await）偏高。

整个诊断流程基于DWS Autopilot监控指标自动完成指标采集、根因分析与报告生成，**无需连接集群执行SQL**，即可获得诊断结论。

使用方式

诊断Skill可通过任意支持 MCP 协议的 AI Agent 使用，包括但不限于：

- **AI Shell**：华为云提供的在线 AI 交互工具，开箱即用，无需本地部署。访问地址：**https://developer.huaweicloud.com/aishell.html?ticket=ST-8760627-1bUREQ0uFwaeEdfuBESoEd7H-sso&locale=zh-cn**

版权所有 © 华为云计算技术有限公司

- **自建 Agent**：用户可在自己的 AI Agent 中安装诊断 Skill 并配置 DWS Autopilot MCP Server，实现相同的诊断能力

无论使用哪种 Agent，触发方式和诊断流程一致。

触发诊断

在 AI Agent 对话窗口中描述集群资源异常即可触发对应诊断，系统会自动识别意图并匹配对应 Skill：

表 4-2 触发诊断

诊断类型	触发词示例
CPU 高	"CPU高"、"CPU诊断"、"CPU使用率过高"、"CPU满"
内存高	"内存高"、"内存诊断"、"内存不足"、"OOM"、"内存溢出"
I/O 过载	"I/O高"、"I/O诊断"、"IO过载"、"磁盘IO负载异常"、"util打满"、"await高"

触发示例：

```
用户：我的DWS集群CPU使用率很高，帮忙诊断一下
    集群ID：xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
    项目ID：xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
用户：DWS集群内存不足，集群ID是xxx，项目ID是yyy
```

提供诊断参数

触发诊断后，系统会根据需要收集诊断参数：

表 4-3 提供诊断参数

参数	必选/可选	说明
cluster_id	必选	集群 ID（实例 ID）
project_id	必选（I/O 诊断可选）	项目 ID

其他信息（如区域、集群名称等）系统会自动获取，无需手动提供。

诊断执行流程

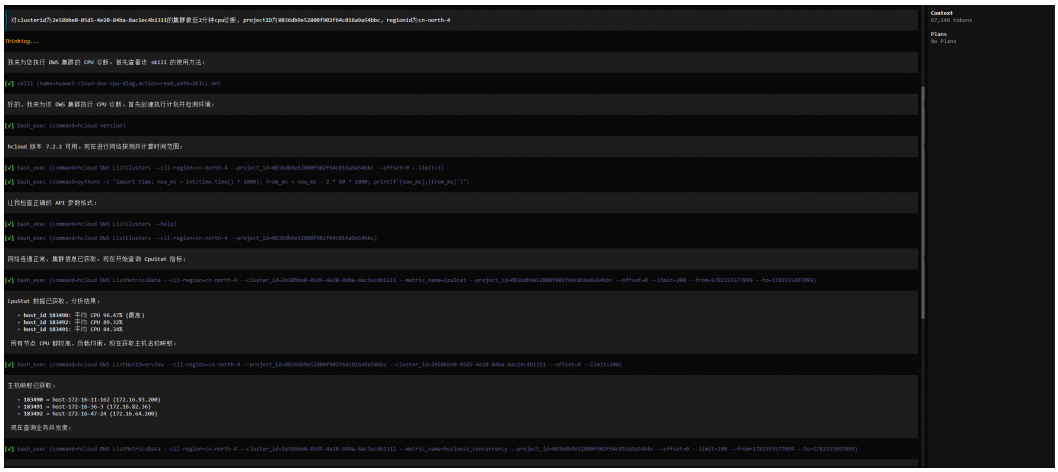
诊断被触发后，将自动执行以下流程：

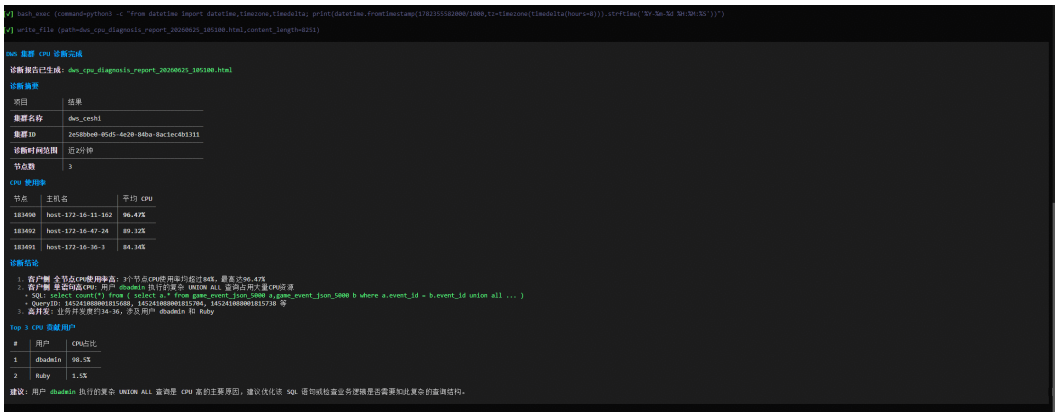
```
1. 指标采集 → 采集集群监控指标
2. 根因分析 → 分析根因，区分客户侧/系统侧
3. 报告生成 → 输出标准化诊断报告
```

查看诊断报告

诊断完成后，系统会生成诊断报告，可直接查看。

版权所有 © 华为云计算技术有限公司





CPU 高诊断

自动采集集群各节点 CPU 使用率、业务并发度、活跃查询、进程 CPU 占用等指标，定位 CPU 高的根因，区分客户侧与系统侧原因。

报告内容

- **基本信息**：集群名称、区域、集群 ID、TOP 节点。
- **诊断结论**：根因分类与事实描述，定位到具体 SQL 时附带 SQL 摘要、QueryID、用户名、启动时间。
- **CPU 贡献 Top3 语句**：QueryID、用户、CPU 占比、时间、SQL 文本。
- **CPU 贡献 Top3 用户**：用户名与 CPU 占比。

内存高诊断

自动采集集群各节点内存使用率、实例级动态内存/进程内存、活跃会话内存占用、memory_pool 数据等指标，定位内存高的根因，区分客户侧与系统侧原因。

报告内容

- **诊断结论**：异常类型与事实描述，定位到具体 SQL 时附带定位信息。
- **内存贡献 Top3 语句**：QueryID、用户、内存占比、时间、SQL 文本。
- **内存贡献 Top3 数据库用户**：用户名与内存占比。

版权所有 © 华为云计算技术有限公司

I/O 过载诊断

自动采集集群各节点磁盘利用率、I/O 响应时间、吞吐量、IOPS、iowait、活跃查询、等待事件、锁冲突等指标，自动判定 I/O 场景并路由到对应排查方向。

报告内容

- **诊断结论**：异常类型与事实描述。
- **问题节点 I/O 概览**：节点磁盘利用率、响应时间、吞吐量等。
- **I/O 贡献 Top3 SQL**：QueryID、用户、I/O 占比、SQL 文本。
- **I/O 贡献 Top3 用户**：用户名与 I/O 占比。
- **排查信息**：按场景输出的排查方向。

注意事项

- **数据真实性**：所有诊断结论来自实际监控指标，指标不可达时标记为"无法获取"，不编造数据。
- **报告格式**：诊断报告仅输出诊断结论，不添加修复建议、不输出 SQL 优化语句。
- **时间显示**：报告中所有时间均为北京时间（UTC+8）。
- **安全**：对话中不会暴露 AK/SK 等凭证信息。

4.2 DWS SQL 检查最佳实践

在SQL开发与上线前，开发人员需要对SQL进行语法和规范审查，以提前发现语法错误和不符合开发规范的写法，避免执行阶段才暴露问题。传统人工审查效率低、易遗漏，且依赖个人对规范的熟悉程度。DWS SQL检查面向该场景，对SQL语句进行静态分析与执行计划检查，帮助用户在执行前快速发现并修正问题，提升SQL质量。

本功能与SQL调优使用同一skill，按诊断深度区分使用场景：

- **SQL 检查（本文）**：只做静态分析与执行计划检查，**不实际执行 SQL**（不运行 EXPLAIN PERFORMANCE），适用于开发阶段快速审查
- **SQL 调优**：在检查基础上进一步执行 EXPLAIN PERFORMANCE 采集实际性能指标，输出优化后 SQL 与性能验证（见**DWS SQL调优最佳实践**）

Skill 地址：hwc-dws-sql-tuning（随 hwc-dws-cli 分发，源码位于 dws-cli 仓库 agent_skills 目录）。

前提条件

本 skill 依赖本地 `hwc-dws-cli` 与 DWS 集群连接，未安装或未配置时**直接中止**（无降级模式）。调用前自检：

```
hwc-dws-cli --version      # CLI 是否已安装
hwc-dws-cli config show    # 配置文件是否可读
hwc-dws-cli check-con      # 集群连接是否正常
```

先完成配置再调用：

```
hwc-dws-cli config set     # 7步向导：AK/SK → 代理 → Region → 集群 → Endpoint → 连接 → 连通性验证
```


版权所有 © 华为云计算技术有限公司

功能概述

DWS SQL 检查支持以下检查模式：

表 4-4 检查功能

模式	说明	是否需要集群连接
语法检查	检查关键字合法性、语句结构完整性、子句顺序、DWS 特有语法兼容性	否
规范检查	检查对象设计标准、数据操作规范、命名规范、SQL 开发最佳实践	否
执行计划检查	检查执行计划中的下推、Stream、重分布、倾斜、估算偏差、分区裁剪、下盘	是（EXPLAIN VERBOSE 不执行 SQL）

本检查模式对应 skill 的 full + verbose 模式：使用 EXPLAIN VERBOSE（只生成执行计划、不实际执行 SQL），不做 EXPLAIN PERFORMANCE。

使用方式

在 AI Agent 对话窗口中直接提供需要检查的 SQL 语句即可触发检查，支持以下典型用法：

表 4-5 使用方式

触发方式	示例
检查 SQL 语法	"检查这条 SQL 的语法：SELECT * FROM t1"
检查建表规范	"这条 CREATE TABLE 是否符合 DWS 规范？"
验证 DWS 特有语法	"验证这个 MERGE 语句的语法"
规范审查	"审查我的 SQL 是否符合开发规范"
指定检查模式	"仅做语法检查：DELETE FROM t1"

也可通过触发词快速调用：SQL检查、SQL规范、SQL审计、SQL语法、SQL优化、检查SQL、SQL review。

检查规则与优化建议解读

语法检查

语法检查用于发现 SQL 语句中的语法错误，完整支持 DWS 特有语法。

版权所有 © 华为云计算技术有限公司

表 4-6 基础语法检查

检查项	级别	说明	优化建议
词法错误	ERROR	SQL 文本中存在无法识别的字符或格式错误。	检查 SQL 文本中是否有非法字符或编码问题。
无效关键字	ERROR	使用了 DWS 不支持的关键字。	替换为 DWS 支持的等效关键字。
保留关键字用作标识符	ERROR	保留关键字不能用作标识符，除非用双引号引用。	使用双引号引用标识符，或改用其他名称。
语法结构错误	ERROR	缺少必选子句或关键字。	补充缺失的子句或关键字。
子句顺序错误	ERROR	SQL 子句顺序不符合语法规范	按正确顺序排列子句：SELECT→FROM→WHERE→GROUP BY→HAVING→ORDER BY→LIMIT

表 4-7 DWS 特有语法检查

检查项	级别	说明	优化建议
DISTRIBUTE BY 语法	ERROR	分布策略不合法，有效值为 HASH/MODULO/REPLICATION/ROUNDROBIN。	使用有效的分布策略。
PARTITION BY 语法	ERROR	分区定义语法不合法，有效值为 RANGE/LIST/INTERVAL。	使用有效的分区策略。
MERGE 语句语法	ERROR	MERGE INTO...USING...ON...WHEN 结构不完整。	补全 MERGE 语句的必需子句。
EXPLAIN 语法	ERROR	EXPLAIN 选项不合法。	使用 DWS 支持的 EXPLAIN 选项（PERFORMANCE/WARMUP/PLAN）。
COMPRESS 语法	ERROR	COMPRESS 选项不合法，有效值为 YES/NO。	使用 COMPRESS YES 或 COMPRESS NO。
TIMECAPSULE 语法	ERROR	TIMECAPSULE TABLE...TO BEFORE {DROP|TRUNCATE} 结构不合法。	按正确语法结构编写 TIMECAPSULE 语句。

版权所有 © 华为云计算技术有限公司

检查项	级别	说明	优化建议
资源管理对象语法	ERROR	CREATE RESOURCE POOL / WORKLOAD GROUP / REDACTION POLICY / OUTLINE 结构不合法。	参照 DWS 语法手册修正结构。
TO NODE/ GROUP 语法	ERROR	TO NODE/GROUP 子句语法不合法。	按正确语法编写 TO NODE/GROUP 子句。
INSERT OVERWRITE 语法	ERROR	INSERT OVERWRITE 语句结构不合法。	按正确语法编写 INSERT OVERWRITE 语句。
ON DUPLICATE KEY UPDATE 语法	ERROR	ON DUPLICATE KEY UPDATE 子句语法不合法。	按正确语法编写 ON DUPLICATE KEY UPDATE 子句。
Oracle (+) 外连接语法	WARNING	(+) 运算符使用不当。	建议改用标准 JOIN 语法。
Optimizer Hint 语法	WARNING	Hint 格式不合法。	按 /*+ hint_name(table) */ 格式编写 Hint。

规范检查

规范检查基于 DWS 开发设计规范，分为四类：

表 4-8 对象设计规范

检查项	级别	说明	优化建议
CREATE TABLE 缺少 DISTRIBUTE BY	ERROR	创建表时必须指定分布策略。	添加 DISTRIBUTE BY 子句指定分布策略。
CREATE TABLE 缺少主键	INFO	建议定义主键约束。	添加 PRIMARY KEY 约束。
CREATE TABLE 缺少表注释	INFO	表未添加注释，不利于维护。	添加 COMMENT ON TABLE 语句。
分布键列不在列定义中	WARNING	分布键应为表中实际存在的列。	确保分布键列在列定义中存在。
分区键与分布键相同	INFO	可能导致数据倾斜。	选择不同的列作为分区键和分布键。
REPLICATION 表不宜过大	WARNING	大表会在每个 DN 上存储完整副本。	大表使用 HASH 分布策略。

