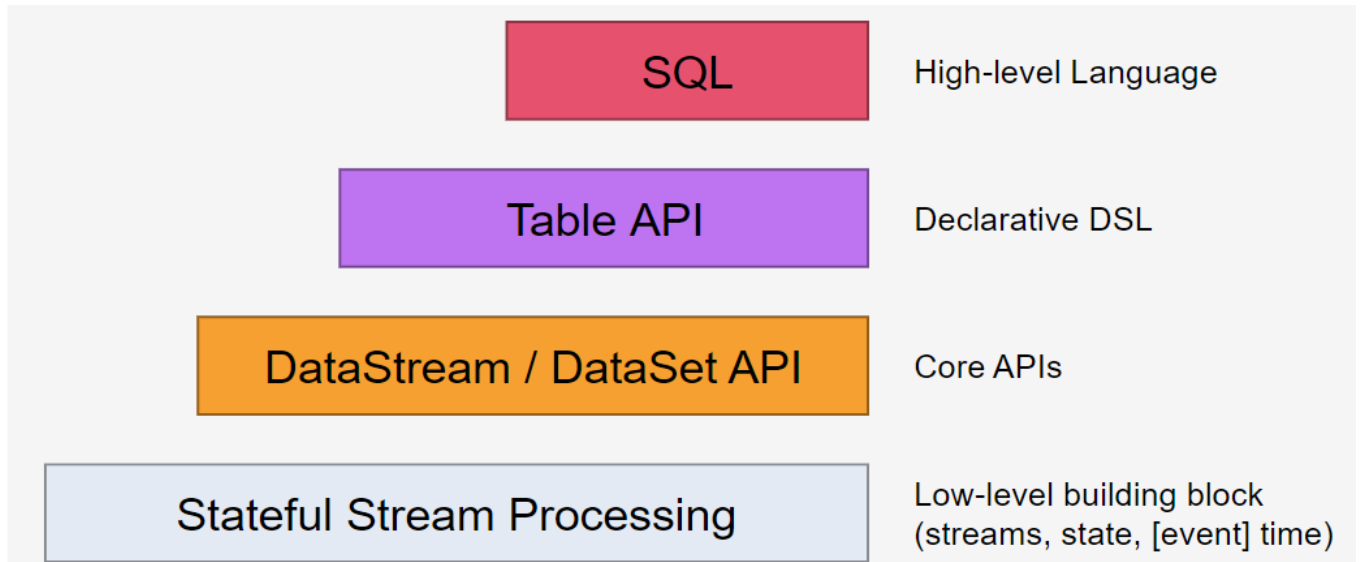


Flink SQL项目实录

一、Flink SQL层级

为Flink最高层的API，易于使用，所以应用更加广泛，eg. ETL、统计分析、实时报表、实时风控等。

Flink SQL所处的层级：



二、Flink聚合：

1、Window Aggregate

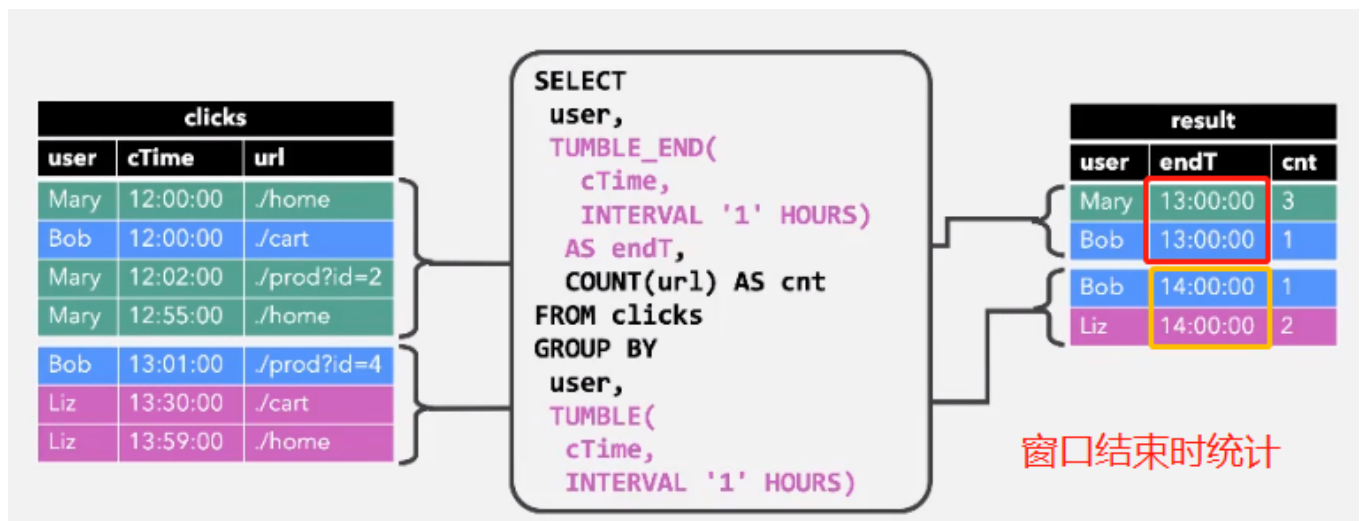
内置了三种常用的窗口：

`TUMBLE(time, INTERVAL '5' SECOND);` //类似于flink 中间层 DataStream API 中 window中的滚动窗口

