



数 / 造 / 未 / 来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

滴普湖仓一体架构探索与实践

吴小前

DTCC
2021



北京国际会议中心

2021/8/18-8/20



ChinaUnix.net

ITPUB



数 / 造 / 未 / 来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

背景

DTCC
2021



北京国际会议中心

🕒 2021/8/18-8/20

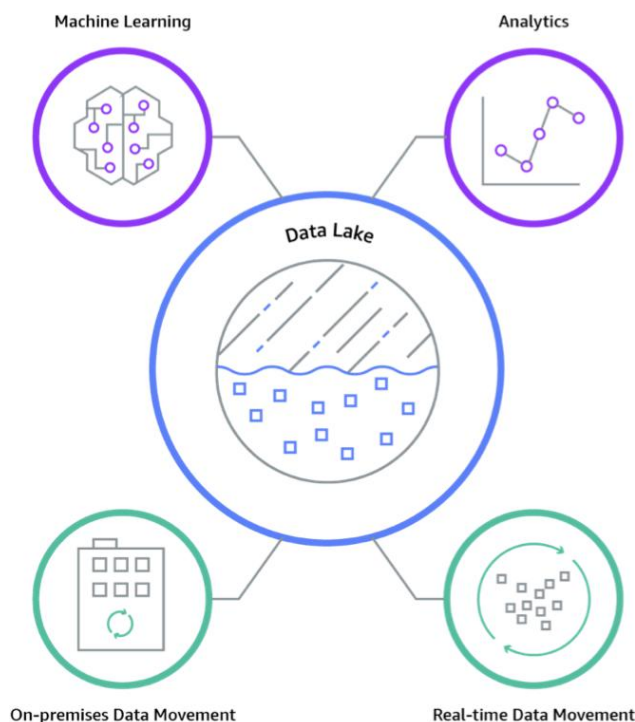


什么是数据湖



DTCC 2021
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

AWS的定义：数据湖提供统一可扩展性的存储，保存结构化与非结构化数据。通常，这些数据以原始形式保存，自此之上，部署各种计算与分析引擎，从而挖掘数据价值。



- 多工作负载 (workload)
 - 流处理
 - 批处理
 - 机器学习
 - 交互式分析
- 多模数据，统一存储
 - 结构化
 - 半结构化
 - 非结构化
 - 二进制数据
- 数据管理
 - 安全与管控
 - 多种数据接入方式



数，造，未，来



IT168.com

ChinaUnix.net

ITPUB

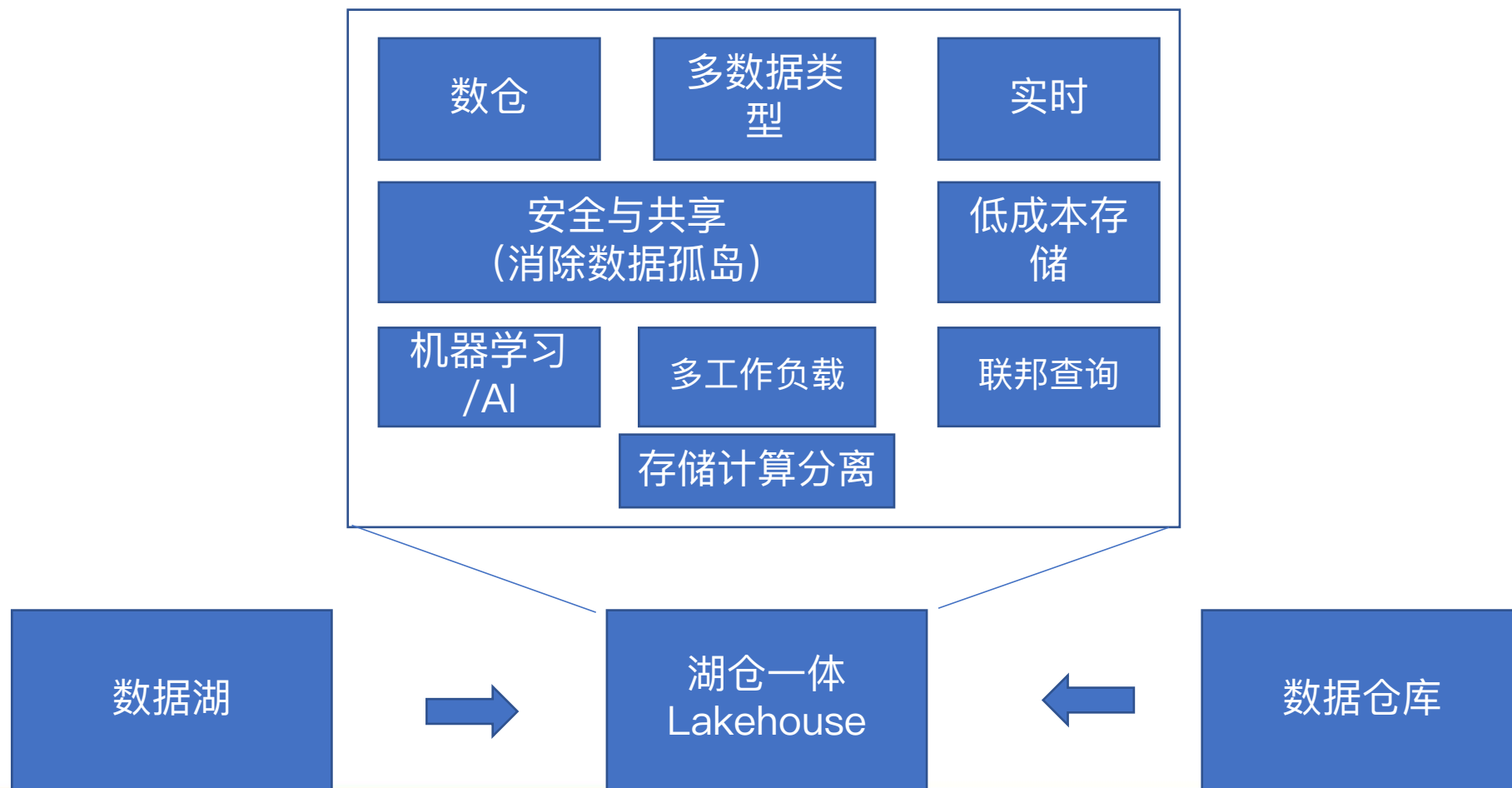


数据仓库与数据湖对比

数据湖	VS	数据仓库
启动成本低 管理成本高	成本	启动成本高 易管理
结构化/半结构化/非结构化/二进制	数据类型	结构化/半结构化
流批处理，AI，数据挖掘和探索	计算负载	流批处理，交互式分析，BI，报表
Schema-on-read，安全与管控方面难度大	数据治理	数据质量高，易管理



数据平台向湖仓一体发展



构建lakehouse非易事



数/造/未/来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

- 存储层
- 表存储引擎
- 数据处理
 - Spark, Flink, MR等
- 数据分析
 - SQL, BI and reporting, ML等
- 数据管控以及安全





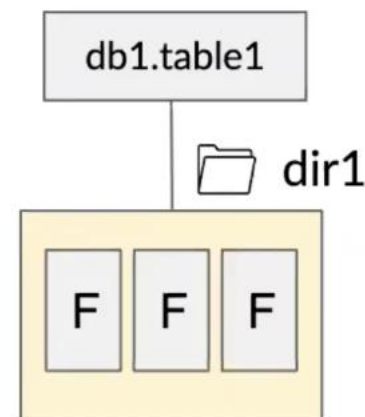
数据湖的问题

- 容易出现数据不一致
 - 数据追加或者更新都是件麻烦的事情
 - 处理作业发生问题，可能出现不完整的数据
- 无法支持并发读写
 - 批处理与实时处理共存的需求
 - 多计算引擎数据访问需求难以协同
- 运维问题
 - 大量小文件
 - 数据质量问题