版权所有 © 华为云计算技术有限公司

检查项	级别	说明	优化建议
ROUNDROBIN 分布性能风险	INFO	不支持本地关联，可能导致网络重分布。	评估是否可使用 HASH 分布。
自定义 TABLESPACE 需确认场景	WARNING	分布式场景可能导致存储倾斜（列存 v3 表除外）。	确认使用场景，列存 v3 表除外。
建议显式指定存储方式	WARNING	建议通过 orientation 参数显式指定行存/列存。	添加 WITH (orientation=row/column)。
行存表禁止指定 compress	ERROR	显式指定行存（orientation=row）时禁止使用 COMPRESS。	移除 COMPRESS 子句或改用列存。
大表建议设计分区	INFO	改善查询性能和数据治理效率。	添加 PARTITION BY 子句。
字段建议添加 NOT NULL	INFO	优化器可在特定场景下利用 NOT NULL 优化。	为非空字段添加 NOT NULL 约束。
避免使用自增列	WARNING	SERIAL 等类型会造成 GTM 压力。	使用 SEQUENCE 替代 SERIAL。
单表索引个数不宜超过 5 个	WARNING	冗余索引浪费空间，影响入库效率（需集群连接）。	审查并删除冗余索引。
视图嵌套不超过三层	INFO	嵌套过深导致执行计划不稳定（需集群连接）。	简化视图嵌套层级。

表 4-9 数据操作规范

检查项	级别	说明	优化建议
禁止使用 SELECT *	ERROR	应明确指定字段列表。	替换 SELECT * 为具体字段列表。
DELETE/UPDATE 缺少 WHERE	ERROR	必须包含 WHERE 条件。	添加 WHERE 条件，防止误操作全表。
NOT IN 子查询性能风险	WARNING	建议改用 NOT EXISTS 或 LEFT JOIN。	将 NOT IN 改写为 NOT EXISTS 或 LEFT JOIN + IS NULL。
DISTINCT 可能影响性能	INFO	大数据量下排序去重可能影响性能。	评估是否可用 GROUP BY 替代。
隐式类型转换风险	WARNING	可能导致索引失效。	确保比较操作两侧类型一致，使用显式类型转换。

版权所有 © 华为云计算技术有限公司

检查项	级别	说明	优化建议
LIKE 前缀通配符	WARNING	LIKE '%...' 无法使用索引。	避免前缀通配符，或使用全文检索。
OR 条件性能风险	INFO	可能影响优化器选择最优计划。	考虑改写为 UNION ALL。
IN 列表过长	WARNING	超过 100 个值建议使用临时表。	将 IN 列表替换为临时表 JOIN。
FROM 子句中的子查询	INFO	建议改用 CTE。	将子查询改写为 WITH 子句（CTE）。
Oracle 外连接语法	INFO	建议使用标准 JOIN。	将 (+) 外连接改写为 LEFT JOIN / RIGHT JOIN。
INSERT 缺少字段列表	WARNING	依赖默认列顺序可能导致兼容性问题。	显式指定 INSERT 的字段列表。

表 4-10 命名规范

检查项	级别	说明	优化建议
表名不符合命名规范	WARNING	应使用小写字母、数字和下划线，以字母开头。	按规范重命名表名。
列名不符合命名规范	WARNING	应使用小写字母、数字和下划线，以字母开头。	按规范重命名列名。
使用保留关键字作标识符	ERROR	可能导致语法歧义。	避免使用保留关键字作为标识符。

表 4-11 SQL 开发规范

检查项	级别	说明	优化建议
DROP 建议使用 IF EXISTS	WARNING	防止对象不存在时报错。	添加 IF EXISTS 子句。
列存表禁止实时小批量 INSERT	WARNING	小 CU 膨胀影响性能。	列存表使用批量 INSERT 或 COPY。
列存表避免频繁 UPDATE/DELETE	WARNING	CU 膨胀及死锁风险。	减少列存表的 UPDATE/DELETE 操作频率。

版权所有 © 华为云计算技术有限公司

检查项	级别	说明	优化建议
禁止执行不下推的 SQL	ERROR	必须下推才能利用分布式计算（需集群连接）。	检查 SQL 是否可下推，改写不支持下推的写法。
过滤条件字段避免函数运算	WARNING	影响统计信息准确性。	将函数运算移至等式右侧或使用函数索引。
行存大表禁止频繁 COUNT	WARNING	全表扫描 I/O 开销大。	使用计数器表或估算方式替代全表 COUNT。
查询建议使用 LIMIT	INFO	避免超大结果集。	添加 LIMIT 子句限制返回行数。
谨慎使用 WITH RECURSIVE	WARNING	确保递归终止条件。	检查递归终止条件，避免无限递归。
访问对象建议带 SCHEMA 前缀	INFO	避免 search_path 导致访问错误对象。	使用 schema.table 格式访问对象。

规则级别说明

表 4-12 规则级别说明

级别	含义	处理建议
ERROR	严重违规，可能导致执行错误或性能严重下降。	必须修复。
WARNING	潜在风险，建议优化。	建议修复。
INFO	提示信息，供参考。	按需处理。

检查报告说明

检查完成后生成检查报告，包含以下部分：

- **检查结果汇总**：检查规则总数、通过数、违规数，按 ERROR/WARNING/INFO 级别统计。
- **违规详情**：每条包含级别、位置（行/列）、描述、修复建议。
- **修正后的 SQL 建议**：针对违规项给出修正后的 SQL 示例及修改说明。

使用约束与注意事项

- **不实际执行 SQL**：本检查模式不使用 EXPLAIN PERFORMANCE，不会实际执行用户 SQL（DML 不会修改数据）。

版权所有 © 华为云计算技术有限公司

- **安全说明**：SQL 文本仅在本地处理，不发送至外部服务，不涉及凭证或认证信息。
- **建议检查流程**：先执行语法检查排除基本错误，再执行规范检查进行深度分析。
- **与 SQL 调优的关系**：检查与调优使用同一 skill；检查不实际执行 SQL（full + verbose），调优进一步执行 EXPLAIN PERFORMANCE 采集实际性能指标并输出优化方案，详见 **DWS SQL调优最佳实践**。

4.3 DWS SQL 调优最佳实践

DWS SQL 调优对 SQL 语句执行统一诊断，结合语法规范、执行计划和对象元数据，定位性能瓶颈并输出优化建议与优化后 SQL，帮助用户提升 SQL 执行性能。

Skill 地址：hwc-dws-sql-tuning（随 hwc-dws-cli 分发，源码位于 dws-cli 仓库 agent_skills 目录）

前提条件

本 skill 依赖本地 `hwc-dws-cli` 与 DWS 集群连接，未安装或未配置时**直接中止**（无降级模式）。调用前自检：

```
hwc-dws-cli --version      # CLI 是否已安装
hwc-dws-cli config show    # 配置文件是否可读
hwc-dws-cli check-con      # 集群连接是否正常
```

先完成配置再调用：

```
hwc-dws-cli config set    # 7步向导：AK/SK → 代理 → Region → 集群 → Endpoint → 连接 → 连通性验证
```

功能概述

DWS SQL 调优支持两种检查模式，所有模式均需集群连接（执行计划与对象元数据来自集群）

表 4-13 SQL 调优检查模式

检查模式	说明	规则覆盖	触发方式
full + perf	用户同意执行 EXPLAIN PERFORMANCE（会实际执行 SQL）	默认118 条（含 11 条执行计划扩展规则）。	诊断调优时回答"允许 EXPLAIN PERFORMANCE"。
full + verbose	用户拒绝执行 EXPLAIN PERFORMANCE	107 条（跳过 11 条扩展规则）。	诊断调优时回答"仅 EXPLAIN VERBOSE"。

默认进入 full 模式并询问是否允许 EXPLAIN PERFORMANCE；用户拒绝则自动切换 full + verbose。

使用方式

在 AI Agent 对话窗口中提供 SQL 文本即可触发调优诊断，支持以下典型用法：

版权所有 © 华为云计算技术有限公司

"SQL调优：SELECT ... FROM t1 JOIN t2 ..."

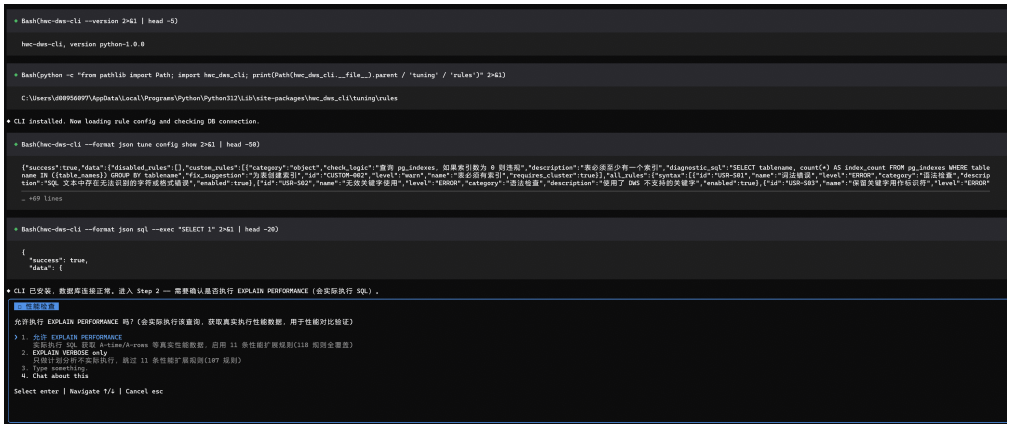
也可通过触发词快速调用：SQL调优、SQL诊断、SQL优化、SQL性能、SQL规范、SQL下推、执行计划、SQL审查、SQL tuning。

```
Debug mode enabled
Logging to: C:\Users\          \.cac\debug\2026-10-12_14-57-45_908e459f-abc8-4172-b1c1-8796ecc6595f.txt

> /dws-sql-tuning 帮我调优sqlSELECT
    t1.ticket_id AS tk_id,
    t1.user_id AS uid,
    t1.order_id AS oid,
    t1.ticket_type AS tk_type,
    CASE t1.ticket_type
        WHEN 'refund' THEN '退款'
        WHEN 'complaint' THEN '投诉'
        WHEN 'inquiry' THEN '咨询'
        WHEN 'exchange' THEN '换货'
        WHEN 'repair' THEN '维修'
        ELSE '其他'
    END AS ticket_type_name,
    CASE t1.priority
        WHEN 'high' THEN 'P0'
        WHEN 'normal' THEN 'P1'
        WHEN 'low' THEN 'P2'
        ELSE 'P3'
    END AS priority_level,
    t1.status AS tk_status,
    CASE
        WHEN t1.status = 'open' AND t1.priority = 'high' THEN '紧急处理'
        WHEN t1.status = 'open' AND t1.priority = 'normal' THEN '正常处理'
        WHEN t1.status = 'open' AND t1.priority = 'low' THEN '低优处理'
        WHEN t1.status = 'closed' THEN '已关闭'
        WHEN t1.status = 'cancelled' THEN '已取消'
        ELSE '未知'
    END AS handle_status,
    t1.create_time AS c_time,
    t1.close_time AS cl_time,
    extract(epoch FROM (coalesce(t1.close_time, now()) - t1.create_time))/3600 AS handle_hours,
    to_char(t1.create_time, 'YYYY-MM-DD') AS create_day,
    to_char(t1.create_time, 'YYYY-MM') AS create_month,
    CASE
        WHEN extract(hour FROM t1.create_time) < 6 THEN '凌晨'
        WHEN extract(hour FROM t1.create_time) < 12 THEN '上午'
        WHEN extract(hour FROM t1.create_time) < 18 THEN '下午'
        ELSE '晚上'
    END AS time_period,
    t2.supplier_id AS sup_id,
    t2.supplier_name AS sup_name,
    t2.supplier_type AS sup_type,
    CASE t2.supplier_type
        WHEN 'manufacturer' THEN '制造商'
        WHEN 'distributor' THEN '分销商'
        WHEN 'wholesaler' THEN '批发商'
        WHEN 'retailer' THEN '零售商'
        ELSE '未知类型'
    END AS supplier_type_name,
    t2.rating AS sup_rating,
    CASE
        WHEN t2.rating >= 4.5 THEN 'A'
        WHEN t2.rating >= 3.5 THEN 'B'
        WHEN t2.rating >= 2.5 THEN 'C'
        WHEN t2.rating >= 1.0 THEN 'D'
        ELSE 'F'
    END AS supplier_grade,
    t3.contact_person,
    t3.contact_phone,
    t3.status AS sup_status,
```

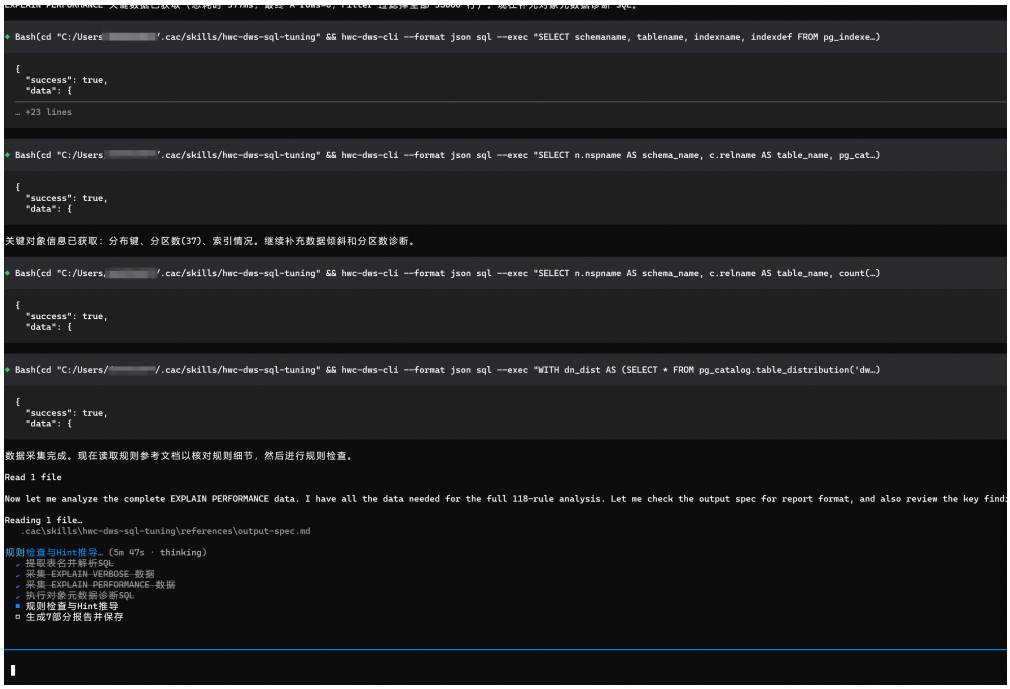
调用后交互流程：

- 环境检测（CLI / 集群连接），失败则中止并提示修复。
- 询问是否允许 EXPLAIN PERFORMANCE（会实际执行 SQL，DML 将修改数据）。



