

Flink sql

```
1 -- 执行sql查询      滚动窗口 10秒      计算10秒窗口内用户点击次数
2 SELECT TUMBLE_END(proctime, INTERVAL '10' SECOND) as processtime,
3     userId,count(*) as pvcount
4 FROM Users
5 GROUP BY TUMBLE(proctime, INTERVAL '10' SECOND), userId
```

```
1 -- 数据库RDS结果表
2 CREATE TABLE source_ods_fact_log_track_action (
3     account_id VARCHAR, --用户ID
4     client_ip VARCHAR, --客户端IP
5     client_info VARCHAR, --设备机型信息
6     `action` VARCHAR, --页面跳转描述
7     gpm VARCHAR, --埋点链路
8     c_time BIGINT, --请求时间
9     udata VARCHAR, --扩展信息, JSON格式
10    `position` VARCHAR, --位置信息
11    network VARCHAR, --网络使用情况
12    p_dt VARCHAR --时间分区天
13 ) WITH (
14     'connector.type' = 'kafka',
15     'connector.version' = 'universal',
16     'connector.topic' = 'topic_uv',
17     'update-mode' = 'append',
18     'connector.properties.zookeeper.connect' = '172.24.103.8:2181',
19     'connector.properties.bootstrap.servers' = '172.24.103.8:9092',
20     'connector.startup-mode' = 'latest-offset',
21     'format.type' = 'json'
22 );
23
24 CREATE TABLE result_cps_total_summary_pvuv_min (
25     summary_date varchar, --统计日期
26     summary_min varchar, --统计分钟
27     pv bigint, --点击量
28     uv bigint, --一天内同个访客多次访问仅计算一个UV
29     current_times varchar --当前时间
30 -- primary key (summary_date, summary_min)
31 ) WITH (
32     'connector.type' = 'jdbc',
33     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
34     'connector.table' = 'ali_pvuv',
35     'connector.username' = 'root',
36     'connector.password' = '123456'
37     'connector.write.flush.max-rows' = '10',
38     'connector.write.flush.interval' = '5s'
39 );
40
41
42 select p_dt as summary_date, --时间分区
43     count (client_ip) as pv, --客户端的IP
```

```

44         count (distinct client_ip) as uv, --客户端去重
45         cast (max (FROM_UNIXTIME(c_time)) as TIMESTAMP) as c_time --请求的时间
46 from source_ods_fact_log_track_action
47 group by p_dt
48
49 INSERT into result_cps_total_summary_pvuv_min
50 select a.summary_date, --时间分区
51        cast (DATE_FORMAT (c_time, 'HH:mm') as varchar) as summary_min, --取出小时分钟级别的时间
52        a.pv,
53        a.uv,
54        cast (LOCALTIMESTAMP as varchar) as current_times --当前时间
55 from result_cps_total_summary_pvuv_min_01 AS a

```

```

1 -- 实时计算当天UV指标sql, 这里使用最简单的group by agg, 没有使用minibatch或窗口, 在大数据量优化时最好使用
  后两种方式
2 insert into uv_index select log_date, ROW(count(distinct mid) as UV) from pageview group by
  log_date
3

```

```

1 use catalog my_hive;
2 -- build streaming database and tables;
3 create database stream_db;
4 use stream_db;
5 create table order_table (
6     id long,
7     amount double,
8     user_id long,
9     status string,
10    ts timestamp,
11    ... -- 可能还有几十个字段
12    ts_day string,
13    ts_hour string
14 ) with (
15     'connector.type' = 'kafka',
16     ... -- Kafka table相关配置
17 );
18 -- build batch database and tables;
19 create database batch_db;
20 use batch_db;
21 create table order_table like stream_db.order_table (excluding options)
22 partitioned by (ts_day, ts_hour)
23 with (
24     'connector.type' = 'hive',
25     ... -- Hive table相关配置
26 );
27
28 insert into [stream_db.|batch_db.]order_table select ... from log_table;
29
30

```

```

1 -- stream 维表
2 use stream_db;
3 create table user_info (
4     user_id long,
5     age int,
6     address,
7     primary key(user_id)

