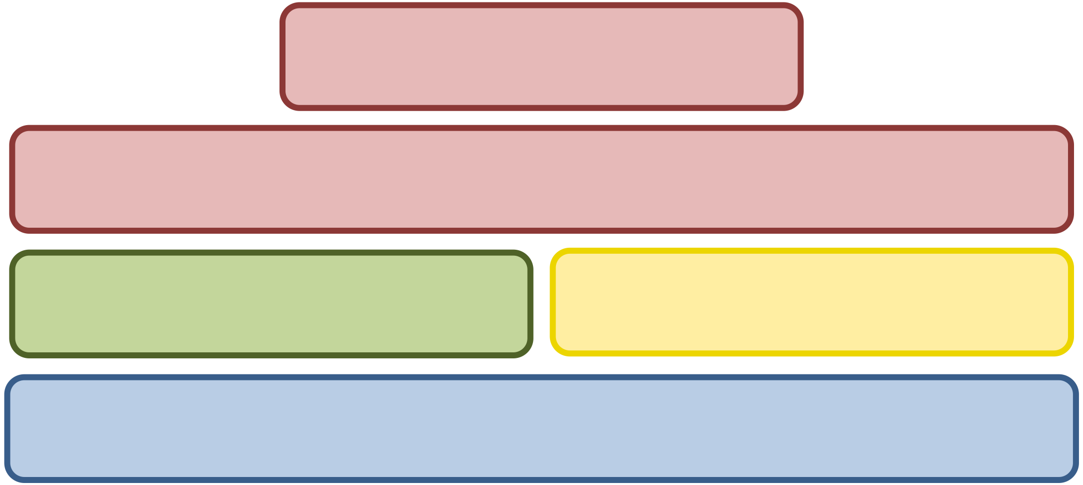
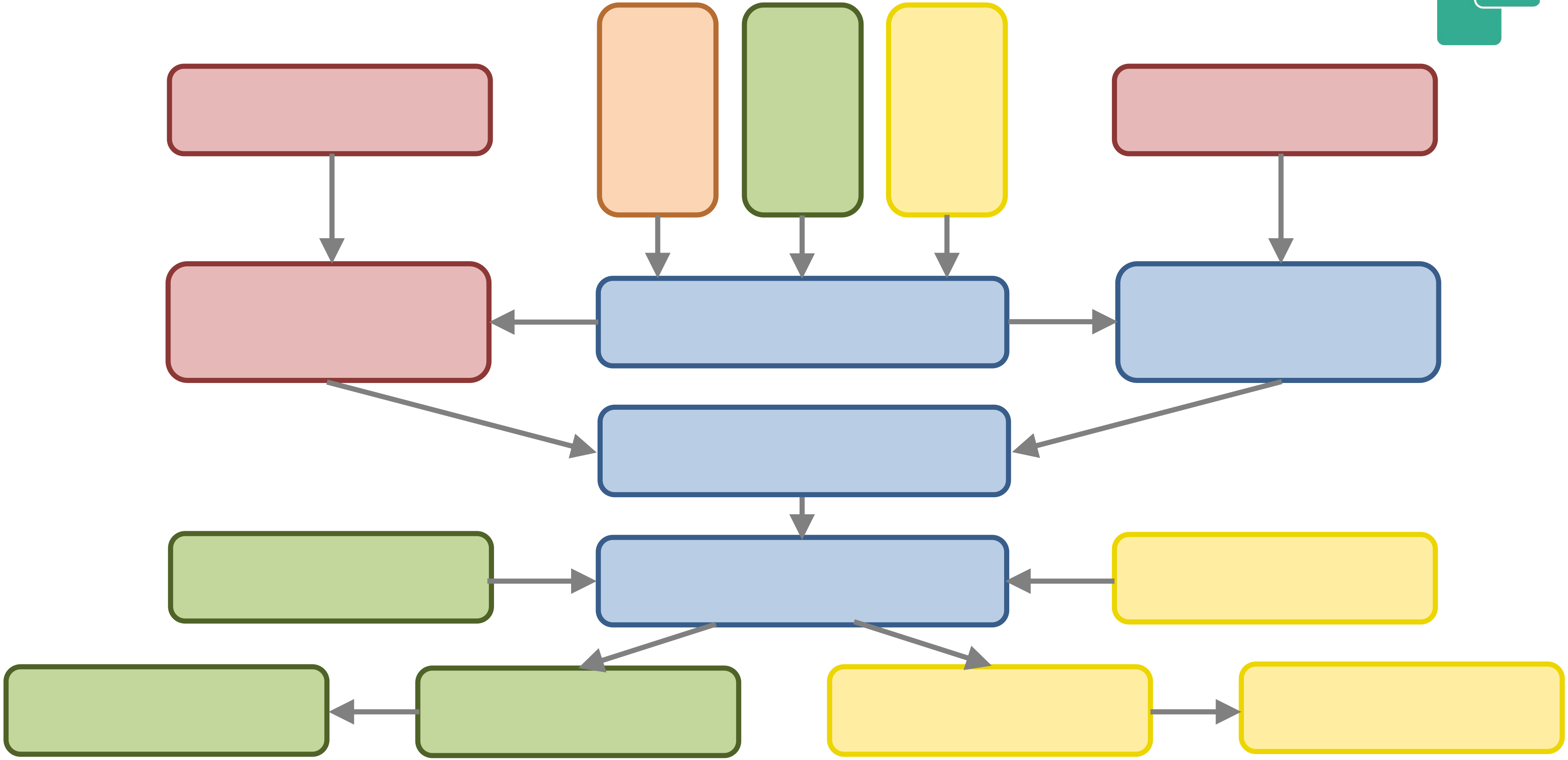
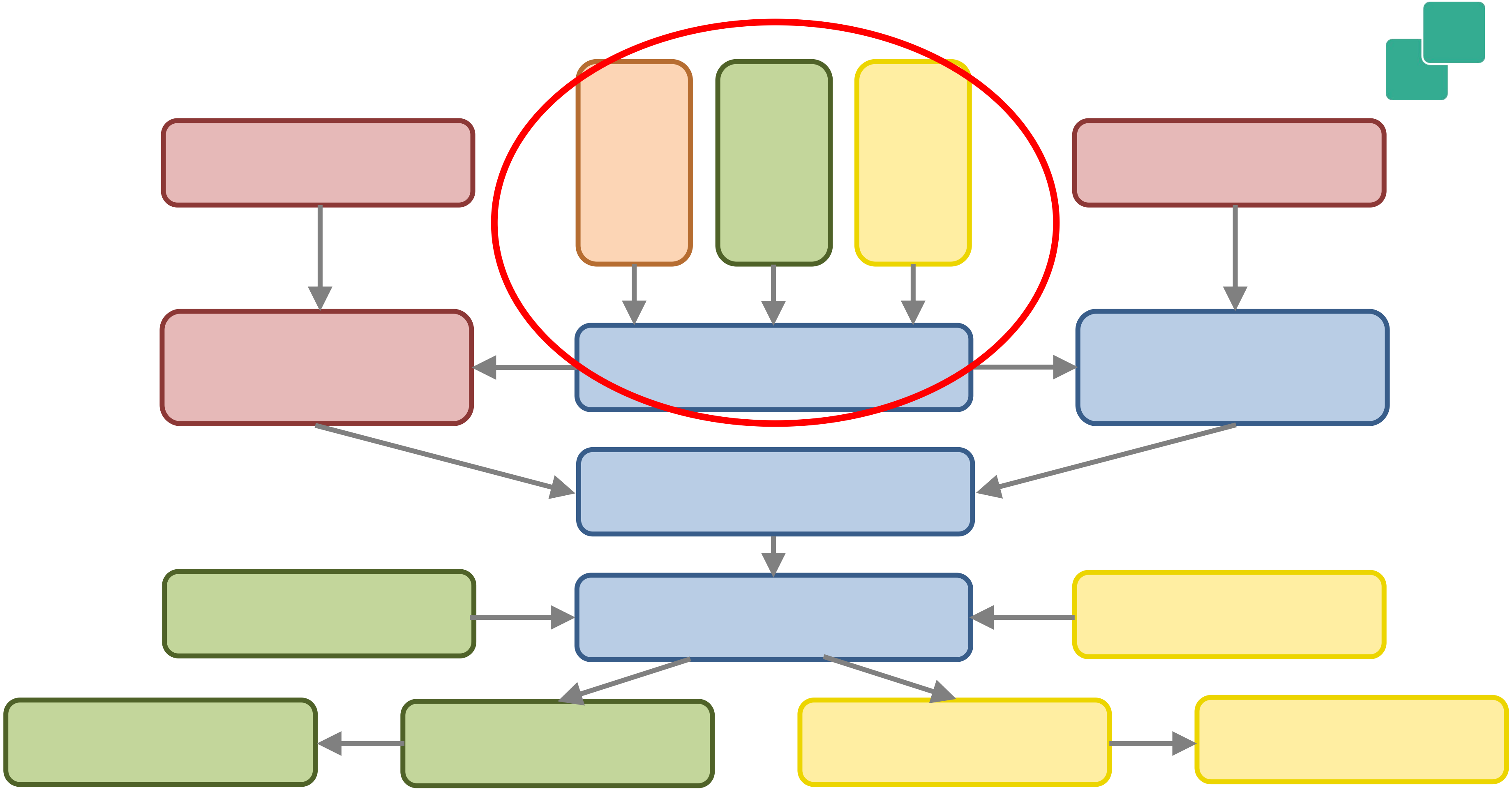
dataArtisans

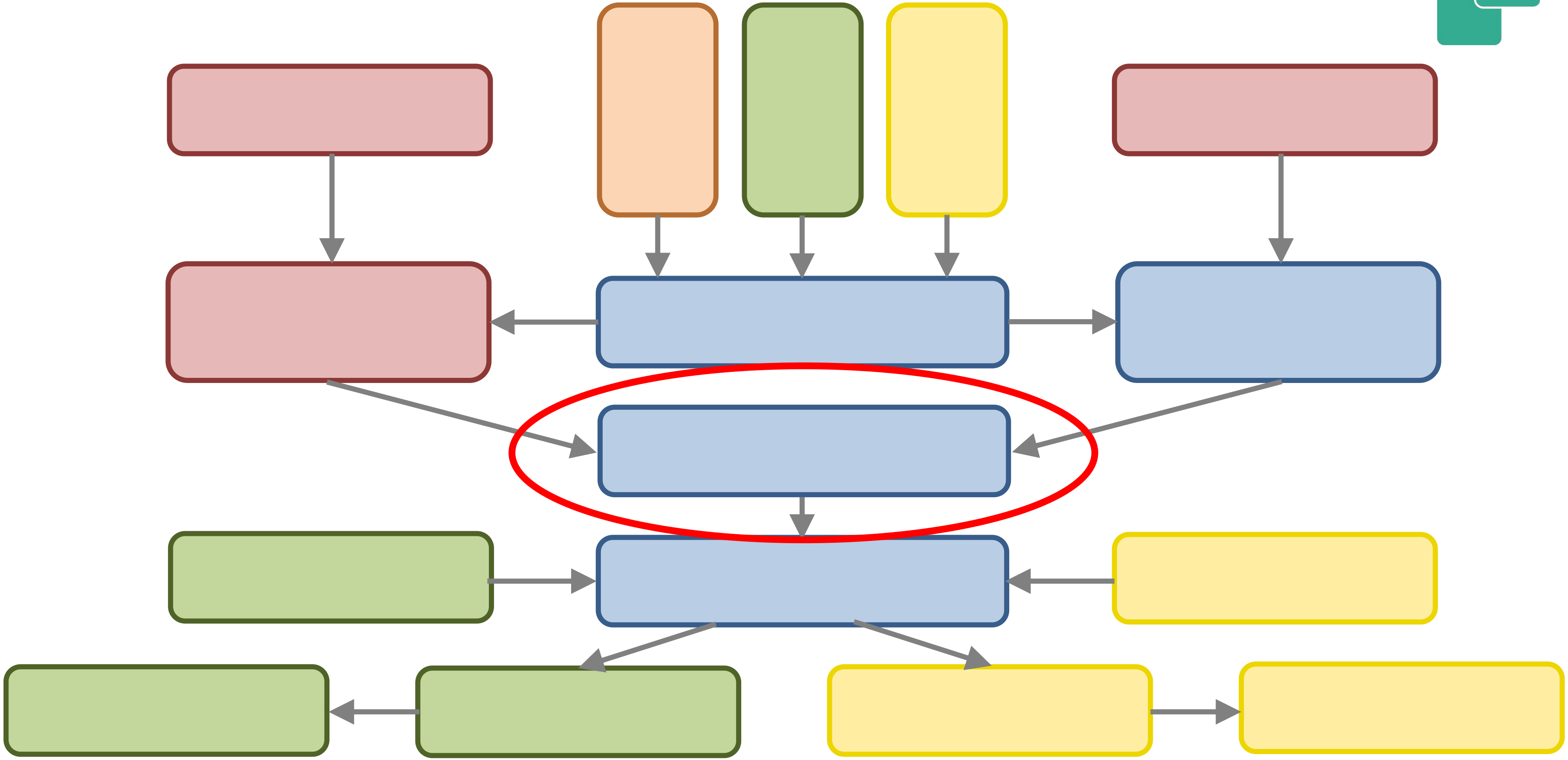


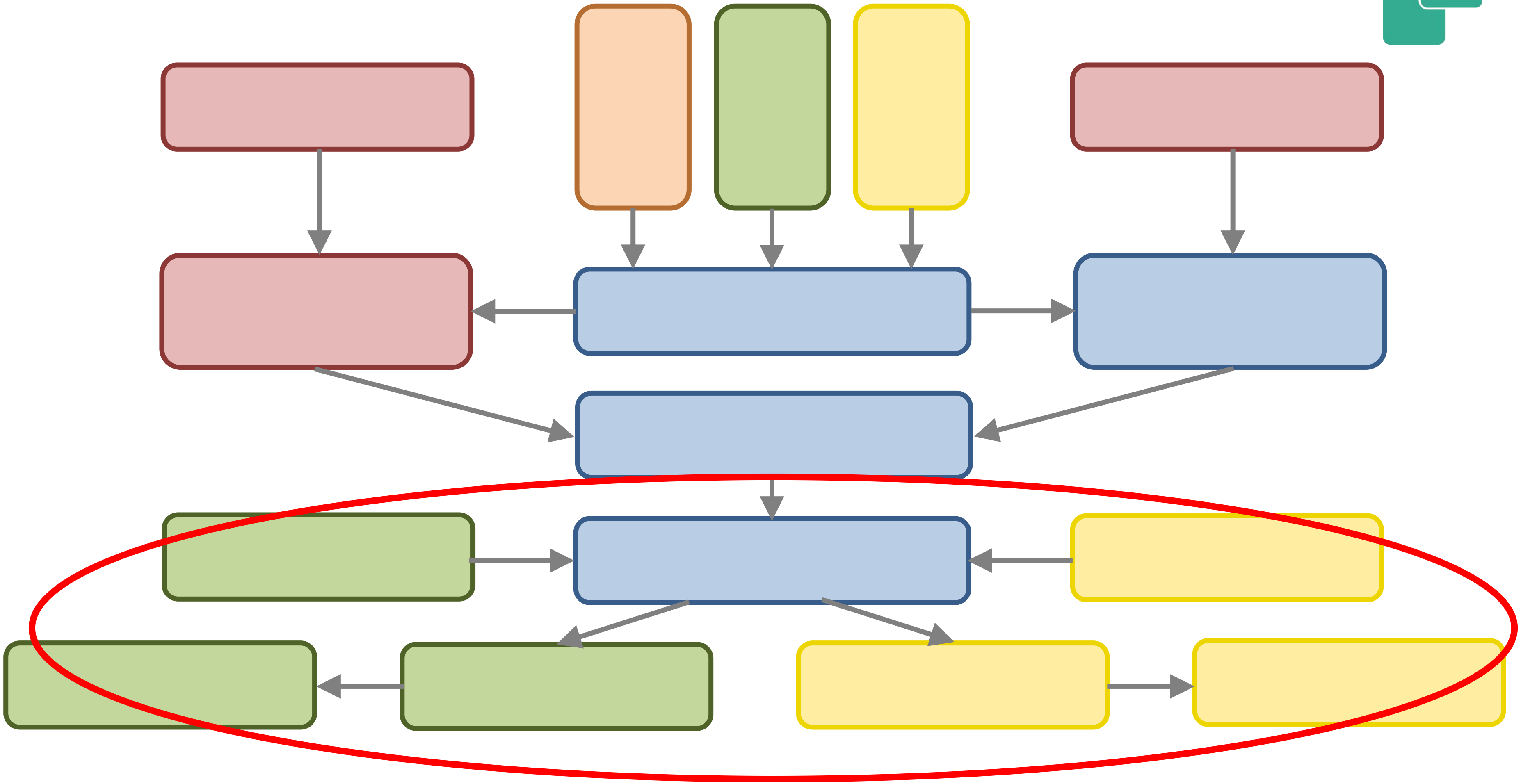
```
val tEnv = TableEnvironment.getTableEnvironment(env)

// configure your data source
val customerSource = CsvTableSource.builder()
  .path("/path/to/customer_data.csv")
  .field("name", Types.STRING).field("prefs", Types.STRING)