- 自动完成：规则加载 → 表名/模式提取 → 数据收集（EXPLAIN VERBOSE + 15 条诊断 SQL，必要时 EXPLAIN PERFORMANCE）→ 118 条规则检查+用户自定义规则。

版权所有 © 华为云计算技术有限公司



- 输出完整诊断报告（7 部分），并自动保存 sql_tuning_report_YYYYMMDD_HHMMSS.md 到当前工作目录。





版权所有 © 华为云计算技术有限公司



Part 4：优化建议（按优先级排序）

优先级	建议	对应问题	预期收益
P0	修复 AND/OR 优先级：为 OR 分支加括号隔离，或将 OR source_type='online' 并入主 AND 组（业务确认）；推荐语义保持等价写法：WHERE (全部AND条件) OR (source_type='online' AND 日期范围)；若业务上 online 也应受日期限制则直接并入 AND	#1	消除 t1 全表扫描 + 过滤全量下推，最大收益项
P1	改 NOT IN → NOT EXISTS（关联列 x.priority = t1.priority）	#2/#10	SubPlan + Hash Anti Join，消除 Broadcast+Materialize
P1	分区键与过滤列对齐：若 dt 为分区键且语义为业务日期，将过滤条件改为 t1.dt >= '2026-09-01' AND t1.dt < '2026-10-01'（或建索将 create_time 作为分区键）	#3/#4	分区剪枝生效，从 37 分区降到 1 个分区
P1	过滤条件下推：确保日期/类型/状态条件推送到 Scan（需先修复 OR）	#1/#12	t1 Scan 即过滤，Join 输入骤减
P2	精简 IN 列表：删除 extra1=extra101 垃圾值，仅保留真实工单类型	#7	恢复 inlist2join，降低常量比较开销
P2	LIKE 改精确/后缀匹配：t2.supplier_name = 'xxx' 或 LIKE 'tech%'（前缀匹配可用索引）	#8	避免全表扫描
P2	日期直接范围比较：create_time >= '2026-09-01' AND create_time < '2026-10-01' 替代 date_trunc	#9	便于优化器利用统计信息
P3	分布键与 JOIN 键对齐（重置建）：t1 按 order_id，t3 按 contact_person 分布可消除重分布；成本高，量级大前再评估	#6	消除 2 次 REDISTRIBUTE + 10ms 网络等待
P3	收集统计信息：ANALYZE dwd.dwd_aftersale_ticket_detail_di_all; 等 4 张	#11	修正 E-rows 估算，优化器选优计划

Part 5：优化后 SQL + Hint

优化后 SQL（已消除 P0/P1/P2 主要问题）

```
SELECT
    t1.ticket_id AS tk_id,
    t1.user_id   AS uid,
    t1.order_id  AS oid,
    t1.ticket_type  AS tk_type,
    CASE t1.ticket_type
        WHEN 'refund'    THEN '退款' WHEN 'complaint' THEN '投诉'
        WHEN 'inquiry'   THEN '咨询' WHEN 'exchange' THEN '换货'
        WHEN 'repair'    THEN '维修' ELSE '其他'
    END AS ticket_type_name,
    CASE t1.priority
        WHEN 'high'   THEN 'P0' WHEN 'normal' THEN 'P1'
        WHEN 'low'    THEN 'P2' ELSE 'P3'
    END AS priority_level,
    t1.status AS tk_status,
    CASE
        WHEN t1.status = 'open' AND t1.priority = 'high' THEN '紧急处理'
        WHEN t1.status = 'open' AND t1.priority = 'normal' THEN '正常处理'
        WHEN t1.status = 'open' AND t1.priority = 'low' THEN '低优处理'
        WHEN t1.status = 'closed' THEN '已关闭'
        WHEN t1.status = 'cancelled' THEN '已取消'
        ELSE '其他'
    END AS handle_status,
    t1.create_time AS c_time,
    t1.close_time AS cl_time,
    extract(epoch FROM (coalesce(t1.close_time, now()) - t1.create_time))/3600 AS handle_hours,
```

Jump to bottom ↓

Hint 建议

场景	Hint
数据量大时强制 t1 主导 Hash Join	/*+ leading((t1 t2) t3) */
t2 关联 supplier_id 时走索引	/*+ indexscan(t2) */
统计信息缺失时的行数纠正	/*+ rows(t1 #100000) */（按实际量级调整）

Part 6：性能验证（EXPLAIN PERFORMANCE 前后对比）

总体对比

指标	原始 SQL	优化后（去除 OR + NOT EXISTS + 日期范围）	改善
总耗时	252.799 ms	58.114 ms	↓ 4.35×
E-costs	7355.65	2363.92	↓ 68%
SubPlan	有（NOT IN）	无（Hash Anti Join）	消除
过滤下推	Join 后 Filter（全量）	Scan CU Predicate Filter	下推
最大算子耗时	Streaming(10) Network Poll 184–195ms	Sort 34–46ms	↓
Remote query poll	211.667 ms	52.669 ms	↓ 75%

算子级对比

算子	原始 A-time	优化后 A-time
Vector Sort（全局排序）	176–197ms	34–46ms
Join 6（t1–t3 Hash）	185–196ms	34–46ms
Streaming(10) 网络等待	184–195ms（Network Poll）	显著下降
t1 CStore Scan	~43万行（37分区×1.1万）	A-rows=0（过滤下推）
SubPlan 1（NOT IN）	Broadcast+Materialize	已消除

关键变化点

1. 过滤条件从 Join 后 Filter 下推到 Scan 的 CU Predicate Filter：t1 原始扫描 43万行全量进入 Join，优化后扫描即过滤为 0 行
2. NOT IN → Hash Anti Join：计划从 SubPlan 1 + Materialize + Broadcast 变为 Vector Sonic Hash Right Anti Join，消除逐行 probe
3. OR 分支移除后过滤全量生效：日期/ticket_type/status 全部推下 t1 Scan

> ⚠ 注：测试库统计信息缺失（n_live_tup=0）且无真实数据（A-rows=0），绝对耗时仅供参考；生产环境数据量越大，优化收益（尤其过 下推与分区扫描）越显著。

SubPlan | 有（NOT IN） | 无（Hash Anti Join） | 消除
过滤下推 | Join 后 Filter（全量） | Scan CU Predicate Filter | 下推
最大算子耗时 | Streaming(10) Network Poll 184–195ms | Sort 34–46ms | ↓
Remote query poll | 211.667 ms | 52.669 ms | ↓ 75%

算子级对比

算子	原始 A-time	优化后 A-time
Vector Sort（全局排序）	176–197ms	34–46ms
Join 6（t1–t3 Hash）	185–196ms	34–46ms
Streaming(10) 网络等待	184–195ms（Network Poll）	显著下降
t1 CStore Scan	~43万行（37分区×1.1万）	A-rows=0（过滤下推）
SubPlan 1（NOT IN）	Broadcast+Materialize	已消除

关键变化点

1. 过滤条件从 Join 后 Filter 下推到 Scan 的 CU Predicate Filter：t1 原始扫描 43万行全量进入 Join，优化后扫描即过滤为 0 行
2. NOT IN → Hash Anti Join：计划从 SubPlan 1 + Materialize + Broadcast 变为 Vector Sonic Hash Right Anti Join，消除逐行 probe
3. OR 分支移除后过滤全量生效：日期/ticket_type/status 全部推下 t1 Scan

> ⚠ 注：测试库统计信息缺失（n_live_tup=0）且无真实数据（A-rows=0），绝对耗时仅供参考；生产环境数据量越大，优化收益（尤其过 下推与分区扫描）越显著。

Part 7：规则统计

类别	总数	通过	违规	无法评估	+禁用	覆盖率
语法检查 USR	46	40	5	1	0	100%
执行计划 UPR（VERBOSE）	27	22	4	1	0	96%
执行计划 UPR（PERFORMANCE）	11	9	2	0	0	100%
对象检查 UOR	34	32	1	1	0	97%
自定义 Custom	1	1	0	0	0	100%
合计	119	104	12	3	0	97.5%

> 有效规则总数 = (218 − 0 禁用) + (1 − 0 禁用) = 119；禁用 0 (disabled_rules 为空)，3 条无法评估：USR-009（需执行计划判断 JOIN 替代 EXISTS）、UOR-010 全表扫描（FQS 下推场景 CN 计划无 Seq Scan，需结合索引规则判断，已按分区扫描处理）、UOR-34（分布列与 JOIN 键比较已人工分析）。

报告同时已保存至：C:\Users\[redacted]\sql_tuning_report_20261015_0310.md

Worked for 36s

版权所有 © 华为云计算技术有限公司

检查规则与优化建议解读

默认共 118 条规则，按数据依赖分为四类：

表 4-14 检查规则

类别	规则数	数据来源	规则文件
语法检查	46	SQL 文本（静态分析）	syntax_rules.yaml
执行计划检查	27	EXPLAIN VERBOSE	plan_rules.yaml
执行计划扩展检查	11	EXPLAIN PERFORMANCE（需用户确认）	plan_rules_performance.yaml
对象检查	34	16 静态 AST + 17 系统视图 + 1 语义分析	object_rules.yaml

每条规则的完整定义（ID、名称、级别、检查逻辑、修复建议）见规则 YAML 文件，位于 CLI 安装目录：<Python安装目录>\Lib\site-packages\hwc_dws_cli\tuning\rules\

规则配置（可选）

规则由 CLI 统一管理，配置文件位于 ~/.hwc-dws-cli/tuning-config.yaml。skill 在诊断前自动读取该配置：禁用的规则直接跳过，自定义规则参与检查，无需修改 skill 文件即可按需调整诊断范围。

表 4-15 规则配置

操作	命令
查看当前规则配置	hwc-dws-cli tune config show
禁用内置规则	hwc-dws-cli tune config set global.tuning.disabled_rules "UPR-001,UOR-15"
重新启用全部规则	hwc-dws-cli tune config set global.tuning.disabled_rules "[]"
添加自定义规则（单条）	hwc-dws-cli tune config add-rule --id CUS-001 --name "批量UPDATE检查" --category syntax --level error --description "..." --check-logic "..." --fix-suggestion "..."
添加自定义规则（YAML 文件批量）	hwc-dws-cli tune config add-rule --file custom_rules.yaml

版权所有 © 华为云计算技术有限公司

操作	命令
移除自定义规则	hwc-dws-cli tune config remove-rule CUS-001

自定义规则要点：

- --check-logic 为判定逻辑描述，skill 诊断时套用该描述执行检查
- 需要集群连接的规则添加 --requires-cluster，并必须提供 --diagnostic-sql（只读 SELECT，支持 {table_names} / {schema_names} / {sql_text} 占位符），skill 会自动执行该 SQL 采集数据后按 check-logic 判定违规

检查报告说明

诊断完成后输出 7 部分报告：

- 诊断结论：违规总数、最严重问题、修复优先级
- SQL 特征：SQL 类型、涉及表数、诊断模式、规则覆盖、数据来源
- 违规详情：按优先级排序（最多 20 条），含规则 ID、级别、问题、证据、修复建议
- 优化建议：按优先级排序，每条含问题/证据/方案/预期
- 优化后 SQL + Hint：优化前后对比、变更 diff、Hint 建议
- 性能验证：优化前后 EXPLAIN PERFORMANCE / VERBOSE 对比（A-time 提升等）
- 统计：各类别规则总数/通过/违规/跳过

报告同时自动保存为 sql_tuning_report_YYYYMMDD_HHMMSS.md。

使用约束与注意事项

- 必须安装 CLI 并配置集群：所有模式均需集群连接；CLI 未安装、连接失败或 EXPLAIN 执行失败时直接中止（无降级模式）。
- EXPLAIN PERFORMANCE 需用户确认：会实际执行 SQL，DML 将修改数据，用户自行判断风险；拒绝后自动切换仅 EXPLAIN VERBOSE。
- 诊断 SQL 只读：15 条诊断 SQL 均为只读 SELECT，查询系统视图。
- 建议流程：先执行 hwc-dws-cli config set（7步向导）配置集群 → check-con 验证连通 → 再调用 skill；性能检查务必先确认 EXPLAIN PERFORMANCE。

4.4 DWS 知识问答最佳实践

面向 AI Agent 的 **DWS数据仓库服务离线知识库**。内置华为云官方文档 ，支持 agent 离线检索并带官方 URL 出处回答。

纯离线运行：检索阶段零联网、零第三方依赖（Python 3 stdlib 即可）。检索由脚本完成，LLM 只读命中的 chunk 正文。

Skill 地址：hwc-dws-knowledge-qa（随 hwc-dws-cli 分发，源码位于 dws-cli 仓库 agent_skills 目录）

版权所有 © 华为云计算技术有限公司

快速开始

1. 目录结构。

```
dws-knowledge-qa/
├── SKILL.md                # Agent 协议（触发词/检索流程/行为约束）——挂载给 agent 的核心
├── README.md               # 说明
├── references/
│   ├── search-guide.md     # 检索策略手册（LLM 检索时参考）
│   └── index/
│       ├── catalog.json    # 文档域元数据
│       ├── sections.json   # 每域页面清单（标题/官方URL/chunk区间）
│       └── pack/*.jsonl    # 全量 DWS 文档切分的 chunk
└── scripts/
    ├── common.py           # 共享工具
    ├── search.py           # 分层检索入口（LLM 唯一需要的工具）
    ├── fetch_llms.py       # 从华为云 llms.txt 下载官方文档（数据更新用）
    └── build_pack.py       # 构建索引（数据更新用）
```

2. 挂载到你的 Agent。

   挂载后，agent 会通过 SKILL.md 的触发词识别 DWS 问题场景，然后调用 search.py 完成分层检索（--list-docs → --list-sections → --query → --read）。

