

1

1. for xml raw 生成简单的xml数据结构

for xml auto 生成复杂的xml嵌套数据结构

```
2. 1 <row tid="1" tname="张一" sid="101" sname="李思" />
    2 <row tid="1" tname="张一" sid="102" sname="王学" />
    3 <row tid="2" tname="赵二" sid="103" sname="孙行" />
```

```
1 <teacher tid="1" tname="张一">
2   <student sid="101" sname="李思" />
3   <student sid="102" sname="王学" />
4 </teacher>
5 <teacher tid="2" tname="赵二">
6   <student sid="103" sname="孙行" />
7 </teacher>
```

2

1. MapReduce: MapReduce 是 Hadoop 的经典并行计算框架, Hive 在早期版本中主要使用 MapReduce 进行查询执行。在这种方式下, Hive 将 SQL 查询转换为一系列的 Map 和 Reduce 任务, 并将这些任务提交给 Hadoop 的 MapReduce 框架进行执行。Map 任务负责数据的分片和并行处理, Reduce 任务负责数据的聚合和汇总。Hive 在将查询转换为 MapReduce 任务时会利用 Hadoop 的分布式文件系统 (HDFS) 来管理数据的存储和读取。
2. Tez: Tez 是一种基于 DAG (有向无环图) 的并行计算框架, 用于替代 MapReduce 的底层引擎。Hive 的较新版本引入了 Tez 作为执行引擎, 以提供更高的查询性能。在这种方式下, Hive 将 SQL 查询转换为一个 DAG, 其中每个节点代表一个操作 (如扫描、过滤、聚合等), 并且通过边连接起来定义数据流。然后, Hive 将该 DAG 提交给 Tez 框架进行执行。Tez 可以更好地利用内存和计算资源, 通过优化数据流和任务调度来提高查询的执行效率

3

- 建立决策树模型
- 创建决策树模型存储表, 存储生成的二进制对象
- 保存存储表, 存储决策树模型
- 预测输入数据

4

```
1. 1 INSERT INTO R1 (sno, cno, eval)
    2 VALUES (1, 3, JSON_OBJECT('evaluation', JSON_ARRAY(
    3   JSON_OBJECT('stage', 1, 'result', 'ok'),
    4   JSON_OBJECT('stage', 2, 'result', 'good'),
    5   JSON_OBJECT('stage', 3, 'result', 'excellent')
    6 )))
```

2. 1 SELECT sno, cno, JSON_EXTRACT(eval, CONCAT('\$.evaluation[' , stage_index,
2 '].stage')) AS stage,
3 JSON_EXTRACT(eval, CONCAT('\$.evaluation[' , stage_index,
4 '].result')) AS result
5 FROM R1
6 CROSS JOIN
7 (
8 SELECT 0 AS stage_index UNION ALL
9 SELECT 1 AS stage_index
10) stages
11 WHERE sno = 1 AND cno = 3;