Hive Table的问题

- 数据访问和修改低效
 - HMS以及元数据设计的问题
 - 大量目录访问
- 跨分区和多作业并发修改会出现数据不一致
- 分区操作复杂，使用者深度参与
 - `WHERE event_ts >= '2021-05-10 12:00:00' AND event_year >= '2021' AND event_month >= '05' AND (event_day >= '10' OR event_month >= '06')`
- 统计成本过高
 - CBO



表内容保存在目录中





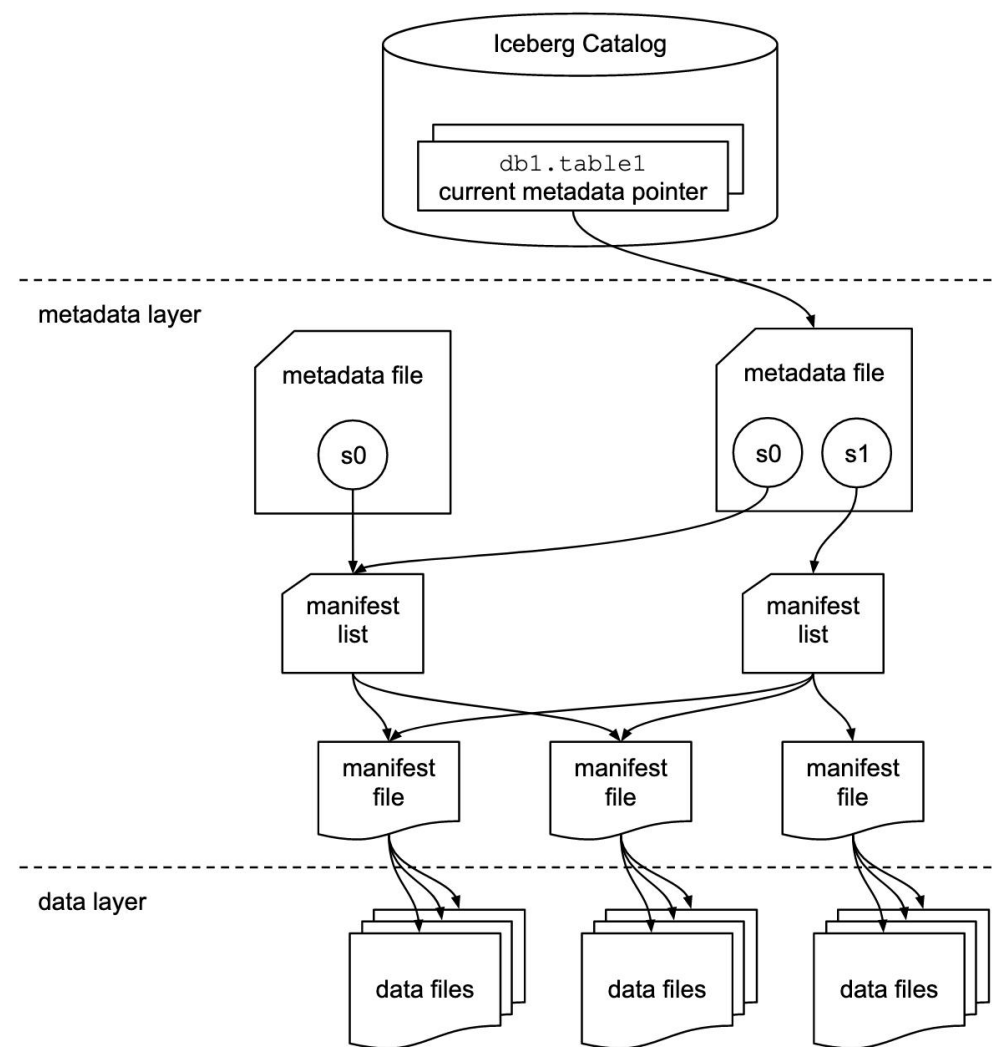
Apache Iceberg



Iceberg: 3种开源解决方案之一

Apache Iceberg is an open table format for huge analytic datasets.

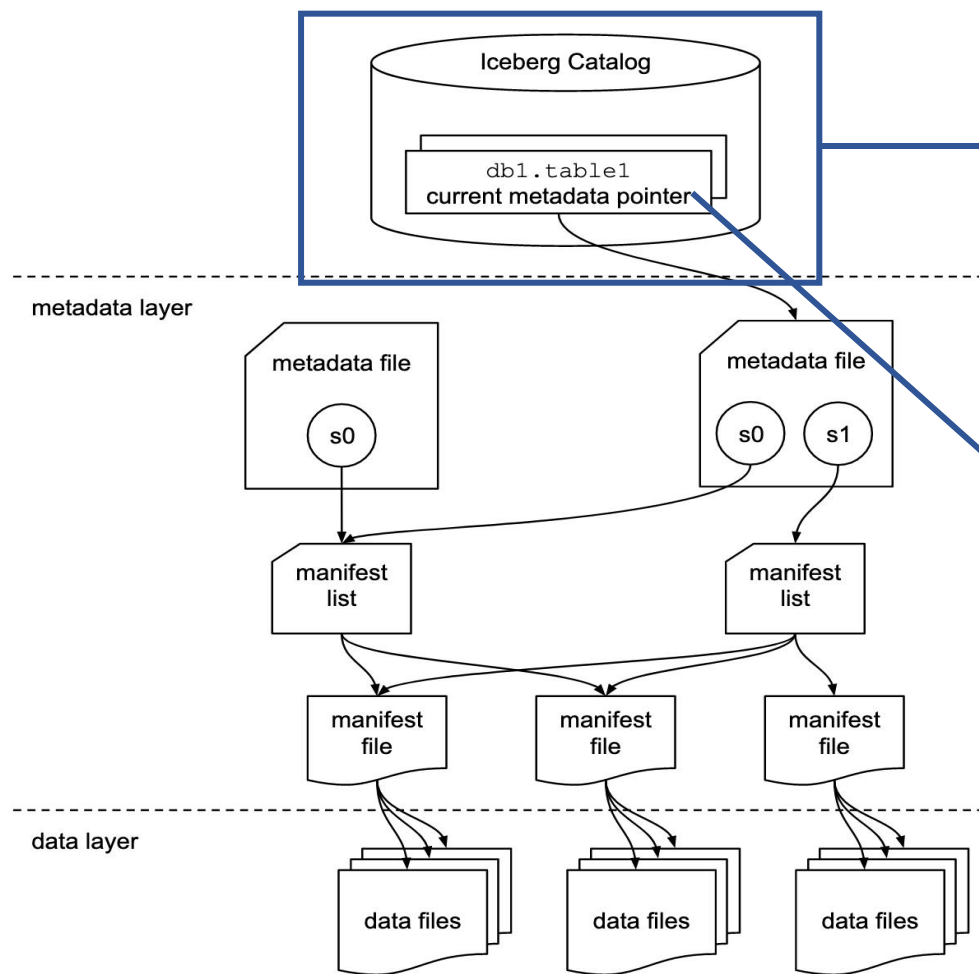
- 支持事务隔离(ACID)
 - 多任务和多引擎并发读写
- 近实时, 且能够胜任小批更新
- 支持历史版本
- 隐式分区以及Partition变更