3. 验证挂载成功。

```
python scripts/search.py --list-docs          # 应列出所有文档域
python scripts/search.py --query "集群扩容" --doc mgtg-dws --topk 5
python scripts/search.py --read mgtg-dws:0001
```

search.py 使用说明

search.py命令说明如表4-16所示。

表 4-16 search.py 命令

命令	说明
--list-docs	列出全部文档域（slug/页数/chunk数）
--list-sections <slug>	某域的页面清单（标题/官方URL/chunk区间）
--query "<关键词>" [--doc <slug>] [--section <子串>] [--topk N] [--json]	分层检索，只输出元数据（不含正文）
--read <doc:chunk-id>...	输出 chunk 全文（唯一放大文本的入口）

示例：

```
python scripts/search.py --query "GDS" --doc migration-dws --topk 5
python scripts/search.py --query "列存表" --doc devg-dws --section "选择DWS表存储模型"
python scripts/search.py --read sqlreference-dws:0001 tg-dws:0007
```

出处格式：答案引用 [页面标题](官方URL)，如 [DWS SSD云盘和SSD本地盘的区别](https://support.huaweicloud.com/dws_faq/dws_03_2141.md)。 联网可单击直达原文；不联网时 URL 作为稳定标识，仍可追溯页面标题。

数据更新

内置库已打包好，**开箱即用**。如需更新到华为云最新文档：

版权所有 © 华为云计算技术有限公司

```
# 1. 下载最新官方文档（需联网，可访问 support.huaweicloud.com）
python scripts/fetch_llms.py --url https://support.huaweicloud.com/dws/llms.txt

# 2. 重建索引
python scripts/build_pack.py
```

增量更新（两种模式）：

```
# 模式A：只下载新增页面，已存在的跳过（不检查内容是否变化）
python scripts/fetch_llms.py --skip-existing --url https://support.huaweicloud.com/dws/llms.txt

# 模式B：已存在的也重新下载，内容有变化才覆盖（推荐用于日常更新）
python scripts/fetch_llms.py --update-changed --url https://support.huaweicloud.com/dws/llms.txt

# 两种模式执行后都要重建索引
python scripts/build_pack.py
```

--update-changed 会输出每页状态：[SAME] 内容未变保留 / [CHANGED] 已更新覆盖 / [OK] 新增。

自定义文档：把自己的 md 放进 sources/<域>/ 后重跑 build_pack.py 即可并入索引。

文档域一览

内置文档域如表4-17所示。

表 4-17 文档域一览

slug（域前缀）	内容
sqlreference-dws / sqlreference-911-dws	SQL 语法参考（9.1.0 / 9.1.1）
devg-dws / devg-911-dws	开发指南（9.1.0 / 9.1.1）
mgtg-dws	用户指南（集群管理/控制台操作）
api-dws	API 参考
dws_faq	常见问题
trouble-dws	故障排除
bestpractice-dws	最佳实践
migration-dws	数据迁移与同步
tg-dws	工具指南（gsql/GDS/DSC/DataCheck）
errorcode-dws / errorcode-new-dws	错误码（8.2.0 及以下 / 8.2.1 及以上）
productdesc-dws	产品介绍
price-dws	计费说明
bulletin-dws	服务公告/版本说明
qs-dws	快速入门
pwp-dws / pwp-910-dws / pwp-911-dws	性能白皮书

版权所有 © 华为云计算技术有限公司

slug（域前缀）	内容
sdkreference-dws / clir-dws	SDK / CLI 参考

常见问题

- **Q: 需要联网吗?**

  检索阶段不需要——索引和 chunk 全部离线内置。只有「 数据更新 」 需要联网拉取华为云文档。

- **Q: 需要安装第三方库吗?**

  不需要。search.py / build_pack.py / fetch_llms.py 全部 使用 Python 3 标准库（ urllib/argparse/json/re/pathlib ）。

- **Q: 支持哪些 Python 版本?**

  Python 3.8+ 均可（ 含 3.12 ）。

版权所有 © 华为云计算技术有限公司

5 Data for AI 融合分析

5.1 Data for AI 能力介绍

背景信息

在企业级AI时代，随着推理模型和垂域模型能力的突破，AI正从开发者工具演变为企业核心生产力。然而，传统数据仓库面临三重挑战：多引擎架构带来的数据冗余与运维负担、AI与数据平台割裂导致的开发效率低下、非结构化数据的价值难以充分释放。企业不再仅仅需要“更快的查询引擎”，而是需要能够理解、推理并创造价值的“智能数据伙伴”。

如何解决这些挑战，实现数据价值的最大化？华为云DWS **9.1.1.200**版本，提供**Data for AI融合分析能力**，集成MCP协议，支持一站式OLAP分析、点查、全文检索、库内推理和AI推理等融合分析能力，重塑企业智能应用开发新范式，便于数据工程师更便捷地使用AI，从而实现数据价值最大化，为企业客户提供实时、智能、高效的决策支持。

为什么企业需要 AI 原生数据仓库

在企业级数据分析场景中，传统融合分析架构依赖多个独立引擎协同工作，如Elasticsearch处理全文检索、Milvus负责向量检索、Doris专注OLAP分析、ClickHouse承载点查场景。然而该架构存在显著问题：

- **存储冗余**：数据在各引擎间重复存储，导致资源利用率低下。
- **运维复杂**：多系统独立维护，故障定位与性能调优困难。
- **开发低效**：数据格式转换与链路适配消耗大量研发资源。
- **实时性不足**：批量数据流转难以满足AI应用毫秒级响应需求。

如何解决这些问题，提高数据分析的效率和实时性？DWS提供AI原生的一站式分析引擎，极大地简化了原有多引擎协同分析架构。该引擎集成MCP协议，打通与大模型平台的交互通道；集成向量检索能力，内置AI Function支持大模型调用，实现端到端的库内推理，从而构筑AI-Ready数据分析基础设施，消除数据孤岛。同时，通过基于Binlog的物化视图实现流批一体增量计算能力，面向开发者提供声明式Pipeline加工新范式，实现近实时分析。整体架构更加轻量化，显著提高开发效率，降低运维管理难度，满足企业AI平台部署一站式多模态融合分析的核心诉求。

版权所有 © 华为云计算技术有限公司

Data for AI 四大核心能力

Data for AI包括以下核心能力：

MCP 协议原生集成：打通 AI 生态的“神经网络”

MCP（Model Context Protocol，模型上下文协议）是由 Anthropic 推出的一种开放标准，即“AI的USB-C”，旨在统一大模型与外部数据源、工具之间的通信协议，从而解决当前AI模型因数据孤岛限制而无法充分发挥潜力的难题，MCP 使得 AI 应用能够安全地访问和操作本地及远程数据，为AI应用提供了连接万物的接口。

DWS支持对接MCP协议，一键配置，即插即用。通过标准化协议接口，企业可无缝接入Claude、Cursor等主流AI平台，内置8个常用的数据分析与运维监控类型API，真正实现“万物互联，开箱即用”。

AI Function：可插拔式的库内 AI 推理

DWS集成开源社区插件pgai，支持对接外部开源大模型、华为云MaaS服务等，通过库内SQL方式调用大模型进行推理，提供文本分析（ai.classify）、向量化（ai.embed）、文本摘要（ai.summarize）以及情感分析（ai.sentiment）等24个AI Function，将大模型推理能力下沉至数据库内核，从而为企业客户提供更加智能化的数据分析服务，实现数据即推理、即处理的全新操作模式，助力企业快速获取洞察，提升决策质量和效率。

向量计算：非结构化分析的“水电煤”

企业80%的数据为文档、图像、音视频等非结构化数据。这类数据过去长期处于数据挖掘盲区，数据价值一直被低估；依托多模态统一向量存储与表征技术，多模态检索不再存在技术门槛。

DWS深度集成开源社区插件pgvector，实现分布式架构的改造，支持向量数据类型，支持IVFFlat、HNSW索引结构，支持相似度计算、最近邻搜索等多种高阶算法，实现基本的向量检索能力。

内置大模型特征算子：赋予用户 AI 分析“无限创造力”

通过PL/Python扩展支持，内置32个自研大模型算子，将Python数据科学生态完整引入SQL工作流。开发者可直接在库内运行复杂AI算法，消除跨系统数据流转开销，实现从特征工程到模型推理的端到端闭环。

DWS AI 数仓使用场景

场景 1. 零售行业：实时运营、智能选品及精准营销三位一体，提升消费者体验

- **当前问题**
  - 传统数据仓库仅能存储交易订单、会员信息等结构化数据，难以对用户浏览日志、商品评价、直播弹幕等非结构化数据进行分析。
  - 离线分析无法支持实时促销决策。例如，直播间中某商品突然热销时，无法迅速调整库存和定价。
  - 用户画像主要依赖规则化标签，难以通过语义推理从商品评价、直播弹幕等来源挖掘用户的真实痛点或需求。
- **业务价值创造**

版权所有 © 华为云计算技术有限公司

  - 在线报表：支持年、月、周、日度报表的高效生成，统计商品类目信息及销量情况等，助力客户日常经营决策。
  - 实时运营：对接直播销售渠道，实时采集商品评价中负向情感占比超过30%的数据，自动触发预警机制，运营人员及时调整话术或发放优惠券。
  - 智能选品：结合“销量数据+评价情感+相似商品热度”，精准筛选高潜商品，为采购决策提供科学指导。
  - 精准营销：基于用户画像，向具有高复购特征且评价中提及“送礼”的用户，推送礼盒装商品优惠券。

场景 2. 制造行业：以设备-工艺-质量联动，助力实时决策与品质提升

- **当前问题**
  - 设备IoT数据、工艺参数表、质检报告分别存储于不同系统（IoT平台、ERP、MES），导致跨系统分析时存在冗余数据同步问题。
  - 质量缺陷分析依赖人工经验，难以快速定位“设备参数异常-工艺偏差-缺陷产生”等关联关系。
  - 离线分析无法支撑实时工艺调整。例如，某台设备振动值超标时，无法即时预警以避免批量次品的产生。
- **业务价值创造**：
  - 流批一体：IoT设备数据实时入库，并定期执行批量增删改操作。离线计算历史工艺参数与缺陷率的关联规则，实时监控IoT数据，当参数偏离最优区间时触发预警。
  - 质量溯源：输入某批次次品编号，一键查询“生产设备IoT数据+工艺参数+质检记录”，快速定位缺陷根因。
  - 预测性维护：基于设备传感器数据和历史故障记录，预测设备故障概率，提前安排维保，减少停机时间。
  - 工艺优化：通过SQL统计分析不同工艺参数组合的合格率，利用回归算法建模“温度-压力-合格率”的关系，输出最优工艺参数区间，将产品合格率提升5%-10%。

场景 3. 金融行业：交易-行为-舆情联动，构筑金融风控实时决策新防线

- **当前问题**
  - 传统风控仅依赖交易数据（如金额、时间、商户），难以识别异常行为与舆情风险。例如，即使账户被盗，交易金额可能保持正常，但登录设备出现异常。
  - 规则引擎也难以应对新型欺诈手段，如“羊毛党”通过批量注册账号、模拟正常行为进行薅羊毛。
  - 舆情风险（例如某企业被曝财务造假）无法实时同步至风控体系，导致信贷决策滞后。
- **业务价值创造**
  - 实时风控：交易发生时，实时完成多维度风险评估（包括账户异常、设备异常等），拦截盗刷、羊毛党等欺诈行为，降低损失率。
  - 实时反欺诈：实时筛选分析“单笔交易大于5万+月交易次数骤增10倍”的用户，通过向量检索匹配用户行为与欺诈样本的相似度。
  - 信贷智能决策：向企业发放贷款前，结合“财务数据+舆情风险+行业趋势”等信息，输出信贷额度和利率建议。

版权所有 © 华为云计算技术有限公司

- 舆情风险预警：当企业被曝出负面新闻时，自动标记相关信贷客户，启动风险排查流程。

场景 4. 政务行业：智慧城市一体化建设，驱动城市智能化管理升级

- **当前问题**
  - 政务数据分散在各部门，数据格式不统一，难以支撑跨部门协同分析。
  - 民生反馈（如市长热线、政务APP留言等）多为非结构化文本，无法量化分析居民诉求。
  - 城市治理依赖人工巡检，无法实时响应突发问题。
- **业务价值创造**：
  - 智能交通调度：结合时空和实时车流数据，动态生成交通拥堵热力图，智能推荐疏导线路，在早晚高峰时段，系统实时调整红绿灯配时，缓解交通拥堵。
  - 民生诉求精准响应：根据居民诉求的语义分类和区域分布，优先调配资源解决高频问题。
  - 突发公共事件预警：当监测到积水超标+强降雨预警，自动触发防汛响应，输出防汛预案建议，辅助相关职能部门应急决策。

5.2 基于 DWS MCP Server 搭建数据分析 Agent

在企业数据分析和运营决策过程中，业务人员常需从数据仓库提取数据并生成分析报告。传统模式下，业务人员需将数据需求提交给开发人员，由开发人员编写SQL语句查询数据，再将结果反馈给业务人员。这一流程通常耗时数天，频繁的沟通迭代进一步拖慢了决策效率。

随着大语言模型（LLM）技术的成熟，是否存在一种方式能让业务人员直接使用自然语言与数据库交互，无需编写SQL即可完成数据查询与分析?