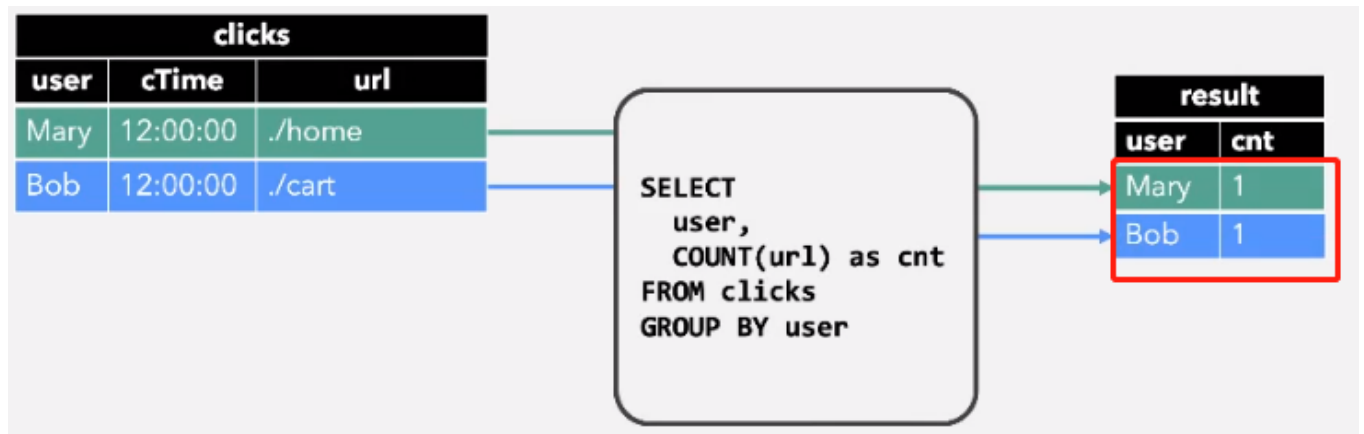
`HOP(time, INTERVAL '10' SECOND, INTERVAL '5' SECOND);` //类似于flink 中间层 DataStream API 中 window的滑动窗口，每10秒中统计最近5秒的数据

`SESSION(time, INTERVAL '5' SECOND)`

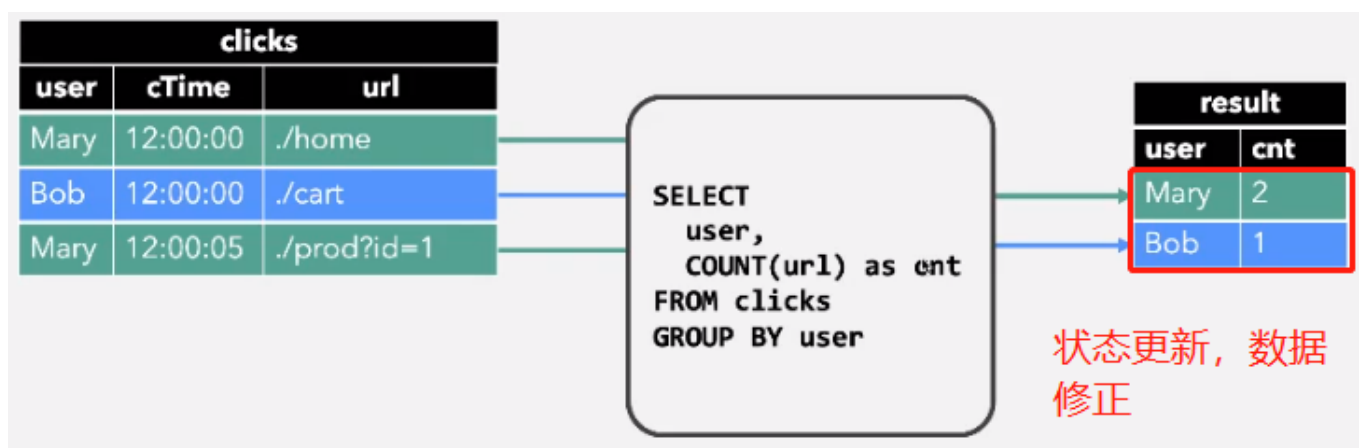
time有两种格式的时间，一种是proctime也就是系统时间， 另一种是rowtime。