```

```

8 ) with (
9     'connector.type' = 'jdbc',
10    ...
11 );
12
13 -- 将离线数仓的维表导入实时数仓中
14 insert into user_info select * from batch_db.user_info;
15
16 -- 维表Join, SQL批流复用
17 insert into order_with_user_age select * from order_table join user_info for system_time as
18 of order_table.proctime on user_info.user_id = user_info.user_id;
19
20 select age, avg(amount) from order_with_user_age group by age;
21
22 -- batch: 计算完成后, 一次性输出到mysql中, 同key只有一个数据
23 -- streaming: mysql里面的数据不断更新, 不断变化
24 insert into mysql_table select age, avg(amount) from order_with_user_age group by age;
25 -- batch: 同key只有一个数据, append即可
26 insert into hive_table select age, avg(amount) from order_with_user_age group by age;
27 -- streaming: kafka里面的数据不断append, 并且多出一列, 来表示这是upsert的消息, 后续的Flink消费会自动做出
28 机制来处理upsert
29 insert into kafka_table select age, avg(amount) from order_with_user_age group by age;
30
31 CREATE TABLE app_heartbeat_stream_source (
32     `ip` VARCHAR,
33     agent VARCHAR,
34     roomid VARCHAR,
35     userid VARCHAR,
36     abytes VARCHAR,
37     afcnt VARCHAR,
38     adrop VARCHAR,
39     afts VARCHAR,
40     alat VARCHAR,
41     vbytes VARCHAR,
42     vfcnt VARCHAR,
43     vdrop VARCHAR,
44     vfts VARCHAR,
45     vlat VARCHAR,
46     ublock VARCHAR,
47     dblock VARCHAR,
48     region VARCHAR,
49     stamp VARCHAR,
50     app_ts AS TO_TIMESTAMP(stamp), --定义生成WATERMARK的字段,
51     WATERMARK FOR app_ts AS app_ts - INTERVAL '10' SECOND --WATERMARK比数据时间线性增加10S
52 ) WITH (
53     'connector.type' = 'kafka',
54     'connector.version' = 'universal',
55     'connector.topic' = 'topic_live',
56     'update-mode' = 'append',
57     'connector.properties.zookeeper.connect' = '172.24.103.8:2181',
58     'connector.properties.bootstrap.servers' = '172.24.103.8:9092',
59     'connector.startup-mode' = 'latest-offset',
60     'format.type' = 'json'
61 )
62
63 SELECT
64     ip,

```

```

35 agent,
36 CAST(roomid AS BIGINT) as roomid,
37 CAST(userid AS BIGINT) as userid,
38 CAST(abytes AS BIGINT) as abytes,
39 CAST(afcmt AS BIGINT) as afcmt,
40 CAST(adrop AS BIGINT) as adrop,
41 unix_timestamp(afts) as afts,
42 CAST(alat AS BIGINT) as alat,
43 CAST(vbytes AS BIGINT) as vbytes,
44 CAST(vfcmt AS BIGINT) as vfcmt,
45 CAST(vdrop AS BIGINT) as vdrop,
46 unix_timestamp(vfts) as vfts,
47 CAST(vlat AS BIGINT) as vlat,
48 CAST(ublock AS BIGINT) as ublock,
49 CAST(dblock AS BIGINT) as dblock,
50 app_ts,
51 region
52 FROM
53 app_heartbeat_stream_source
54
55
56 SELECT
57 CAST(TUMBLE_START(app_ts, INTERVAL '1' MINUTE) as VARCHAR) as app_ts,
58 roomid,
59 SUM(ublock) as ublock,
60 SUM(dblock) as dblock,
61 SUM(adrop) as adrop,
62 SUM(vdrop) as vdrop,
63 SUM(alat) as alat,
64 SUM(vlat) as vlat
65 FROM
66 view_app_heartbeat_stream
67 GROUP BY
68 TUMBLE(app_ts,INTERVAL '1' MINUTE),roomid
69
70
71 CREATE TABLE output_1 (
72 app_ts          VARCHAR,
73 roomid          BIGINT,
74 ublock          BIGINT,
75 dblock          BIGINT,
76 adrop           BIGINT,
77 vdrop           BIGINT,
78 alat            BIGINT,
79 vlat            BIGINT
80 --不支持 PRIMARY KEY (roomid)
81 ) WITH (
82 'connector.type' = 'jdbc',
83 'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
84 'connector.table' = 'live_output_1',
85 'connector.username' = 'root',
86 'connector.password' = '123456',
87 'connector.write.flush.max-rows' = '10',
88 'connector.write.flush.interval' = '5s'
89 )
90