图片来源: iceberg官网



Iceberg catalog



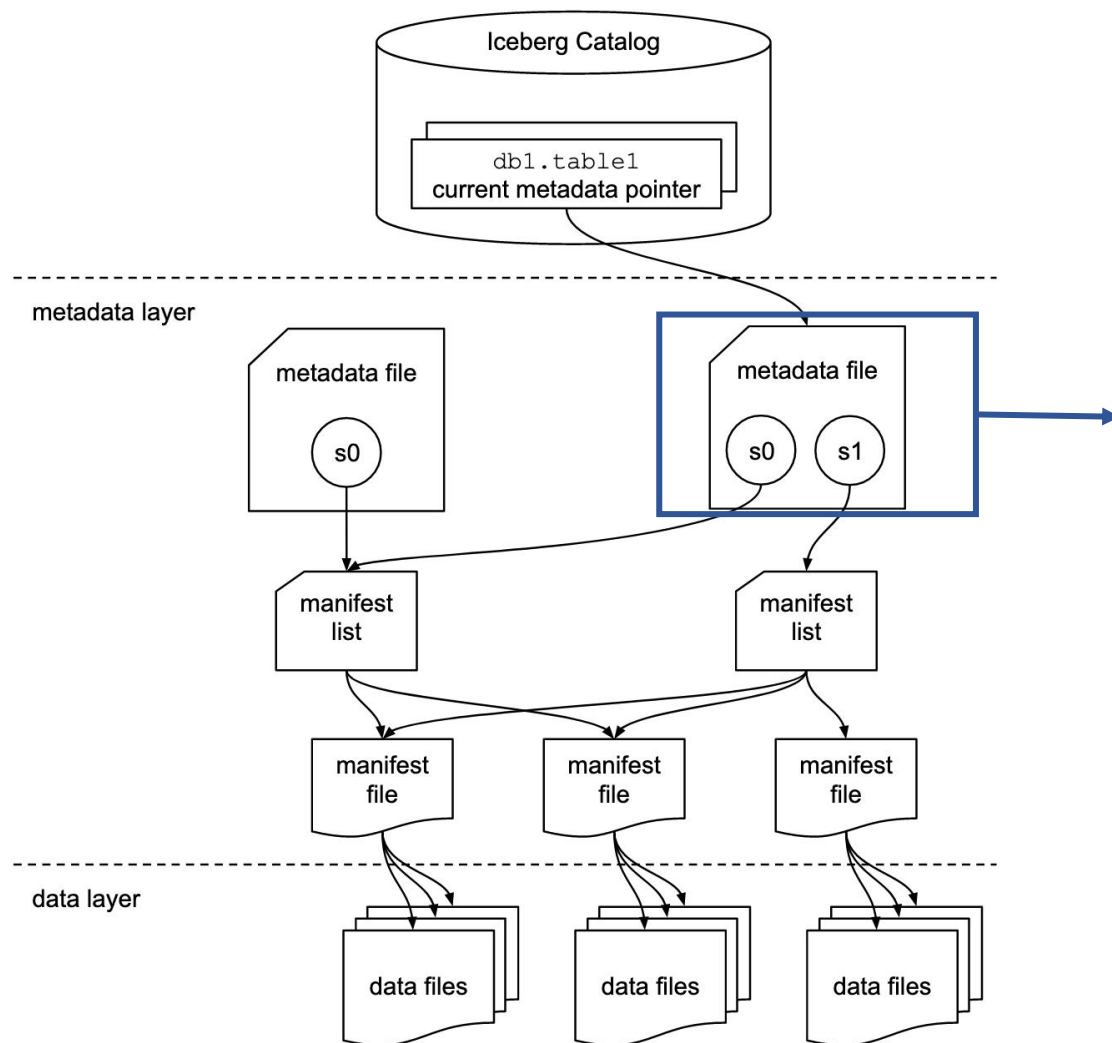
- Iceberg Catalog

- 保存“当前元数据指针”的地方
- 修改此指针时需要原子操作

- Table1的当前指针

- 这个“指针”记录table1的元数据在文件系统的位置



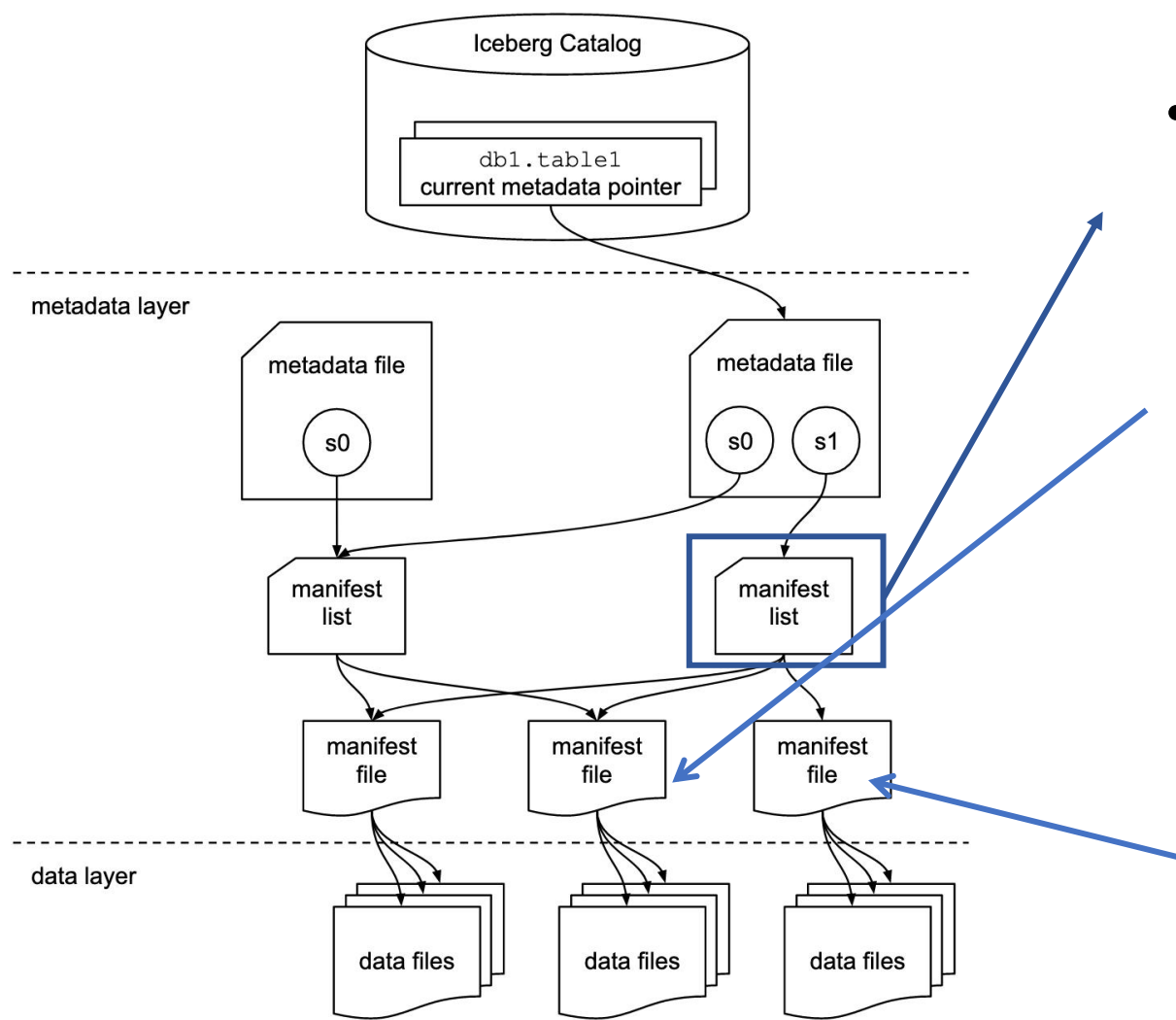


• Metadata file保存基于timeline的元数据

```
{  
  "table-uuid": "<uuid>",  
  "location": "/path/to/table/dir",  
  "schema": {...},  
  "partition-spec": [{partition-details}, ...],  
  "current-snapshot-id": <snapshot-id>,  
  "snapshots": [{  
    "snapshot-id": <snapshot-id>,  
    "manifest-list": "/path/to/manifest/list.avro"  
  }, ...],  
}
```