DWS MCP Server正是为解决这一需求而生的方案。它基于**MCP（Model Context Protocol）协议**，将DWS数据库的元数据查询、SQL执行等能力以标准化工具接口暴露给大模型，使LLM能够自动理解数据库结构、生成SQL语句并执行查询，最终将查询结果进行业务解读与可视化呈现。

通过本实践，您可以快速搭建一个以**DWS + MCP + LLM**为核心的数据分析Agent，实现“**自然语言提问 > 自动生成SQL > 查询执行 > 分析报告**”的一键式数据服务。

基本概念

在开始配置之前，了解以下核心概念有助于您更好地理解本文档的操作背景。

- **MCP（Model Context Protocol）**：由Anthropic于2024年11月提出的开放协议标准，旨在解决大语言模型与外部系统（如数据库、API）交互的碎片化问题。MCP通过标准化接口，让LLM动态理解工具功能并执行操作，降低集成成本。类比理解：MCP之于AI应用，相当于USB之于外设——提供统一的标准接口，让不同系统可以即插即用。
- **MCP Server**：实现了MCP协议的服务端程序，负责将特定系统（如数据库）的能力以Tools和Resources的形式暴露给支持MCP协议的客户端。本文中的DWS MCP Server即是一个将DWS数据库能力暴露给LLM的服务端。

版权所有 © 华为云计算技术有限公司

- **MCP Client**：支持MCP协议的客户端应用，负责与MCP Server通信并向LLM转发工具调用请求。常见的MCP客户端包括Cline（VS Code插件）、Claude Desktop等。
- **Agent（智能体）**：基于LLM构建的自动化系统，能够根据用户输入的自然语言指令，自主规划任务步骤、调用外部工具并完成复杂工作流。本文搭建的数据分析Agent即为一个能够自动查询数据库并生成分析报告的智能体。
- **uv**：一个高性能的Python包管理器和运行器，用Rust编写，可快速安装和运行Python项目。在本实践中用于启动DWS MCP Server。可通过pip install uv安装。
- **Psycopg2**：Python语言中最流行的PostgreSQL数据库适配器，DWS基于PostgreSQL内核，因此使用Psycopg2作为DWS MCP Server与数据库之间的连接驱动。

功能介绍

DWS MCP Server目前支持元数据查询、语句执行、监控信息查询等基本功能，以MCP协议中的工具（Tools）和资源（Resource）两种形式向支持MCP协议的客户端暴露。相关工具和资源参见表5-1和表5-2。

表 5-1 数据库管理工具（Tools）

名称	描述
list_databases	列出所有数据库。
get_activity	从 pgxc_stat_activity视图获取最近的查询活动。
execute_query	执行 SQL 查询。
list_schemas	列出当前数据库中的所有模式。
list_tables	列出指定模式下的所有表。
list_views	列出指定模式下的所有视图。
get_table_info	获取表/视图的定义。
get_comment	获取模式/表的注释。

表 5-2 可用资源（Resources）

资源URI	功能说明
gaussdb:///{schema}/tables	列出指定模式下的所有表
gaussdb:///{schema}/views	列出指定模式下的所有视图
gaussdb:///{schema}/{table}/attributes	列出指定表/视图所有的列
system:///{system_path}	系统信息（例如 /version）

版权所有 © 华为云计算技术有限公司

前提条件

- 已创建DWS集群，且集群状态为”可用”。如需创建集群，请参见**创建DWS存算分离集群**。
- 已获取DWS集群的连接信息（主机IP、端口号、数据库名、用户名及密码）。参见**获取DWS集群连接地址**。
- 客户端所在环境与DWS集群之间网络连通。如果集群为内网访问，需确保客户端与集群在同一VPC下；如需公网访问，请先绑定弹性公网IP。
- Python版本为3.10及以上。可通过命令python --version或python3 --version查看当前版本。
- uv版本为0.6.7及以上。可通过命令uv --version查看当前版本；如未安装，执行pip install uv安装。
- 已安装VS Code编辑器并安装Cline插件。如使用其他MCP客户端（如Claude Desktop），需确保客户端支持MCP协议的stdio传输方式。

步骤一：搭建 Agent 和配置 Server

以下使用Cline作为客户端演示如何配置使用DWS MCP Server。您也可根据需求选择其他支持MCP协议的客户端（如Claude Desktop等），配置方式类似。

步骤1 从GitHub下载DWS MCP Server源码到本地环境。

1. 打开VS Code，按下Ctrl + `（反引号，即键盘左上角Esc下方的按键）打开终端面板。也可通过菜单栏**终端** → **新建终端**打开。
2. 在终端中执行以下命令下载或更新uv。

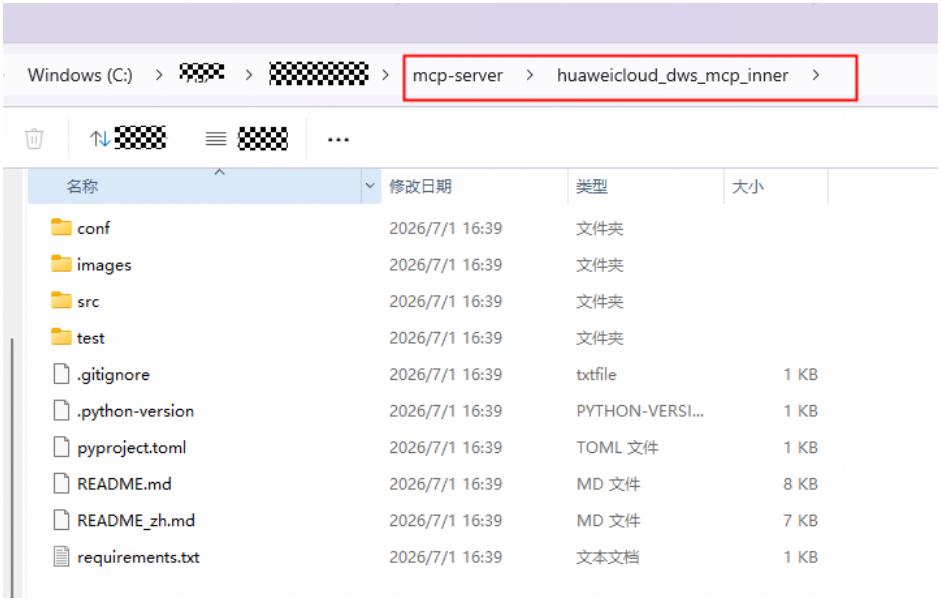
   如果已安装uv可跳过此步骤。

   ```
   pip install uv
   ```

3. 在终端中执行以下命令克隆源码仓库：

   ```
   git clone https://github.com/HuaweiCloudDeveloper/mcp-server.git
   ```

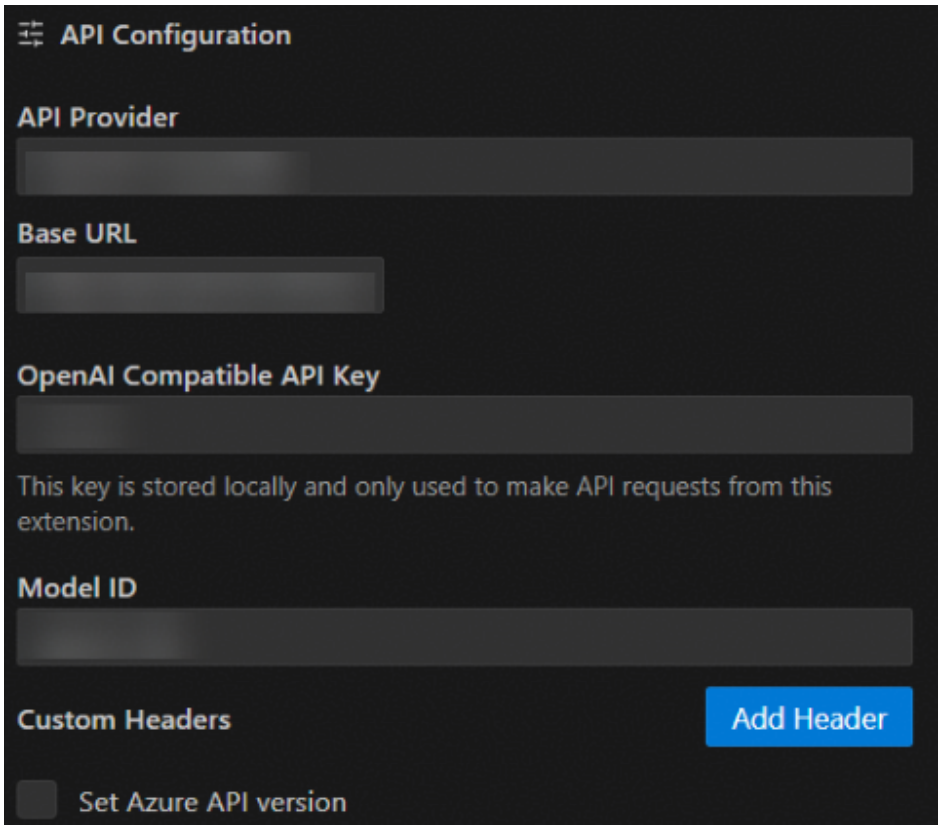
   注意源码默认下载到**huaweicloud_dws_mcp_inner**目录下，请在后续Cline配置步骤中进行设置。例如，若在/home/user/下执行clone，则该目录路径为/home/user/mcp-server/huaweicloud_dws_mcp_inner。



版权所有 © 华为云计算技术有限公司

步骤2 为Cline配置大模型访问凭证，使其能够调用LLM进行自然语言推理与工具调用。

1. 单击Cline对话面板右上角的**齿轮图标**（设置按钮），进入Cline设置界面。
2. 在设置界面的**API Configuration**区域，根据自身使用情况填入以下信息：
   - **API Provider**：选择您使用的大模型服务商（如OpenAI、Anthropic等）。
   - **OpenAI Compatible API Key**：填入对应的API密钥。
3. 填写完成后，设置会自动保存。



步骤3 **配置MCP Server**。将DWS MCP Server注册到Cline客户端，使其能够发现并调用DWS数据库工具。

1. 单击Cline对话面板右上角的MCP图标，进入MCP配置界面。
2. 单击**Installed**页签，在下方单击**Configure MCP Servers**按钮，此时会打开一个JSON配置文件。
3. 在配置文件中填入以下DWS MCP Server配置，各参数说明参见**表5-3**。

```
{
  "mcpServers": {
    "DWS": {
      "disabled": false,
      "timeout": 60,
      "type": "stdio",
      "command": "uv",
      "args": [
        "--directory",
        "/path/to/huaweicloud_dws_mcp_inner",
        "run",
        "server.py"
      ],
      "env": {
        "DB_HOST": "192.168.0.1",
        "DB_PORT": "8000",
```


版权所有 © 华为云计算技术有限公司

```
      "DB_NAME": "gaussdb",
      "DB_USER": "dbadmin",
      "DB_PWD": "password"
    }
   }
  }
}
```

表 5-3 Cline 参数说明

参数名称	描述	示例
/path/to/huaweicloud_dws_mcp_inner	MCP Server源码所在目录。	替换为huaweicloud_dws_mcp_inner目录所在路径。
host_ip	DWS集群IP地址。	192.168.0.1
port_no	DWS集群端口号。	8000
database	DWS数据库名称。	gaussdb
username	DWS数据库用户名称。	dbadmin
password	DWS数据库用户密码。	-

步骤4 如果因网络问题无法使用uv，可以**通过python启动server**，步骤如下。

1. 在源码目录下通过pip安装dws-mcp-server。
   ```
   pip install .
   ```
2. 将Cline的MCP Server配置更换为如下参数。注意，/path/to/huaweicloud_dws_mcp_inner/src/server.py 替换为 DWS MCP Server 源码中server.py的完整路径。
   ```
   {
    "mcpServers": {
     "DWS": {
       "disabled": false,
       "timeout": 60,
       "type": "stdio",
       "command": "python",
       "args": [
         "/path/to/huaweicloud_dws_mcp_inner/src/server.py",
       ],
       "env": {
         "DB_HOST": "host_ip",
         "DB_PORT": "port_no",
         "DB_NAME": "database",
         "DB_USER": "username",
         "DB_PWD": "password"
       }
     }
    }
   }
   ```
   保存配置信息后，观察Cline MCP页面是否成功加载DWS MCP Server，如能够加载DWS Server并显示对应工具及资源，说明配置成功。

版权所有 © 华为云计算技术有限公司



步骤5 DWS MCP Server通过 Psycopg2 连接集群，需要放通DWS安全组网络。

1. 进入**DWS控制台**，左侧导航栏单击“集群 > 集群列表”，单击集群名称进入DWS集群详情。
2. 单击安全组名称。

图 5-1 DWS 安全组



3. 选择“入方向规则 ”，检查是否已放通端口为8000的所有IP。如下图，如果没有则手动添加一条规则。



----结束

步骤二：使用自然语言完成 SQL 查询并生成报表

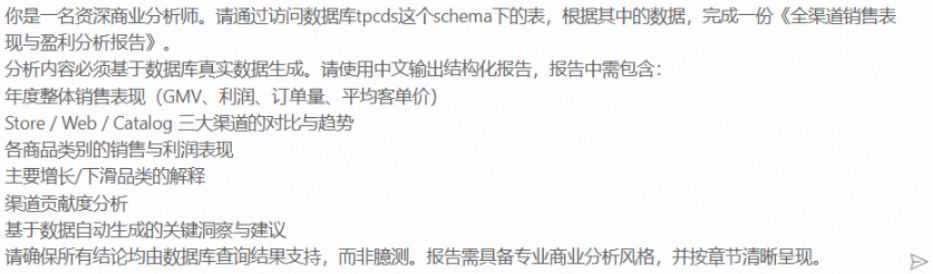
完成客户端及集群相关配置后，即可开始使用数据分析Agent。以下以Cline为客户端，基于TPC-DS数据集进行演示。

演示场景：根据TPC-DS相关表的数据，对1998~2002年间的销售情况进行分析并提出洞察与建议，最终交付数据分析报告。

DWS MCP Server为LLM提供了准确的数据来源，依赖模型的推理和分析能力，您可以在无需手动编写SQL查询语句的情况下获取数据，用自然语言一键式完成查询，并进一步借助LLM对数据进行初步的分析和洞察。

步骤1 在Cline的对话框中输入需要完成的数据分析任务的Prompt。例如：

“请根据数据库中的TPC-DS数据，分析1998年至2002年间的销售趋势，包括年度销售额变化、热销品类排名、季节性波动等，并给出业务建议。”