```

```

91
92
93 CREATE TABLE output_1 (
94     app_ts          VARCHAR,
95     roomid          BIGINT,
96     ublock          BIGINT,
97     dblock          BIGINT,
98     adrop           BIGINT,
99     vdrop           BIGINT,
100    alat            BIGINT,
101    vlat            BIGINT
102    --不支持 PRIMARY KEY (roomid)
103 ) WITH (
104     'connector.type' = 'jdbc',
105     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
106     'connector.table' = 'live_output_1',
107     'connector.username' = 'root',
108     'connector.password' = '123456',
109     'connector.write.flush.max-rows' = '10',
110     'connector.write.flush.interval' = '5s'
111 )
112
113
114
115 INSERT INTO output_1
116 SELECT * FROM room_error_statistics_10min
117
118
119 SELECT
120     CAST(TUMBLE_START(app_ts, INTERVAL '1' MINUTE) as VARCHAR) as app_ts,
121     region,
122     SUM(alat)/COUNT(alat) as alat,
123     SUM(vlat)/COUNT(vlat) as vlat
124 FROM
125     view_app_heartbeat_stream
126 GROUP BY
127     TUMBLE(app_ts, INTERVAL '1' MINUTE), region
128
129
130 CREATE TABLE output_2 (
131     app_ts VARCHAR,
132     region VARCHAR,
133     alat DOUBLE,
134     vlat DOUBLE
135 ) WITH (
136     'connector.type' = 'jdbc',
137     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
138     'connector.table' = 'live_output_2',
139     'connector.username' = 'root',
140     'connector.password' = '123456',
141     'connector.write.flush.max-rows' = '10',
142     'connector.write.flush.interval' = '5s'
143 )
144
145
146 INSERT INTO output_2

```

```

147 SELECT * FROM region_lat_statistics_10min
148
149
150 SELECT
151     CAST(TUMBLE_START(app_ts, INTERVAL '1' MINUTE) as VARCHAR) as app_ts,
152     SUM(IF(ublock <> 0 OR dblock <> 0, 1, 0)) / CAST(COUNT(DISTINCT userid) AS DOUBLE) as
    block_rate
153 FROM
154     view_app_heartbeat_stream
155 GROUP BY
156     TUMBLE(app_ts, INTERVAL '1' MINUTE)
157
158
159 CREATE TABLE output_3 (
160     app_ts          VARCHAR,
161     block_rate      DOUBLE
162 ) WITH (
163     'connector.type' = 'jdbc',
164     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
165     'connector.table' = 'live_output_3',
166     'connector.username' = 'root',
167     'connector.password' = '123456',
168     'connector.write.flush.max-rows' = '10',
169     'connector.write.flush.interval' = '5s'
170 )
171
172
173 INSERT INTO output_3
174 SELECT * FROM block_total_statistics_10min
175
176
177 SELECT
178     CAST(TUMBLE_START(app_ts, INTERVAL '1' MINUTE) as VARCHAR) as app_ts,
179     SUM(ublock+dblock) / CAST(COUNT(DISTINCT userid) AS DOUBLE) as block_peruser
180 FROM
181     view_app_heartbeat_stream
182 GROUP BY
183     TUMBLE(app_ts, INTERVAL '1' MINUTE)
184
185
186 CREATE TABLE output_4 (
187     app_ts          VARCHAR,
188     block_peruser    DOUBLE
189 ) WITH (
190     'connector.type' = 'jdbc',
191     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
192     'connector.table' = 'live_output_4',
193     'connector.username' = 'root',
194     'connector.password' = '123456',
195     'connector.write.flush.max-rows' = '10',
196     'connector.write.flush.interval' = '5s'
197 )
198
199
200 INSERT INTO output_4
201 SELECT * FROM block_peruser_statistics_10min

```

```

202
203
204 -- TopN
205 SELECT
206     CAST(TUMBLE_START(app_ts, INTERVAL '60' SECOND) as VARCHAR) as app_ts,
207     roomid as room_id,
208     COUNT(DISTINCT userid) as app_room_user_cnt
209 FROM
210     view_app_heartbeat_stream
211 GROUP BY
212     TUMBLE(app_ts, INTERVAL '60' SECOND), roomid
213
214
215 SELECT
216     app_ts,
217     room_id,
218     app_room_user_cnt,
219     ranking
220     --PRIMARY KEY (app_ts,room_id,ranking)
221 FROM
222 (
223     SELECT
224         app_ts,
225         room_id,
226         app_room_user_cnt,
227         ROW_NUMBER() OVER (PARTITION BY 1 ORDER BY app_room_user_cnt desc) AS ranking
228     FROM
229         view_app_room_visit_1min
230 ) WHERE ranking <= 10
231
232
233
234 CREATE TABLE output_7 (
235     app_ts          VARCHAR,
236     roomid          BIGINT,
237     app_room_user_cnt BIGINT,
238     ranking         BIGINT
239 ) WITH (
240     'connector.type' = 'jdbc',
241     'connector.url' = 'jdbc:mysql://172.24.103.3:3306/flink',
242     'connector.table' = 'live_output_7',
243     'connector.username' = 'root',
244     'connector.password' = '123456',
245     'connector.write.flush.max-rows' = '10',
246     'connector.write.flush.interval' = '5s'
247 )
248
249
250
251 INSERT INTO output_7
252 SELECT * FROM view_app_room_visit_top10
253
254
255

```