Manifest list



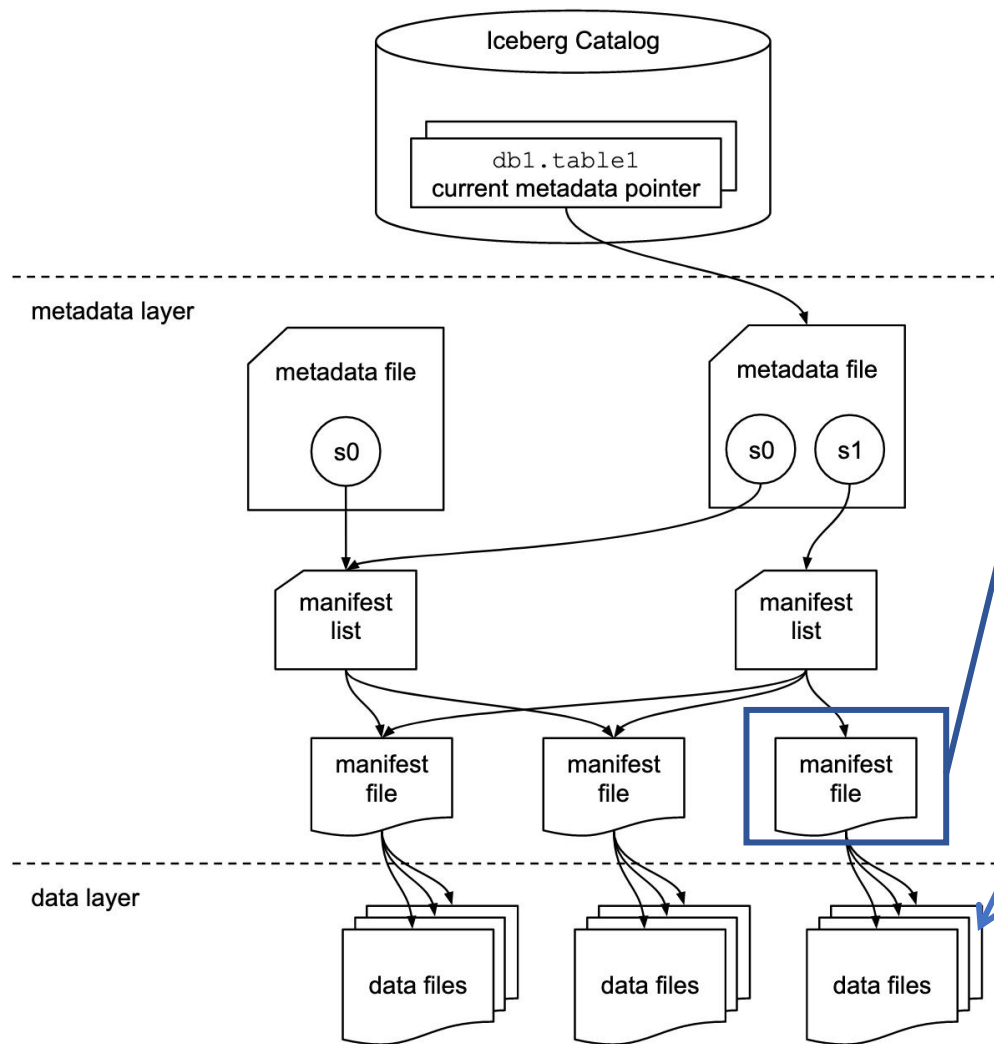
• Manifest list

```
{  
  "manifest-path":  
    "/path/to/manifest/file.avro",  
  "added-snapshot-id": <snapshot-id>,  
  "partition-spec-id": <partition-spec-id>,  
  "partitions": [{partition-info}, ...],  
  ...  
}  
  
{  
  "manifest-path":  
    "/path/to/manifest/file2.avro",  
  "added-snapshot-id": <snapshot-id>,  
  "partition-spec-id": <partition-spec-id>,  
  "partitions": [{partition-info}, ...],  
  ...  
}
```



Manifest文件

- **Manifest文件：保存数据文件列表，以及每个文件的统计信息等**



```
{  
  "data-file": {  
    "file-path": "/path/to/data/file.orc",  
    "file-format": "ORC",  
    "partition": {"<partition-field>":{"<data-type>": <value>}},  
    "record-count": <num-record>,  
    "null-value-counts": [{  
      "column-index": "1", "value": 4  
    }, ...],  
    "lower-bounds": [{  
      "column-index": "1", "value": "xxx"  
    }, ...],  
    "upper-bounds": [{  
      "column-index": "1", "value": "yyyy"  
    }, ...]  
  }  
  ....  
}
```



创建表

```
CREATE TABLE db1.table1 (  
  order_id bigint,  
  customer_id bigint,  
  order_amount DECIMAL(10,2),  
  order_ts TIMESTAMP  
)  
PARTITIONED BY (hour(order_ts));
```

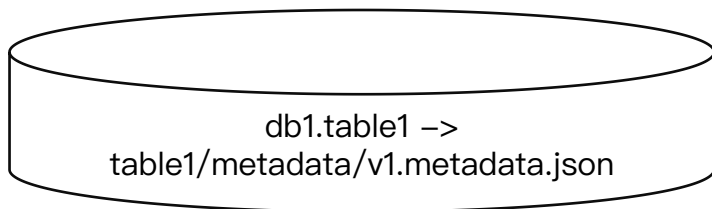
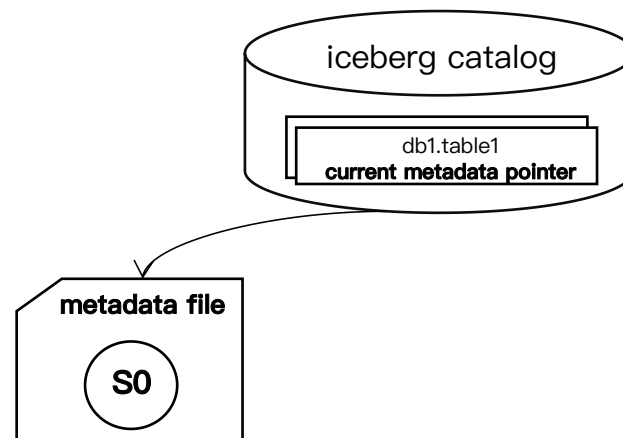
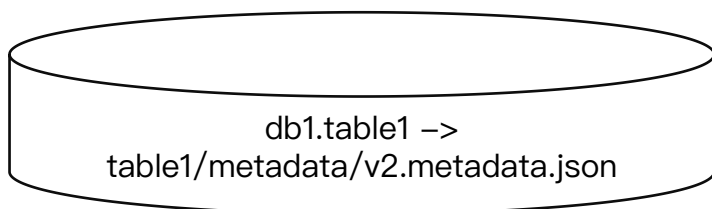