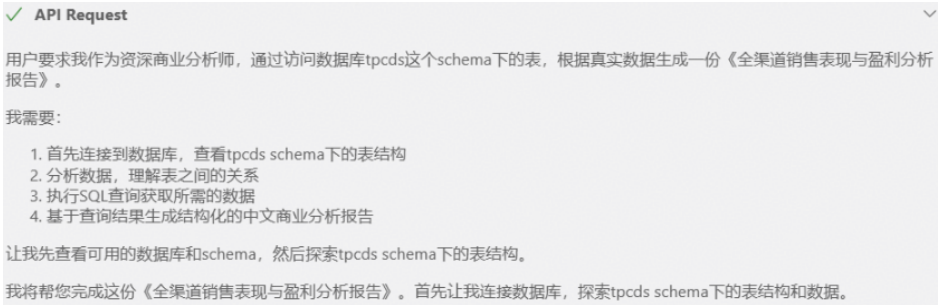


 版权所有 © 华为云计算技术有限公司

步骤2 发送任务后，Cline会调用模型并依据推理结果发起一系列tools或resource的调用请求(如下显示)。

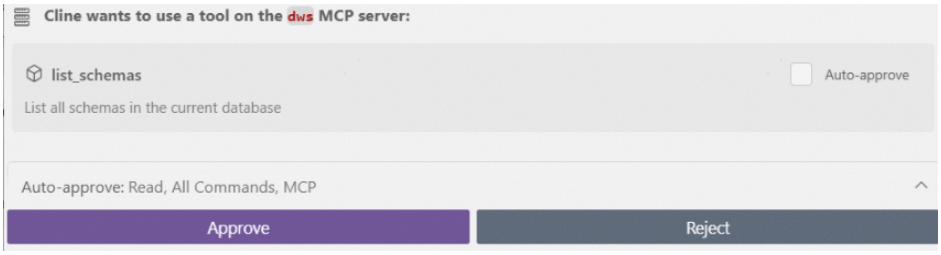
大部分无需干预，您只需要观察请求主体并**选择同意**（可以按需开启自动同意执行）。

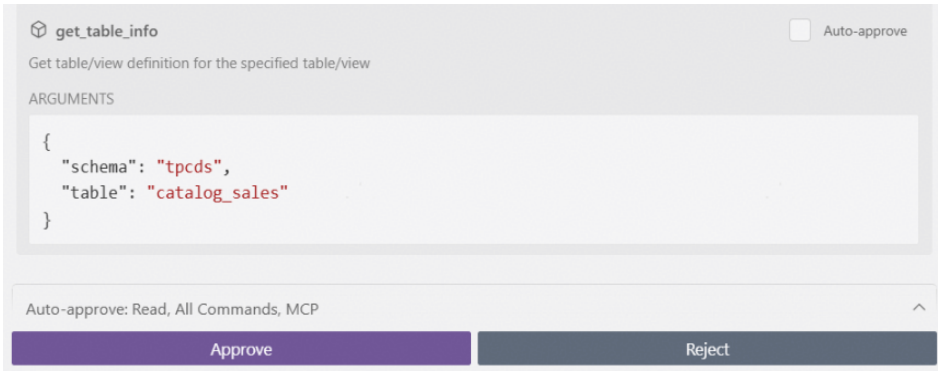
- **解析任务并生成执行计划**。模型理解用户意图，拆解分析任务为多个子步骤。





- **调用工具获取元数据**。模型自动调用list_tables、get_table_info等工具（参见**表5-1**），了解数据库中有哪些表以及表结构。





- **生成SQL并执行查询**。根据元数据信息，模型推理并生成查询SQL，调用execute_query工具获取具体数据。

版权所有 © 华为云计算技术有限公司



- **生成分析报告**。模型将查询结果整理为结构化的分析报告，包含数据表格和业务解读。



版权所有 © 华为云计算技术有限公司



- **输出总结**。模型给出关键洞察和业务建议。



----结束

总结

通过搭建DWS MCP Server，企业与数据分析团队可以将自然语言对话与DWS数据仓库无缝衔接，实现”一键查询、自动化报表、动态分析”的全链路数据服务。按本实践步骤完成配置后，LLM将直接识别MCP工具接口（如list_databases、execute_query等），在安全约束内调用接口进行SQL生成与执行。元数据查询（schema、tables、views）与业务查询结果无缝串联，随后模型可立即进行业务解读、可视化图表与洞察报告的自动生成。该方案不仅显著降低了传统SQL开发的门槛和运维成本，还提升了数据迭代的响应速度，为数据驱动决策提供了高度可靠且易扩展的技术平台。

5.3 基于 DWS 构建 RAG 框架生成行业调研报告

在企业战略研判、行业竞品分析、政策研究等场景中，调研人员通常需要从大量行业报告、新闻稿、研究论文等长文本语料中提取关键信息，整合生成结构化的调研报告。然而，传统的人工撰写方式面临以下困难：

- **数据规模庞大**：一份调研报告往往需要阅读数十篇甚至上百篇长文档，人工筛选与整理耗时耗力。
- **信息碎片化**：关键信息散落在不同文档的不同章节中，难以快速定位和关联。
- **效率与一致性难以兼顾**：不同人员撰写的报告风格和深度差异大，质量参差不齐。

版权所有 © 华为云计算技术有限公司

如何从海量语料中高效、准确地提取相关内容并自动生成高质量调研报告，成为企业亟待解决的问题。

DWS集成了**pgai**和**pgvector**插件，支持在数据库内直接调用**大语言模型（LLM）和Embedding模型**，并提供向量存储与近似检索能力。基于DWS构建**RAG（Retrieval-Augmented Generation，检索增强生成）框架**，可以实现”语料入库 > 智能分块 > 向量检索 > LLM生成”的端到端自动化流程，从而在保证内容可溯源的同时，显著提升调研报告的生成效率与质量。

基本概念

在阅读本文档前，建议了解以下核心概念：

- **RAG（Retrieval-Augmented Generation，检索增强生成）：**

  一种将信息检索与大语言模型生成相结合的技术范式。核心思想是在LLM生成文本前，先从外部知识库中检索与任务相关的信息，将其作为上下文注入生成过程，从而缓解LLM幻觉和知识滞后问题。

- **LLM（Large Language Model，大语言模型）**：

  基于深度学习的大规模预训练语言模型，能够根据输入的提示文本生成连贯的自然语言内容。本方案中用于根据检索到的文本块生成调研报告。

- **向量**

  将文本、图像等非结构化数据通过数学方式映射为固定维度的数值数组（如[0.12, 0.34, -0.21, ...]），使计算机能够通过数值计算来衡量数据之间的语义相似度。例如描述一个人的年龄、身高、体重，可以用[30, 175, 70]这个三维向量表示，让计算机可以理解其语义特征。

- **Embedding（向量化/嵌入）**：

  将非结构化数据（如文本）转换为向量的过程，通常借助大语言模型（如BERT、GPT等）完成。Embedding后的向量能够保留原始数据的语义信息，语义相近的文本在向量空间中距离也相近，使得通过向量相似度比较即可实现语义级别的文本检索。

- **Chunk（文本分块）**：

  将长文本按照一定粒度切分为语义完整的短文本片段的过程。分块后的文本便于向量化处理和精确检索，避免长文本整体向量导致的语义模糊。

- **ANN（Approximate Nearest Neighbor，近似最近邻）**

  一种在高维向量空间中快速查找与目标向量最相似的Top-K向量的算法。相比精确搜索，ANN以极小的精度损失换取数量级的性能提升，适合大规模向量检索场景。

- **pgai**：DWS集成的AI插件，支持在数据库内调用LLM和Embedding模型，并提供文本分块（chunk_text_recursively）、文本排序（rank）、文本生成（openai_chat_complete）、向量化（openai_embed）等AI Functions。
- **pgvector**：DWS集成的向量存储与检索插件，提供vector数据类型和基于HNSW算法的向量索引，支持高效的近似最近邻搜索(ANN)。

约束与限制

- DWS版本需为9.1.1.200及以上。如需使用，请联系技术支持设置GUC参数**feature_support_options**为**enable_pgvector**、**enable_pgai_extension**。
- 如果是存算分离集群，pgvector扩展功能（CREATE EXTENSION语句）仅支持在主VW上执行。

版权所有 © 华为云计算技术有限公司

前提条件

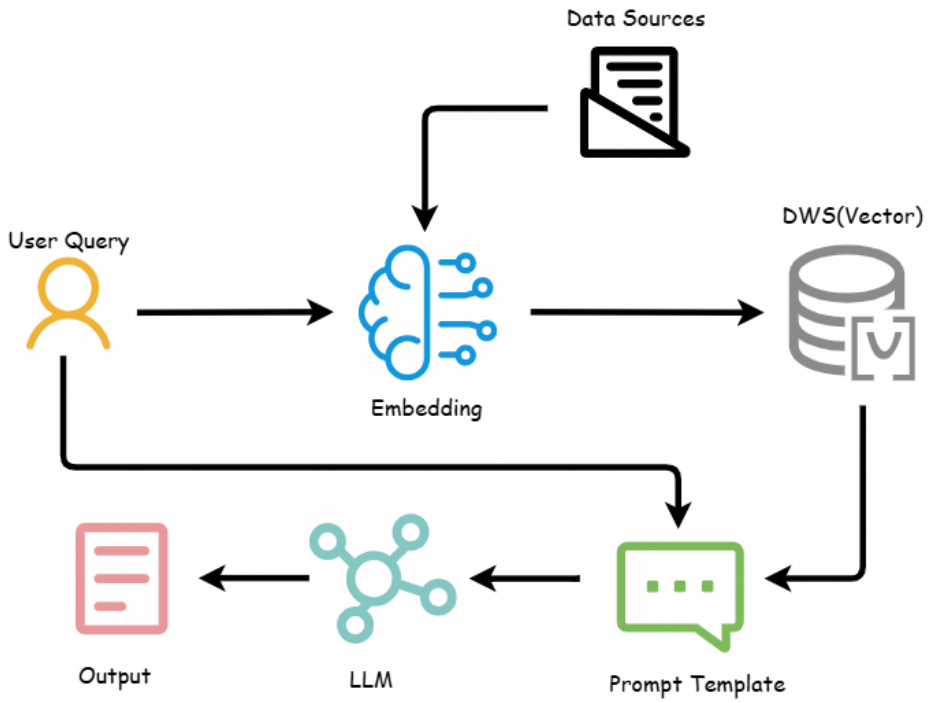
- 已创建DWS集群，且集群状态为”可用”。如需创建集群，请参见**创建DWS集群**。
- **使用本功能前，请先联系技术支持安装pgai插件。**
- **模型API**：需提前准备好可访问的大模型API服务（兼容OpenAI接口规范），并获取有效的Base URL和API Key。API服务需提供以下模型能力：
  - Embedding模型：用于文本向量化（如text-embedding-ada-002等）。
  - Chat模型：用于文本生成与排序（如GPT-4、GPT-3.5-turbo等）。
- **网络连通**：DWS集群需能访问模型API服务地址。若API服务部署在公网，需确保集群已配置公网访问能力；若使用华为云ModelArts等服务，需确保VPC网络互通。

基于 DWS 构建的 RAG 架构

Naive RAG

Naive RAG框架的核心思想是：在大语言模型（LLM）生成文本前，先通过检索模块从外部知识库中获取与任务高度相关的信息，并将其作为上下文注入生成过程。这一机制有效缓解了LLM幻觉、知识滞后和领域适配性弱等问题，特别适用于需依赖特定领域知识的复杂任务，如专业问答、长文档摘要、行业调研报告生成等。

- **检索阶段（Retrieval）**：根据用户输入的问题或提示，从预构建的知识库中检索出若干相关文本片段（如段落或文档块）。
- **生成阶段（Generation）**：将检索结果与原始查询组合为结构化提示（Prompt），交由LLM生成最终输出，确保回答内容有据可依、逻辑严谨。



在系统初始化阶段，原始文本数据源（Data Sources）会被预处理并切分为语义完整的文本块（chunks）；每个chunk经Embedding模型转换为高维向量后，持久化存储于DWS中（借助pgvector的向量存储和HNSW索引能力）。

版权所有 © 华为云计算技术有限公司

当接收到用户查询时，系统同步将查询文本编码为向量，并在DWS中执行近似最近邻搜索（ANN），召回语义最相关的Top-K文本块。这些检索结果与用户原始查询共同构成增强型Prompt，输入LLM以生成准确、可溯源的响应。

步骤一：创建扩展与存储表

创建pgai和pgvector扩展，并建立用于存储长文本语料、分块文本向量及调研报告结果的数据表，为后续RAG流程提供数据存储基础。

步骤1 确认GUC参数feature_support_options已打开**enable_pgvector**和***enable_pgai_extension***。请联系技术支持进行设置。

步骤2 **联系技术支持部署pgai插件**。

步骤3 连接DWS数据库，创建所需扩展。

```
CREATE EXTENSION ai;
CREATE EXTENSION pgvector;
```

如提示以下类似报错，均是GUC参数未开通，请联系技术支持设置。

```
gaussdb=> CREATE EXTENSION ai;
ERROR:  pgai is not yet supported, please add enable_pgai_extension into feature_support_options.
gaussdb=> CREATE EXTENSION pgvector;
ERROR:  pgvector is not yet supported, please add enable_pgvector into feature_support_options.
gaussdb=>
```

如开通GUC参数后，提示以下错误，则是未安装pgai插件，请联系技术支持安装。

```
gaussdb=> CREATE EXTENSION ai;
ERROR:  cn_5001: could not open extension control file: No such file or directory
```

步骤4 创建长文本语料表。documents表中每一行代表一个长文本语料：id作为主键区分行，topic记录长文本语料的主题，content记录长文本语料的所有内容。

```
CREATE TABLE documents(
    id SERIAL PRIMARY KEY,
    topic text,
    content text
);
```

步骤5 创建长文本向量化表。chunk_text表记录所有长文本分块后的内容：chunk记录分块后的文本内容，embedding记录分块后文本的向量表示。hnsw索引用于加速向量近似检索。

```
CREATE TABLE chunk_text(
    id SERIAL PRIMARY KEY,
    chunk text,
    embedding vector
);
CREATE INDEX ON chunk_text USING hnsw(embedding vector_cosine_ops);
```

步骤6 创建调研报告结果表。reports表每一行代表一篇调研报告，content中存储一篇调研报告的文本内容。

```
CREATE TABLE reports(
    id SERIAL PRIMARY KEY,
    content text
);
```

----结束

版权所有 © 华为云计算技术有限公司

步骤二：配置模型 API

为pgai函数配置模型API服务的连接信息和模型名称，使DWS能够调用外部大模型服务。

步骤1 请先准备好可调用的模型，例如华为云MaaS，参见相关**产品文档**进行购买。

步骤2 设置API服务的Base URL和API Key。

将baseurl替换为模型API服务的地址（如https://api.openai.com/v1），将apikey替换为有效的API Key。API Key将加密存储在DWS中。

```
SELECT ai.dws_pgai_encrypt_info('baseurl', 'apikey');
```

步骤3 设置各AI函数所使用的模型名称。

```
-- 设置Embedding模型（用于文本向量化）
SELECT ai.set_func_model('openai_embed', 'text-embedding-ada-002');
-- 设置Chat模型（用于文本生成）
SELECT ai.set_func_model('openai_chat_complete', 'gpt-4');
-- 设置排序模型（用于文档粗筛）
SELECT ai.set_func_model('rank', 'gpt-4');
```

