Hive

添加字段一次增加一个列(默认添加为最后一列)

ALTER TABLE table\_name ADD COLUMNS (new\_col INT);

可以一次增加多个列

ALTER TABLE table\_name ADD COLUMNS (c1 INT,c2 STRING);

CREATE TABLE table\_name (

column1\_name column1\_datatype,

column2\_name column2\_datatype,

...

) [COMMENT table\_comment]

[ROW FORMAT row\_format]

[STORED AS file\_format]

[LOCATION hdfs\_path];

table\_name: 你希望创建的Hive表的名称。

column1\_name, column2\_name, ...: 表中的列名。

column1\_datatype, column2\_datatype, ...: 每列的数据类型，例如STRING, INT, FLOAT, DOUBLE, DATE等。

COMMENT: 可选的表注释，用于描述表的内容或目的。

ROW FORMAT: 指定行的格式，例如DELIMITED（默认），SERDE等。

STORED AS: 指定文件存储格式，例如TEXTFILE（默认），ORC, PARQUET等。

LOCATION: 指定HDFS上表数据存储的路径。如果省略，Hive会将数据存储在默认的仓库位置。

行格式（ROW FORMAT）:

DELIMITED: 指定字段分隔符，常用的字段分隔符包括逗号,、制表符\t、空格 等。

FIELDS TERMINATED BY: 后跟一个字符，表示字段的分隔符。

COLLECTION ITEMS TERMINATED BY: 用于集合类型的字段，指定集合中元素的分隔符。

MAP KEYS TERMINATED BY: 用于映射类型的字段，指定键值对的分隔符。

SERDE: 指定一个特定的序列化/反序列化（SerDe）类，用于处理复杂的数据格式。

存储格式（STORED AS）:

TEXTFILE: 默认的存储格式，存储为纯文本文件。

ORC: 优化的行列存储格式，适合于大数据处理，提供高效的压缩和性能。

PARQUET: 另一种列存储格式，支持更复杂的数据类型和高级压缩。

位置（LOCATION）:

指定HDFS上表数据的存储路径。如果表是外部表，那么LOCATION是必需的。

CREATE TABLE sales\_data (

product\_name STRING,

units\_sold INT

) COMMENT 'This table stores sales data'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hive/warehouse/sales\_data';

在这个例子中，

创建了一个名为sales\_data的表，

它有两个列：product\_name和units\_sold。

表的注释说明了表的用途。

数据行由逗号分隔，

存储为文本文件，

并放置在指定的HDFS路径下。

动态分区插入

LOAD DATA INPATH '/path/to/sales\_data\_with\_year\_month.csv' INTO TABLE sales\_data PARTITION (year, month);

使用一个CSV文件来插入数据，确保CSV文件中包含year和month列，并且列名与表定义中的分区列名称一致

静态

LOAD DATA INPATH '/path/to/your/data/file' INTO TABLE table\_name PARTITION (year=2024, month=04);