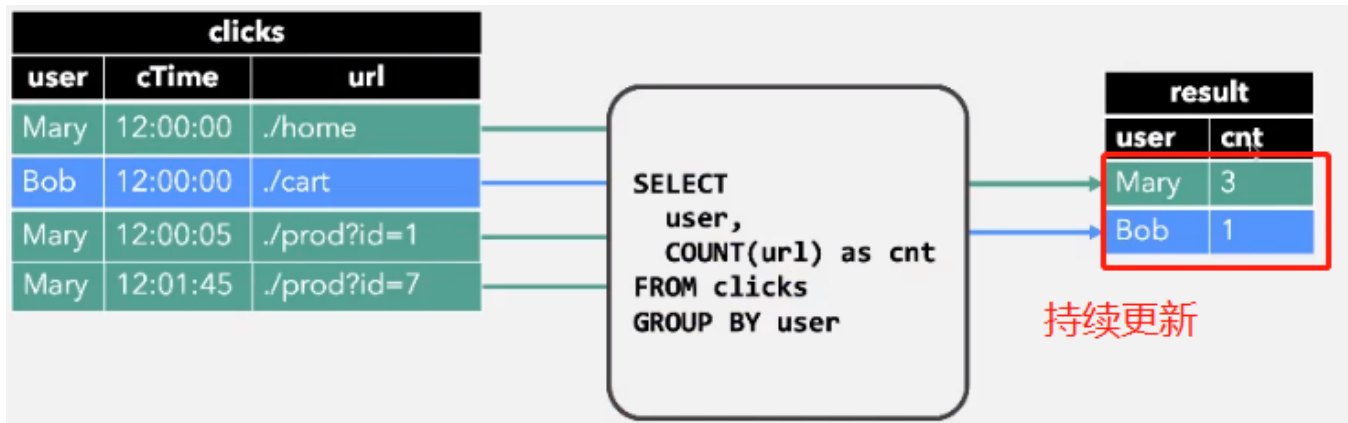
2、Group Aggregate



继续加入数据时:



继续进入数据:



结果是一个不断更新的过程。

Window Aggregate 与 Group Aggregate 的区别

1)、Window Aggregate 与 Group Aggregate 是有一些明显的区别的。其主要的区别是，**Window Aggregate** 是当window结束时才输出，其输出的结果是最终值，不会再进行修改，其输出流是一个 Append 流。

而 Group Aggregate 是每处理一条数据，就输出最新的结果，其结果是在不断更新的，就好像数据库中的数据一样，其输出流是一个 Update 流。

2)、另外一个区别是，window Aggregate 由于有 watermark，可以精确知道哪些窗口已经过期了，所以可以及时清理过期状态，保证状态维持在稳定的大小。

而 Group Aggregate 因为不知道哪些数据是过期的，所以状态会无限增长，这对于生产作业来说不是很稳定，所以建议对 Group Aggregate 的作业配上 State TTL 的配置。

对比图：

	Window Aggregation	Group Aggregation
输出模式	按时输出	提前输出
输出量	只输出一次结果	Per key 输出N个结果 (Sink 压力)
输出流	Append Stream	Update Stream
状态清理	及时清理过期数据	状态无限增长 !
Sink	均可	可更新的结果表 (Hbase, MySQL)

精确性 ↔ 状态大小

State TTL Config

```
streamQueryConfig
.withIdleStateRetentionTime(Time.day(1), Time.day(2))
```

项目代码设置:

```
tEnv.getConfig().setIdleStateRetentionTime(org.apache.flink.api.common.time.Time.minutes(1),org
.apache.flink.api.common.time.Time.minutes(10));
```