----结束

步骤三：导入实践测试数据

步骤1 **导入长文本语料**。将原始的长文本语料导入语料表documents，作为RAG检索的知识库。根据实际语料来源，可通过INSERT语句逐条导入，或使用DWS的批量导入工具（如\copy命令、GDS导入等）从文件中批量加载。

```
INSERT INTO documents(topic, content) VALUES ('数据库市场需求', '随着数字化转型的加速，全球数据库市场需求不断攀升。企业在数据驱动决策、提升效率和创新方面的需求推动了数据库技术的广泛应用。从传统的关系型数据库（RDBMS）到新兴的非关系型数据库（NoSQL），不同类型的数据库在各种行业中得到了应用，尤其是在金融、电商、医疗和物联网领域。尤其是大数据分析的普及，使得企业需要处理海量数据，并快速从中提取有价值的信息，这推动了对分布式数据库和高性能数据库的需求增长。此外，云计算的普及也是推动数据库市场需求的一个重要因素。越来越多的企业选择将数据库迁移到云端，利用云数据库的弹性和可扩展性来支持其快速发展的业务需求。根据市场研究公司Gartner的数据，2023年全球云数据库市场将达到约60亿美元，预计未来几年将继续增长。尤其是在企业对快速数据访问和分析的需求日益增加的背景下，数据库技术的角色变得愈发重要。从数据安全的角度来看，随着数据泄露事件的频发，数据库安全性已成为企业不可忽视的议题。越来越多的企业开始重视数据加密、身份验证、访问控制等安全功能，这对数据库厂商提出了更高的技术要求。企业不仅希望从数据库中获得高效的数据处理能力，还要求其具备强大的安全保护能力。');
INSERT INTO documents(topic, content) VALUES ('数据库技术热点', '近年来，数据库技术持续创新，出现了一些显著的技术热点。首先，分布式数据库逐渐成为技术发展的主流。随着云计算、大数据以及人工智能的不断发展，传统的单机数据库难以满足处理海量数据的需求。分布式数据库通过将数据存储和计算任务分配到多个节点，提供了更高的可扩展性和容错能力。其次，多模态数据库和图数据库正在引领数据库技术的另一个热点。多模态数据库能够支持不同的数据模型（如文档型、关系型、图形、列存储等），使得企业能够在同一个平台上处理各种类型的数据。图数据库，因其在社交网络、推荐系统和路径分析等领域的强大优势而受到越来越多企业的关注。另外，随着 人工智能（AI） 和 机器学习（ML） 的发展，智能数据库成为了一个新兴趋势。智能数据库通过集成AI功能，能够自动化数据处理过程，包括自动索引优化、查询优化、异常检测等。这不仅提高了数据库的效率，还帮助用户降低了维护成本。Serverless 和 无服务器数据库也是当前的热点技术之一，特别是在云数据库领域。这种架构使得用户只需为使用的计算资源付费，而无需管理和维护数据库实例。无服务器架构简化了数据库管理，特别适合短期、大量负载波动的应用场景。');
INSERT INTO documents(topic, content) VALUES ('数据库未来方向', '未来，数据库行业将朝着更高效、更智能、更安全的方向发展。一方面，随着数据量的剧增，自动化和智能化将成为未来数据库技术发展的重要方向。AI和ML的集成将使得数据库能够自主学习、优化自身的性能，例如通过预测查询负载、自动调整索引等方式来提升整体效率。数据库的自动化管理也将大大减轻开发人员和运维人员的负担，提升企业的生产力。云数据库的进一步普及和无服务器架构的推广也是未来的关键发展趋势。随着更多企业采用混合云和多云架构，云数据库的弹性、扩展性和成本效益将成为企业选择云数据库的主要动力。云数据库不仅能够支持高效的数据存储，还能够为企业提供高可用性、容错性和灾难恢复能力。此外，随着 数据隐私保护 和 合规性要求 的不断提高，数据库厂商将不断增强数据加密、访问控制和审计等功能。区块链技术也有望在数据库领域得到应用，通过去中心化的方式提升数据安全性和透明度，尤其是在涉及敏感数据和财务数据的场景中。边缘计算的兴起也将影响数据库技术的发展。未来的数据库不仅仅会在数据中心中运行，还将能够在网络边缘设备上运行，支持实时数据处理和低延迟的应用场景，特别是在物联网（IoT）和智能设备领域，数据库将具备更强的边缘计算能力。');
INSERT INTO documents(topic, content) VALUES ('数据库营收状态', '近年来，数据库市场的营收状态整体呈现出稳定增长的态势。根据IDC和Gartner的数据显示，2022年全球数据库市场的整体规模达到了近500亿美元。随着数字化转型的推进，企业对数据库的需求不断增长，预计未来几年这一市场将继续扩展。云数据库的普及尤其
```


版权所有 © 华为云计算技术有限公司

```
推动了市场的增长，许多大型云服务提供商的云数据库产品不断取得市场份额，成为营收增长的重要来源。其中，数据库即服务（DBaaS）作为云计算的一个重要分支，正在成为最具增长潜力的市场之一。DBaaS能够为企业提供即插即用的数据库服务，减少了企业在硬件、软件、运维等方面的投入。因此，DBaaS已成为一些数据库厂商（如 MongoDB、Snowflake）的主要营收来源。在传统数据库厂商中，如 Oracle 和 Microsoft SQL Server，尽管其依然在全球市场占据领先地位，但其增长速度相对放缓。为了应对云数据库的挑战，这些传统厂商也在逐步将其数据库产品转向云端，提供混合云数据库和云数据库服务。此外，随着更多企业将数据存储和计算需求迁移到云端，NoSQL 数据库和 NewSQL 数据库的兴起为市场提供了新的增长动力。随着大数据和人工智能应用场景的增加，这些数据库产品的市场需求也在不断扩大，进一步推动了数据库行业的营收增长。');
```

步骤2 粗筛相关文档。使用rank函数先对documents表中的topic与用户问题进行粗略的相关性匹配，返回相关性最高的5个长文本语料，再利用chunk_text_recursively函数划分chunk，存入chunk_text表中。

```
WITH topics AS (
    SELECT array_agg(topic) AS topics_array
    FROM documents
),
rank_result AS (
    SELECT
        ai.rank('请生成一份2020年到2025年数据库行业的发展情况的调研报告，重点涵盖市场需求、技术热点和未来方向三个方面', topics_array) AS result
    FROM topics
),
rank_score AS (
    SELECT
        (jsonb_each_text(result)).key AS content,
        (jsonb_each_text(result)).value AS score
    FROM rank_result
),
related_documents AS (
    SELECT content, score
    FROM rank_score
    ORDER BY score DESC
    LIMIT 5
)
INSERT INTO chunk_text (chunk)
SELECT unnest(ai.chunk_text_recursively(rd.content, 300)) AS chunk
FROM related_documents rd;
```

步骤3 将文本分块（chunk）进行向量化。将每个chunk经Embedding模型转换为高维向量，持久化存储于DWS。

```
UPDATE chunk_text
SET embedding = ai.openai_embed(chunk)::VECTOR;
```

----结束

步骤四：生成调研报告

步骤1 检索相似chunk并生成调研报告。

```
WITH query_embedding AS (
    SELECT ai.openai_embed('请生成一份2020年到2025年数据库行业的发展情况的调研报告，重点涵盖市场需求、技术热点和未来方向三个方面')::VECTOR AS embedding
),
similarity_chunks AS (
    SELECT
        ct.id,
        ct.chunk,
        (ct.embedding <=> qe.embedding) AS similarity
    FROM
        chunk_text ct, query_embedding qe
    ORDER BY similarity
    LIMIT 10
),
chunks AS (
    SELECT string_agg(chunk, ' ') AS concatenated_chunks
    FROM similarity_chunks
),
```


版权所有 © 华为云计算技术有限公司

```
report AS (
    SELECT ai.openai_chat_complete(
        jsonb_build_array(
            jsonb_build_object(
                'role', 'system',
                'content', '请基于我给你的问题和文本内容生成一份调研报告'
            ),
            jsonb_build_object(
                'role', 'user',
                'content', '请生成一份2020年到2025年数据库行业的发展情况的调研报告，重点涵盖市场需求、技术热点和未来方向三个方面 ' || concatenated_chunks
            )
        )
    ) AS report_content
    FROM chunks
)
INSERT INTO reports (content)
SELECT report_content FROM report;
```

步骤2 **查看生成结果。**示例的结果是将Markdown格式以纯文本的形态存入reports表中，可以直接导出content保存为.md文件。

```
SELECT * FROM reports;
```

----结束

导出的 Markdown 格式展示

以下为reports表中，content列导出文本数据的Markdown展示。（报告中具体数值在本示例中仅供参考，不代表具体真实结果）

2020-2025年数据库行业调研报告

——市场需求、技术热点与未来方向分析

一、摘要

本报告基于2020年至2025年数据库行业的市场数据、技术演进及行业趋势，系统分析了市场需求变化、技术热点演进及未来发展方向。核心结论如下：

- **市场需求**：全球数据库市场规模从2020年的450亿美元增长至2025年的780亿美元，复合年增长率（CAGR）达15%，主要驱动力来自数字化转型、数据量激增及云原生需求。
- **技术热点**：分布式数据库、多模数据处理、AI与数据库融合成为核心趋势，云原生存储与弹性扩展技术显著提升市场竞争力。
- **未来方向**：边缘计算与数据库协同、量子数据库探索、可持续性设计（如绿色数据库）将成为关键创新领域。

二、市场需求分析

1. **市场规模与增长**
   - **营收状态**：
     - 2020年全球数据库市场营收约450亿美元，2025年预计突破780亿美元（IDC数据）。
     - 云数据库占比从2020年的35%提升至2025年的60%，成为市场增长核心驱动力。
   - **区域分布**：
     - 北美（40%）、亚太（30%）、欧洲（25%）为三大市场，亚太地区增速最快（CAGR 18%），受益于中国、印度数字化转型。
2. **需求驱动因素**
   - **数字化转型**：企业对实时数据处理、业务敏捷性需求推动数据库升级，如金融、电商、医疗行业对高可用性数据库的依赖度提升。
   - **数据量激增**：全球数据总量从2020年的49ZB增长至2025年的175ZB（Statista），非结构化数据占比超80%，推动多模数据库（如支持文档、图、时空数据）需求。
   - **云原生与混合云**：企业从传统本地部署向云迁移，2025年全球云数据库用户渗透率超70%。
3. **行业痛点与挑战**
   - 数据安全与合规（如GDPR）成本上升。
   - 传统数据库性能瓶颈难以满足实时分析需求。

5.4 基于 DWS 的向量计算功能实现简单的商品搜索推荐系统

在AI与大模型技术快速发展的背景下，数据处理需求从传统的“**精确匹配**”转向了更复杂的“**语义理解**”。例如，在电商平台上，用户可能希望通过上传一张图片来搜索相似的商品，或者通过一段文字描述来找到最相关的商品。然而，传统的数据库系统在处理结构化数据时表现良好，但在处理非结构化数据时，如文本、图像、音视频等，由于缺乏对数据背后“含义”的理解，导致搜索结果不准确。

版权所有 © 华为云计算技术有限公司

具体来说，传统数据库依赖于精确匹配或字段索引查询，无法满足AI应用中对模糊相似搜索的需求，如**以文搜图**、**智能问答**和**推荐系统**等。如何解决这一问题，提高数据处理的效率和准确性？DWS集成**pgvector（0.8.0）**插件，可插拔式加载，实现**库内向量计算检索**能力。用户无需迁移数据或重构应用架构，即可在现有数据仓库中实现**向量检索**功能，从而有效解决上述问题。

典型场景如下：

- **以文搜图/以图搜图**：找到与输入图片最相似的图片。
- **智能问答/RAG**：在知识库中找到与用户问题最相关的文档片段。
- **推荐系统**：找到与用户兴趣向量最相似的商品。

本文以一个**商品搜索推荐系统**为例，介绍如何使用DWS向量计算功能实现语义搜索和商品推荐。

业务背景：

- 商品本身有标题、描述、分类、价格等字段。
- 借助大模型**Embedding**能力，将商品描述编码为10维向量（仅用于演示，实际常用768维向量）。
- 将用户的搜索关键字同样转化为10维向量。
- **目标**：通过相似性搜索找出最符合用户搜索意图的商品。

基本概念

在阅读本文之前，建议先了解以下概念：

表 5-4 基本概念

概念	说明
向量	将文本、图像等非结构化数据通过数学方式映射为固定维度的数值数组（如[0.12, 0.34, -0.21, ...]），使计算机能够通过数值计算来衡量数据之间的语义相似度。例如描述一个人的年龄、身高、体重，可以用[30, 175, 70]这个三维向量表示，让计算机可以理解其语义特征。
Embedding（向量化/嵌入）	将非结构化数据（如商品描述文本）转换为向量的过程，通常借助大语言模型（如BERT、GPT等）完成。Embedding后的向量能够保留原始数据的语义信息。
相似度度量	衡量两个向量之间“接近程度”的数学方法。常用度量包括**欧氏距离（L2距离）**、**内积**和**余弦相似度**。距离越小或相似度越高，表示两个向量在语义上越接近。 • **欧氏距离（L2距离）**：计算两个向量在空间中的“直线距离”。 • **内积（Inner Product）**：两个向量在方向和大小上的“投影”乘积，结果是一个标量。 • **余弦相似度（Cosine）**：比较两个向量的“方向”是否一致。

版权所有 © 华为云计算技术有限公司

概念	说明
召回率	检索结果中相关结果占全部相关结果的比例。举例：数据库中与查询相关的内容有100条，系统检索后返回了80条相关的，那召回率就是 80/100 = 80%。 召回率与查询速度往往是此消彼长的关系，精确搜索的召回率为100%，近似搜索会牺牲部分召回率以换取更快的查询速度。

向量计算功能介绍

DWS的向量计算的详细语法介绍请参见**向量计算**章节。

向量数据类型

表 5-5 向量数据类型

数据类型	精度	支持维度	典型用途
vector	单精度浮点，4 字节。	<=16000	传统 embedding。
halfvec	半精度浮点（16 位浮点数，每元素 2 字节）。	<=16000	高维 embedding、内存敏感场景。
bit	二进制向量类型（bit 向量）。	<= 64000	二进制哈希、图像 / 特征布尔表示。
sparsevec	稀疏向量类型。	<=16000非0维度	文本TF-IDF稀疏特征、超高维稀疏 embedding。