table1
├── data
├── metadata
└── v1.metadata.json

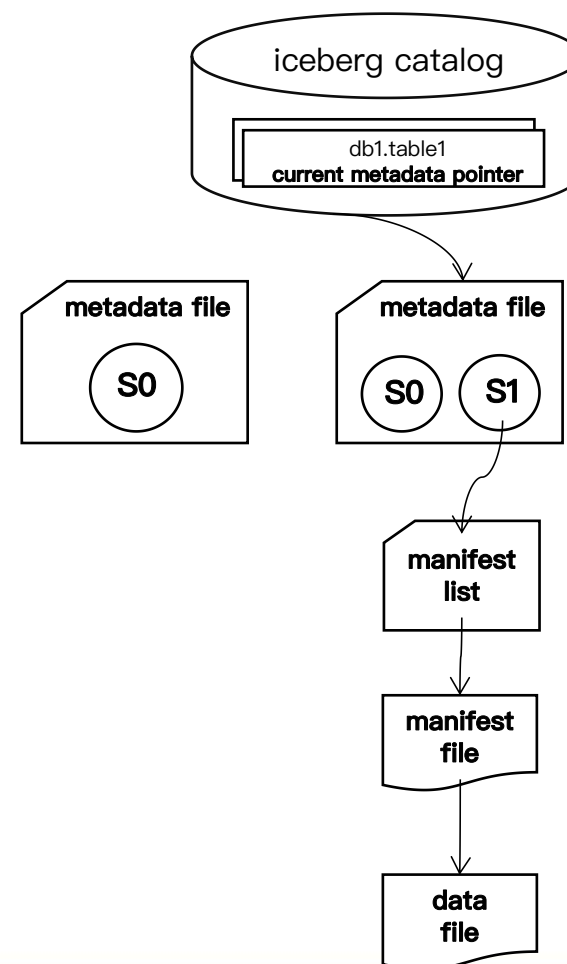


插入数据

```
INSERT INTO db1.table1 VALUES (  
    322,  
    765,  
    23.1,  
    '2021-09-13 12:21:39'  
);
```

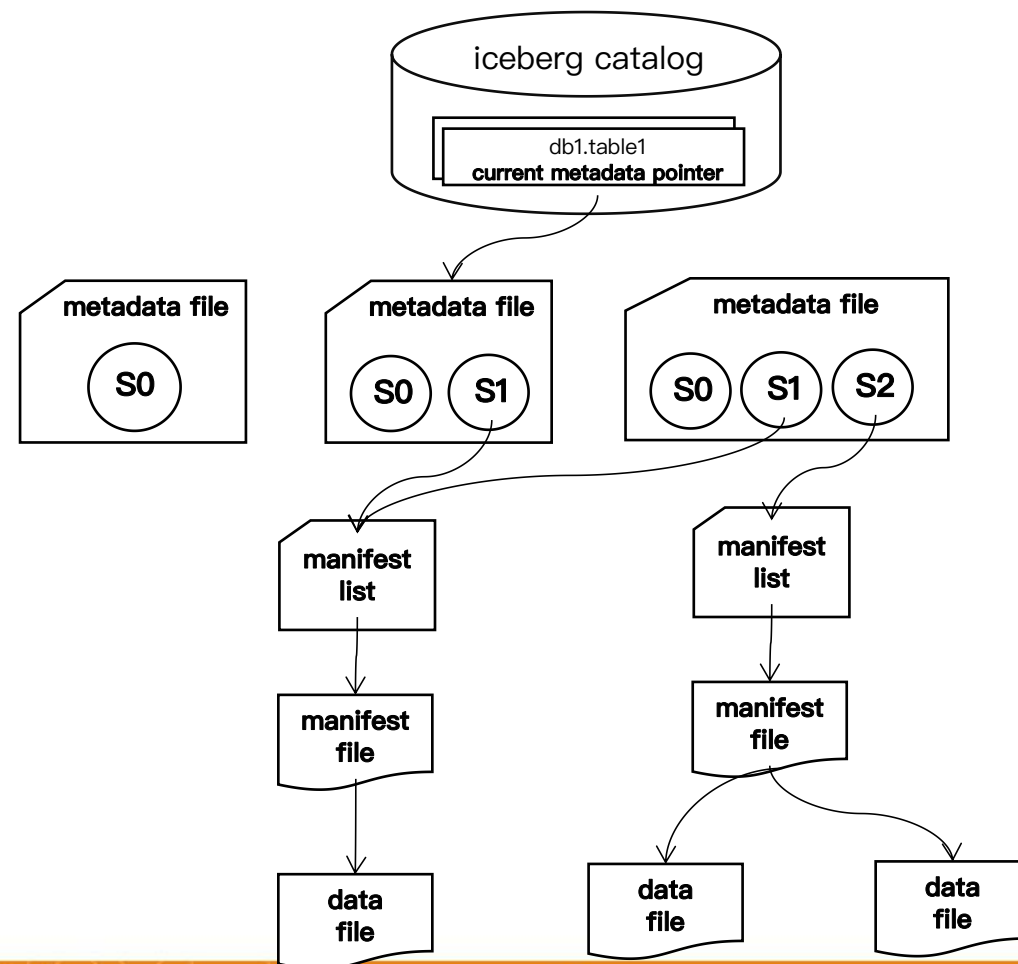
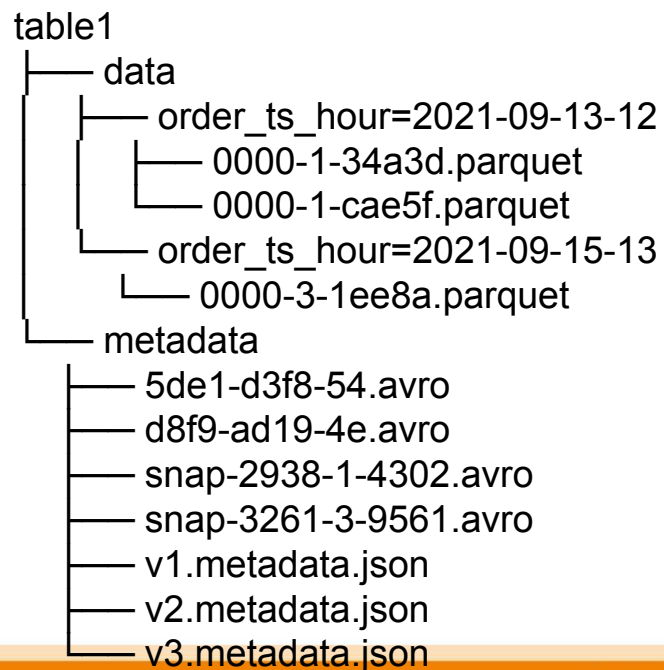
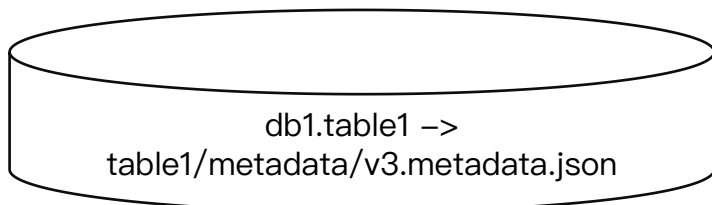


```
table1  
├── data  
│   ├── order_ts_hour=2021-09-13-12  
│   │   └── 0000-1-cae5f.parquet  
└── metadata  
    ├── d8f9-ad19-4e.avro  
    ├── snap-2938-1-4302.avro  
    ├── v1.metadata.json  
    └── v2.metadata.json
```



插入更多数据

```
INSERT INTO db1.table1 select * from  
table1_stage
```





实践方案



实时数据湖方案



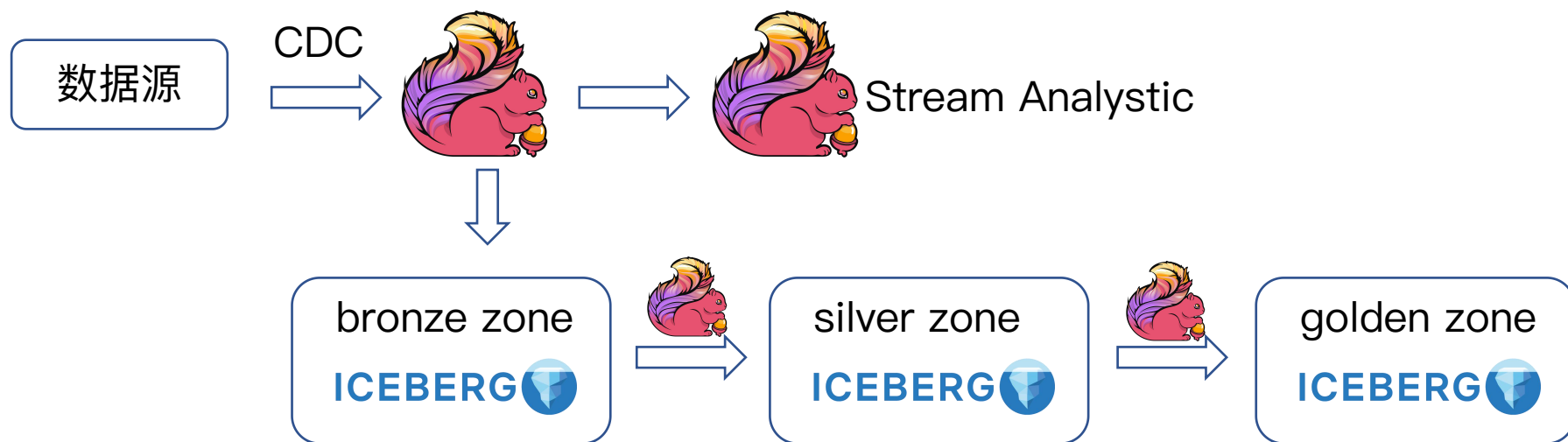


优劣分析

- + 存储层统一
- + 降低存储成本
- + 实现了部分业务的流批统一，达到分钟级延时
- + 解决了小文件问题，提升了查询速度
- 达不到秒级实时
- 实时开发难度变大



部分业务实现秒级实时



一套代码批和流两种运行模式，实现开发效率提升

未来实现Late batching，去掉批处理，节省计算资源





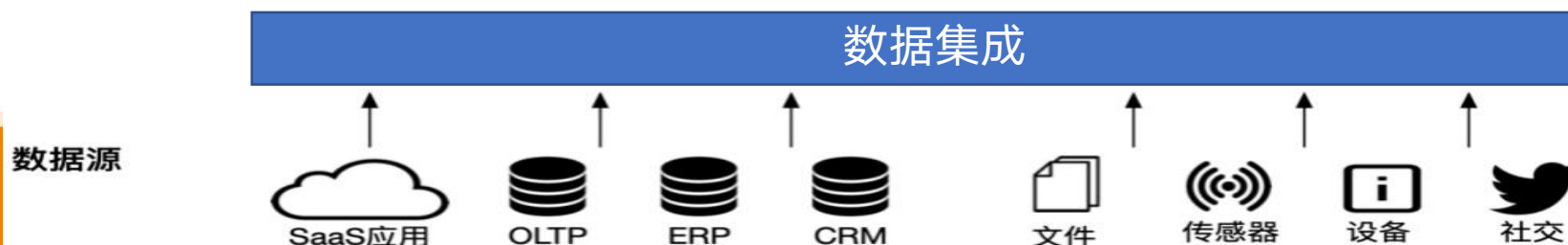
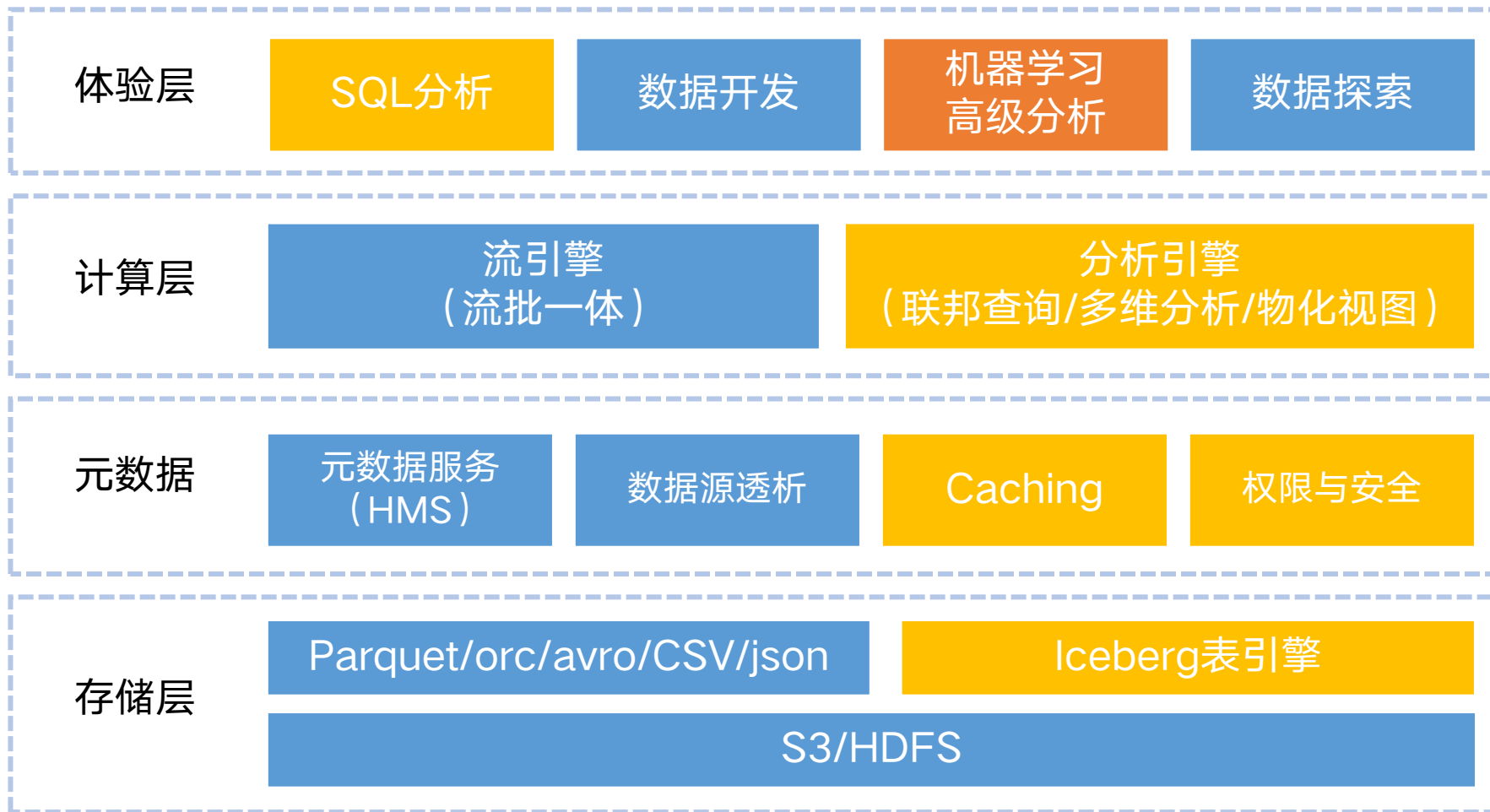
Fastdata数据平台介绍