向量距离/相似度操作符

表 5-6 向量距离/相似度操作符

操作符	距离 / 相似度类型	说明
<->	Euclidean距离 (L2)	欧氏距离。
<#>	（负）内积 (Inner Product)	内积。
<=>	余弦（Cosine）相似度	向量方向差异。
<+>	L1距离 (Taxicab)	向量坐标绝对差之和。
<~>	Hamming距离	用于二进制bit向量，计算位不同数目。
<%>	Jaccard距离	用于二进制bit向量/集合表示，衡量交集/并集差异。

版权所有 © 华为云计算技术有限公司

向量索引类型

表 5-7 向量索引类型

索引类型	原理	构建成本	查询速度 / 召回 (speed- recall)	主要调优参数
IVFFlat	将向量分成若干“lists”（簇/桶），查询时只扫描部分 lists。	构建快、内存使用低。	查询速度中等；召回率略低。扫描范围受“probes”参数影响。	• lists（簇数） • probes（扫描簇数） • maxProbes（最大扫描簇数）
HNSW	基于图结构（多层可导航小世界图），通过跳跃访问近邻点。	构建较慢、内存占用较高。	查询速度快、召回率高；尤其适合高维、低延迟场景。	• m（每层最大连接数） • ef_construction（构建候选数） • ef_search（查询候选数） • max_scan_tuples（最大扫描节点数） • mem_scan_multiplier（内存动态扩张率）

pgvector 扩展里的操作符类

在pgvector（PostgreSQL的向量扩展）中，针对不同的距离度量方式，提供了三种主要的操作符类。该操作符，通常在创建索引的语句中指定。例如，

```
CREATE INDEX ON products USING hnsw (embedding vector_l2_ops)
WITH (m = 16, ef_construction = 200);
```

表 5-8 操作符

操作符类	距离度量方式	运算符	数学含义	适用场景
vector_l2_ops	**欧氏距离** (L2)	<->	两点间的直线距离。	通用场景，推荐系统、聚类分析。
vector_ip_ops	**内积 (Inner Product)**	<#>	向量点积，衡量方向一致性。	语义搜索、信息检索（尤其是文本匹配）。
vector_cosine_ops	**余弦相似度 (Cosine)**	<=>	向量夹角的余弦值，忽略长度。	文本相似度、句子匹配（最常用）。

版权所有 © 华为云计算技术有限公司

前提条件

- DWS集群版本须为**9.1.1.200及以上**。
- 使用向量计算功能前，需**联系技术支持**修改集群参数**feature_support_options**，开启**enable_pgvector**选项。
- 已通过大语言模型（如BERT、GPT等）获得商品描述的Embedding向量。**本文不涉及向量生成过程，假设向量已就绪**。

步骤一：安装 pgvector 扩展

步骤1 确认GUC参数**feature_support_options**已设置为**enable_pgvector**。请联系技术支持进行设置。

步骤2 在数据库中启用向量计算功能。

```
CREATE EXTENSION pgvector;
```

步骤3 执行以下查询，若返回pgvector则表示安装成功。

```
SELECT extname FROM pg_extension WHERE extname = 'pgvector';
```

```
gaussdb=> SELECT extname FROM pg_extension WHERE extname = 'pgvector';
 extname
----------
 pgvector
(1 row)
```

----结束

步骤二：创建商品表及插入数据

步骤1 创建存储商品信息及向量数据的表。

业界常用768或1024维向量，涉及数据太大不便演示。为便于示例演示，本实践取10维向量数据用于模拟。

其中，vector(10)表示该字段存储10维的向量。向量维度需与Embedding模型输出维度一致。

```
DROP TABLE IF EXISTS products;
CREATE TABLE products (
    id bigserial PRIMARY KEY,
    title text,
    description text,
    price numeric,
    embedding vector(10)   -- 10维向量，由商品描述(description)通过大模型Embedding生成
);
```

步骤2 向商品表中插入商品信息及对应的向量数据。

```
INSERT INTO products (title, description, price, embedding) VALUES
('Wireless Earbuds', 'Bluetooth wireless earbuds with charging case', 59.9, '[0.12, 0.34, -0.21, 0.15, -0.08, 0.33, 0.02, -0.41, 0.19, 0.07]'),
('Noise Cancelling Headphones', 'Over-ear headphones with active noise cancellation', 129.9, '[0.11, 0.36, -0.19, 0.17, -0.06, 0.31, 0.04, -0.39, 0.21, 0.05]'),
('Gaming Headset', 'Wired gaming headset with microphone', 79.9, '[0.10, 0.33, -0.25, 0.14, -0.03, 0.29, 0.01, -0.37, 0.18, 0.09]'),
('Smartphone Stand', 'Adjustable phone stand for desk use', 12.9, '[0.45, -0.12, 0.08, 0.48, -0.20, 0.11, 0.37, -0.05, 0.44, -0.15]'),
('USB-C Charger', 'Fast charging USB-C power adapter', 19.9, '[0.42, -0.15, 0.05, 0.46, -0.18, 0.08, 0.34, -0.07, 0.41, -0.12]'),
('Mechanical Keyboard', 'Mechanical keyboard with blue switches', 89.9, '[0.55, 0.02, -0.31, 0.50, 0.06, -0.28, 0.22, -0.44, 0.53, -0.09]'),
```


版权所有 © 华为云计算技术有限公司

```
('Wireless Mouse', 'Ergonomic wireless mouse', 29.9, '[0.53, 0.01, -0.28, 0.51, 0.04, -0.25, 0.20, -0.42, 0.49, -0.07]'),
('Laptop Backpack', 'Water-resistant laptop backpack', 49.9, '[0.60, -0.05, -0.10, 0.58, -0.08, -0.15, 0.42, -0.02, 0.56, -0.13]'),
('4K Monitor', '27-inch 4K UHD computer monitor', 299.9, '[0.58, 0.04, -0.35, 0.55, 0.07, -0.32, 0.39, -0.10, 0.61, -0.18]'),
('Webcam', 'HD webcam for video conferencing', 39.9, '[0.52, -0.01, -0.20, 0.49, -0.03, 0.15, 0.36, -0.08, 0.47, -0.11]'),
('Bluetooth Speaker', 'Portable Bluetooth speaker with deep bass', 45.9, '[0.14, 0.30, -0.18, 0.13, -0.09, 0.28, 0.03, -0.35, 0.16, 0.01]'),
('Smart Watch', 'Fitness tracking smart watch', 99.9, '[0.20, 0.40, -0.22, 0.18, -0.05, 0.37, 0.09, -0.41, 0.23, 0.06]'),
('Fitness Tracker', 'Lightweight activity and sleep tracker', 49.9, '[0.22, 0.38, -0.24, 0.19, -0.07, 0.35, 0.07, -0.39, 0.25, 0.04]'),
('Tablet Stylus', 'Stylus pen for tablets', 25.9, '[0.48, -0.10, 0.12, 0.45, -0.17, 0.14, 0.40, -0.06, 0.50, -0.16]'),
('Laptop Cooling Pad', 'Cooling pad with dual fans', 34.9, '[0.57, -0.02, -0.15, 0.54, -0.04, -0.12, 0.46, -0.01, 0.59, -0.14]');
```

----结束

步骤三：查询数据

步骤1 **语义相似度查询**。根据用户搜索词的向量，查找语义最相似的商品。<->为L2距离操作符，值越小表示越相似。LIMIT 10表示返回相似度最高的Top-10商品。

```
SELECT id, title, price
FROM products
ORDER BY embedding <=> '[0.09, -0.05, 0.92, 0.11, -0.03, 0.85, 0.30, -0.50, 0.15, 0.02]'   -- 用户搜索词的Embedding向量（10维）
LIMIT 10;
```

```
gaussdb=> SELECT id, title, price
gaussdb-> FROM products
gaussdb-> ORDER BY embedding <=> '[0.09, -0.05, 0.92, 0.11, -0.03, 0.85, 0.30, -0.50, 0.15, 0.02]'
gaussdb-> LIMIT 10;
 id |            title            | price
----+-----------------------------+-------
 14 | Tablet Stylus               |  25.9
  4 | Smartphone Stand            |  12.9
 12 | Smart Watch                 |  99.9
  2 | Noise Cancelling Headphones | 129.9
  5 | USB-C Charger               |  19.9
  1 | Wireless Earbuds            |  59.9
 11 | Bluetooth Speaker           |  45.9
 13 | Fitness Tracker             |  49.9
  3 | Gaming Headset              |  79.9
 10 | Webcam                      |  39.9
(10 rows)
```

步骤2 **混合查询。**在语义相似度查询的基础上，叠加传统结构化过滤条件，实现更精准的搜索。先按价格过滤（price < 2000），再按语义相似度排序，适用于“预算范围内找最相似商品”的场景。

```
SELECT id, title, price
FROM products
WHERE price < 2000                          -- 先按价格过滤
ORDER BY embedding <=> '[0.09, -0.05, 0.92, 0.11, -0.03, 0.85, 0.30, -0.50, 0.15, 0.02]'  -- 再按语义相似度排序
LIMIT 10;
```


版权所有 © 华为云计算技术有限公司

```
 id |            title            | price
----+-----------------------------+-------
 14 | Tablet Stylus               |  25.9
  4 | Smartphone Stand            |  12.9
 12 | Smart Watch                 |  99.9
  2 | Noise Cancelling Headphones | 129.9
  5 | USB-C Charger               |  19.9
  1 | Wireless Earbuds            |  59.9
 11 | Bluetooth Speaker           |  45.9
 13 | Fitness Tracker             |  49.9
  3 | Gaming Headset              |  79.9
 10 | Webcam                      |  39.9
(10 rows)
```

步骤3 **创建向量索引加速查询。**在大数据量场景下，通过创建**近似近邻搜索索引**提升向量查询性能。

DWS向量计算默认使用**精确近邻**搜索，提供100%召回率但查询速度较慢。当数据量较大时，建议创建HNSW或IVFFlat索引，以牺牲少量召回率为代价大幅提升查询速度。

索引参数说明参见**表5-9**。vector_l2_ops表示**欧氏距离** (L2)所用的操作符类，更多请参见**表5-8**。

⚠ 注意

索引创建时指定的操作符类（如vector_l2_ops）必须与查询中使用的距离操作符（如<->）一致。若查询使用<=>（余弦相似度），则需创建索引时使用vector_cosine_ops。

```
-- 方式一：创建HNSW索引（查询性能更优，适合查询频繁的场景）
CREATE INDEX ON products USING hnsw (embedding vector_l2_ops)
WITH (m = 16, ef_construction = 200);


-- 方式二：创建IVFFlat索引（构建速度更快，适合数据量大的场景）
CREATE INDEX ON products USING ivfflat (embedding vector_l2_ops)
WITH (lists = 100);
```

表 5-9 索引参数说明

索引类型	参数	说明	建议值
HNSW	m	每层图中每个节点的最大连接数，影响索引质量和内存占用。	16（默认），数据量大时可增大至48
HNSW	ef_construction	构建索引时的搜索宽度，值越大索引质量越高但构建越慢。	200（默认）
IVFFlat	lists	倒排列表数量，值越大查询越快但召回率可能降低。	建议设为sqrt(行数)左右

----结束

版权所有 © 华为云计算技术有限公司

查询结果解读

场景一：商品搜索

用户搜索词为wireless audio headset，将其转为向量后查询：

```
SELECT id, title
FROM products
ORDER BY embedding <=> '[0.13, 0.35, -0.20, 0.16, -0.07, 0.32, 0.03, -0.38, 0.20, 0.04]'   -- 用户搜索词的Embedding向量（10维）
LIMIT 10;
```

耳机、音响类商品相较于表中其他商品类别，与用户搜索词在语义上更为接近。即使商品标题中没有精确包含headset这个词，仍然会在结果中排序靠前，体现了语义搜索的优势。

```
 id |            title
----+-----------------------------
  2 | Noise Cancelling Headphones
  1 | Wireless Earbuds
 11 | Bluetooth Speaker
 12 | Smart Watch
 13 | Fitness Tracker
  3 | Gaming Headset
 10 | Webcam
  7 | Wireless Mouse
  6 | Mechanical Keyboard
 14 | Tablet Stylus
(10 rows)
```

场景二：商品推荐

用户正在浏览Noise Cancelling Headphones商品，根据商品向量相似性进行关联推荐：

```
SELECT p2.title, p2.price
FROM products p1
JOIN products p2 ON p1.id <> p2.id          -- 排除自身
WHERE p1.title = 'Noise Cancelling Headphones'
ORDER BY p2.embedding <=> p1.embedding      -- 计算两个商品向量的L2距离，越小越相似
LIMIT 10;
```

耳机、音响类商品与用户正在浏览的Noise Cancelling Headphones产品相关性更高，因此排在推荐列表前列。

```
       title        | price
--------------------+-------
 Wireless Earbuds   |  59.9
 Smart Watch        |  99.9
 Bluetooth Speaker  |  45.9
 Gaming Headset     |  79.9
 Fitness Tracker    |  49.9
 Webcam             |  39.9
 Wireless Mouse     |  29.9
 Mechanical Keyboard |  89.9
 Tablet Stylus      |  25.9
 4K Monitor         | 299.9
(10 rows)
```


版权所有 © 华为云计算技术有限公司

总结

通过DWS向量计算扩展能力，我们将传统以结构化查询为核心的数据库系统升级为能够具备语义理解与相似度计算能力的统一数据底座。系统不仅支持向量数据的存储与高效相似度检索，还允许业务方直接通过标准SQL完成语义搜索、推荐与相似内容匹配，无需引入额外的向量引擎或复杂的数据同步链路。

依托HNSW、IVFFlat近似最近邻索引，向量查询在大规模数据场景下依然具备可控的性能与稳定的响应能力。同时，向量检索能力与传统结构化查询、过滤条件、JOIN逻辑的深度融合，使得搜索推荐、个性化分析以及RAG等AI应用可以自然落地在现有数仓与业务体系之上。

最终，这一能力扩展不仅降低了系统架构复杂度，也显著提升了数据平台对新一代智能应用的支撑能力，实现了从”数据查询”向”语义计算与业务决策支持”的演进。

版权所有 © 华为云计算技术有限公司