Fastdata实时数据分析平台 架构



数 / 造 / 未 / 来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021



Fastdata敏捷开发平台



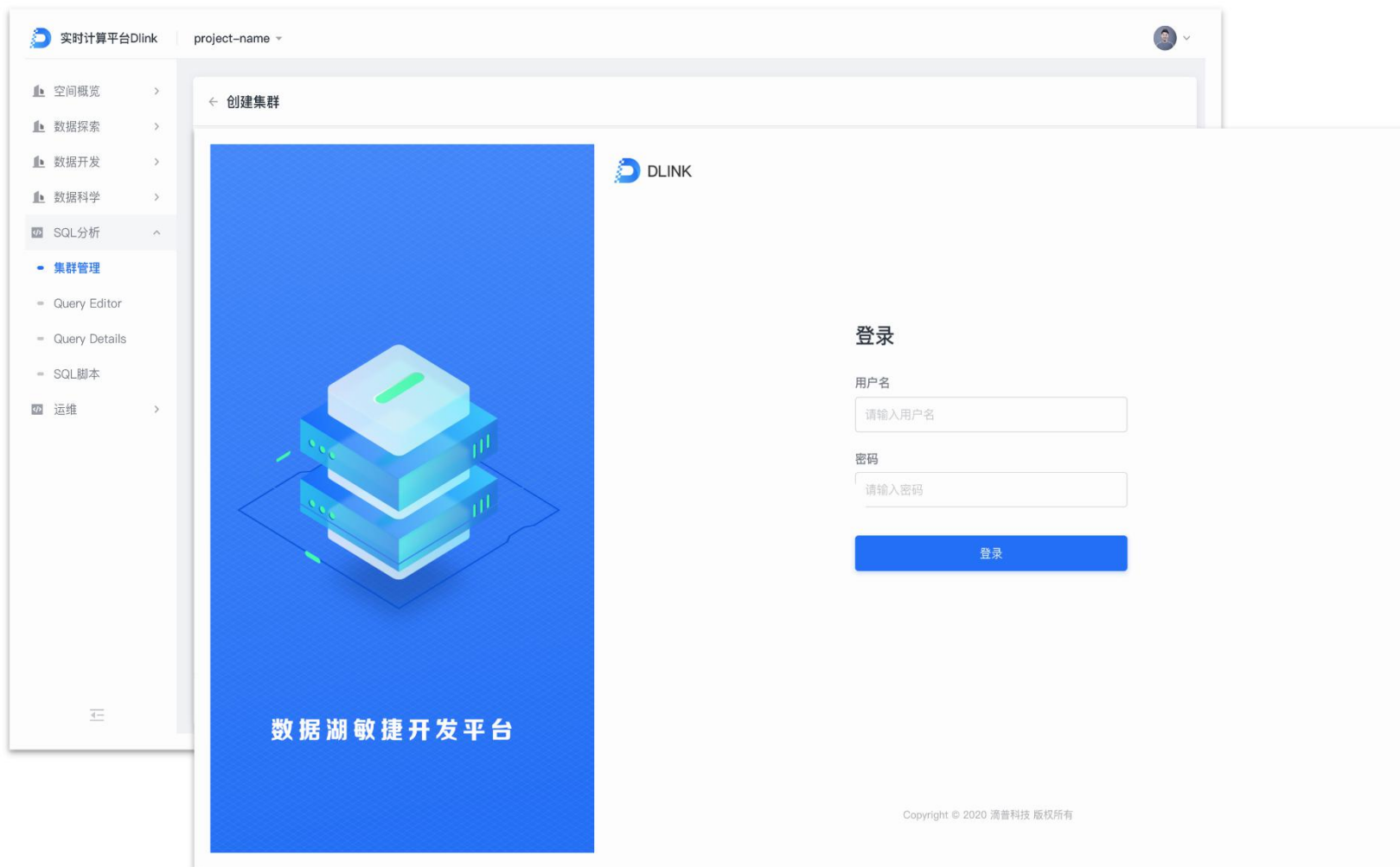
数/造/未/来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

支持作业开发、数据处理、数据分析、数据模型、元数据 等功能。

优势：简单、中立、敏捷、低成本

特性：

- 1、空间概览（数据安全）
- 2、数据探索
- 3、数据开发
- 4、数据科学
- 5、SQL分析
- 6、运维监控



特性：数据探索



数 / 造 / 未 / 来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

数据探索能力：数据源数据profiling

创建数据源

数据源分配

数据预览

数据应用

请选择存储连接类型

- ☒ Kafka
- ☐ MySQL
- ☐ S3
- ☐ HDFS

取消 下一步

新增MYSQL存储连接

* 连接名称

请为当前连接命名

* 主机

请输入

* 端口

请输入

* 用户名

请输入

* 密码

请输入

网络探测

取消

创建

数据源名称	数据源ID	连接状态	连接时间	创建时间	操作
name	123456	连接正常	2021-08-19 12:00:00	2021-08-19 12:00:00	编辑参数 权限配置 网络探测 删除
kuantest	123456	连接失败	2021-08-19 12:00:00	2021-08-19 12:00:00	编辑参数 权限配置 网络探测 删除

连接参数

主机名: kuantest
主机号: 127.0.0.1
端口号: 3306
用户名: admin
密码: *****

test-mysql111

搜索

database1

table1

table2

table3

table4

table5

database1

database1

<

表详情

数据预览

名称	类型	长度	小数点	不是null	键
ID	int	255	2	<input checked="" type="checkbox"/>	
name	int	255		<input checked="" type="checkbox"/>	
order	varchar	10	2	<input type="checkbox"/>	
status	varchar	10		<input type="checkbox"/>	

结果表引用

请将以下SQL语句复制到SQL编辑器中运行

```
CREATE TABLE CHECK_CONSTRAINTS_sink (  
  'CONSTRAINT_SCHEMA' STRING,  
  'CONSTRAINT_NAME' STRING,  
  'CONSTRAINT_CATALOG' STRING,  
  'CHECK_CLAUSE' LONGSTRING  
) WITH (  
  'connector' = 'jdbc',  
  'url' = 'jdbc:mysql://10.201.0.215:33306/information_schema',  
  'table-name' = 'CHECK_CONSTRAINTS',  
  'username' = 'dlink-faas-ddl-config-9dd4b155-21f6-4181-b754-11d042ae05b5-user',  
  'password' = 'dlink-faas-ddl-config-9dd4b155-21f6-4181-b754-11d042ae05b5-password'  
)
```

复制

DTCC
2021



北京国际会议中心

2021/8/18-8/20

IT168.com

ChinaUnix.net

ITPUB

特性：数据开发



数/造/未/来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

数据开发能力：sql全流程开发

构建映射表

sql预览

调试sql

sql作业

```
Schema
+ @ almu
+ @ default
+ @ test1
+ @ test22

--2. 创建映射表
28 CREATE TABLE prod_orders_wide_pg
29 (
21 o_orderkey INT,
22 o_custkey INT,
23 o_orderstatus STRING,
24 o_totalprice DOUBLE,
25 o_currency STRING,
26 o_orderline TIMESTAMP(6),
27 o_orderpriority STRING,
28 o_clerk STRING,
29 o_shippriority INT,
30 o_comment STRING,
31 customer_name STRING,
32 n_nationkey INT,
33 n_nationname STRING,
34 rs_rate DOUBLE,
35 PRIMARY KEY (o_orderkey) NOT ENFORCED
36 ) WITH (
37 'connector' = 'jdbc',
38 'url' = 'jdbc:postgresql://10.201.8.124:55432/almu_project',
39 'username' = 'postgres',
40 'password' = '123456',
41 'table-name' = 'prod_orders_wide_pg',
42 'scan.fetch-size' = '5'
43 )
-- 2.1. 查询源表数据
46 select * from prod_orders_wide
```

Schema

sql-pg kafka-pg New Query 预览 kafka-pg-kafka X New Query New Query New Query +

Catalog default Database default

Schema

almu default test1 test22

--3.1 订单表-客户表

51 select * from prod_orders_kafka orders

52 left join prod_customer_pg customer on orders.o_custkey = customer.c_custkey where customer.c_custkey is not null

--3.2 订单表-客户表-国家表

53 select * from prod_orders_kafka orders

54 select * from prod_orders_kafka orders

正在运行

o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_currency	o_orderline	o_orderpriority	o_clerk	o_shippriority	o_comment	c_custkey	c_name
5024551	31693	O	58857.68	CNY	2021-06-2	HIGH	Clerk#00	0	Tiresia 31693	Customer	T8
5024577	33782	O	250877.4	CNY	2021-06-2	HIGH	Clerk#00	0	longside 33782	Customer	Inc
5024578	19433	P	281933.4	CAD	2021-06-3	MEDIUM	Clerk#00	0	ptotes c 19433	Customer	Rpg
5024579	58106	P	23810.3	USD	2021-06-1	URGENT	Clerk#00	0	unic reg 58106	Customer	Dal
5024580	82342	P	78463.84	USD	2021-06-1	URGENT	Clerk#00	0	thraab 82342	Customer	WdC
5024581	19087	P	275126.0	USD	2021-06-4	NOT SP	Clerk#00	0	ld reque 19087	Customer	XZ1
5024582	85387	P	101385.3	CAD	2021-06-3	MEDIUM	Clerk#00	0	careful 85387	Customer	AM0
5024583	45071	P	247367.0	CAD	2021-06-3	MEDIUM	Clerk#00	0	y silent 45071	Customer	805
5024610	13219	P	86778.85	JPY	2021-06-4	NOT SP	Clerk#00	0	ffily re 13219	Customer	hai
5024611	443	O	80542.69	GBP	2021-06-1	URGENT	Clerk#00	0	ar accou 443	Customer	0d5
5024612	90382	P	7892.44	NOB	2021-06-1	URGENT	Clerk#00	0	ages. fo 90382	Customer	E2z
5024614	85531	O	328103.9	JPY	2021-06-4	NOT SP	Clerk#00	0	ar fomes 85531	Customer	y51
5024615	43135	P	66849.3	CNY	2021-06-1	URGENT	Clerk#00	0	regular 43135	Customer	29C
5024640	13393	O	167674.2	NOB	2021-06-2	HIGH	Clerk#00	0	riously 13393	Customer	x1C

共有94条 执行时间: 00:11

```
Schema
+ @ almu
+ @ default
+ @ test1
+ @ test22

--3. 调试sql
51 --3.1 订单表-客户表
52 select * from prod_orders_kafka orders
53 --3.2 订单表-客户表-国家表
54 select * from prod_orders_kafka orders
55 left join prod_customer_pg customer on orders.o_custkey = customer.c_custkey
56 left join prod_nation_pg nation on customer.c_nationkey = nation.n_nationkey
57 where customer.c_custkey is not null
58
59 --3.3 订单表-客户表-国家表-汇率表
60 select * from prod_orders_kafka orders
61 left join prod_customer_pg customer on orders.o_custkey = customer.c_custkey
62 left join prod_nation_pg nation on customer.c_nationkey = nation.n_nationkey
63 left join rate_pg rate on nation.n_name = rate.rs_symbol
64 where customer.c_custkey is not null
65
66 --3.4 只查前5条数据
67 select orders.o_nationkey as n_nationkey,
68 nation.n_name as nation_name, rate.rs_rate as rs_rate, custid
69 from prod_orders_kafka orders
70 left join prod_customer_pg customer on orders.o_custkey =
71 left join prod_nation_pg nation on customer.c_nationkey =
72 left join rate_pg rate on nation.n_name = rate.rs_symbol
73 where customer.c_custkey is not null
```

kafka-pg-pg(实... RUNNING

作业实例 运行事件 作业快照 启动日志 Job Manager Task Manager 监控告警 配置信息 Flink UI 指标

当前Job实例: f25705bb382da0c0073365dab3b6115

实例状态	Shuffled/Total	重试次数	启动时间	持续时间	操作
RUNNING	1/1	0	2021-08-16 19:50:12	3天14时35分45秒	Flink UI 指标 版本对比

Source: TableSourceScan[table=([default, default, prod_orders_kafka], ...]

名称	实例状态	Received(Bytes/Records)	Send(Bytes/Records)	并行度	Tasks
Source: TableSourceScan[table=([default, default, prod_orders_kafka], ...]	RUNNING	0B / 0	43.56MB / 243859	1	1

DTCC
2021



北京国际会议中心

2021/8/18-8/20

IT168.com

ChinaUnix.net

ITPUB

特性：数据科学



数据科学能力：

数据模型

在线机器学习

数据挖掘

模型中心 / 模型管理

创建模型

请输入模型名称或ID搜索

搜索

重置

模型ID	模型名称	框架名称	导入状态	模型来源	发布状态	模型描述	更新时间	操作
306	用户中心模型	tensorflow	等待中	本地上传	未发布	用户中心1.0版本		删除
305	计费中心模型	tensorflow	导入成功	本地上传	未发布			
304	成本中心模型	tensorflow	导入成功	本地上传	未发布			

模型开发 / 自动学习

图片 文本

图片分类

☒ 苹果
☐ 梨
☐ 西红柿

识别图片的物体类型/状态/场景等

目标检测

☒ 苹果 ☒ 梨

识别图片中有什么物体，以及其所在的位置

图像分割

☒ 苹果 ☒ 梨

提取图片中每个目标的轮廓，并识别目标数量、类型、位置等信息

OCR

对图片中的印刷文字进行定位与识别，如银行卡号，发票文字识别等

创建实例

请输入实例名称搜索

搜索

重置

序号	实例编号	实例名称	实例类型	服务类型	训练编号	当前版本	训练状态	创建时间	操作
----	------	------	------	------	------	------	------	------	----

请输入模型名称

搜索

全部 我发布的

模型类型
☐ 图片
☐ 文本

行业
☐ 制造业

水果分类测试

模型本地

猫狗分类V1
暂无
发 布 人：zhaosinbo
发布时间：2021-06-03

猫狗分类大数据量
暂无
发 布 人：tfg
发布时间：2021-05-26

lmt_data_0515
暂无
发 布 人：183***430
发布时间：2021-05-17

共 7 个实例 10个/页 < 1



北京国际会议中心

2021/8/18-8/20





数 / 造 / 未 / 来
第十二届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2021

谢谢

DTCC
2021



北京国际会议中心

🕒 2021/8/18-8/20



ChinaUnix^{.net}

ITPUB