  .build()

// register as a table
tEnv.registerTableSource(”cust", customerSource)

// define your table program
val table = tEnv.scan("cust").select('name.lowerCase(), myParser('prefs))
val table = tEnv.sql("SELECT LOWER(name), myParser(prefs) FROM cust")

// convert
val ds: DataStream[Customer] = table.toDataStream[Customer]
val sensorData: DataStream[(String, Long, Double)] = ???

// convert DataStream into Table
val sensorTable: Table = sensorData
  .toTable(tableEnv, 'location, 'rowtime, 'tempF)

// define query on Table
val avgTempCTable: Table = sensorTable
  .window(Tumble over 1.day on 'rowtime as 'w)
  .groupBy('location, ’w)
  .select('w.start as 'day,
    'location,
    (('tempF.avg - 32) * 0.556) as 'avgTempC)
  .where('location like "room%")
```

```
val sensorData: DataStream[(String, Long, Double)] = ???

// register DataStream
tableEnv.registerDataStream(
  "sensorData", sensorData, 'location, 'rowtime, 'tempF)

// query registered Table
val avgTempCTable: Table = tableEnv.sql("""
    SELECT TUMBLE_START(TUMBLE(time, INTERVAL '1' DAY) AS day,
         location,
         AVG((tempF − 32) ∗ 0.556) AS avgTempC
    FROM sensorData
    WHERE location LIKE 'room%’
    GROUP BY location, TUMBLE(time, INTERVAL '1' DAY)
  """)
```

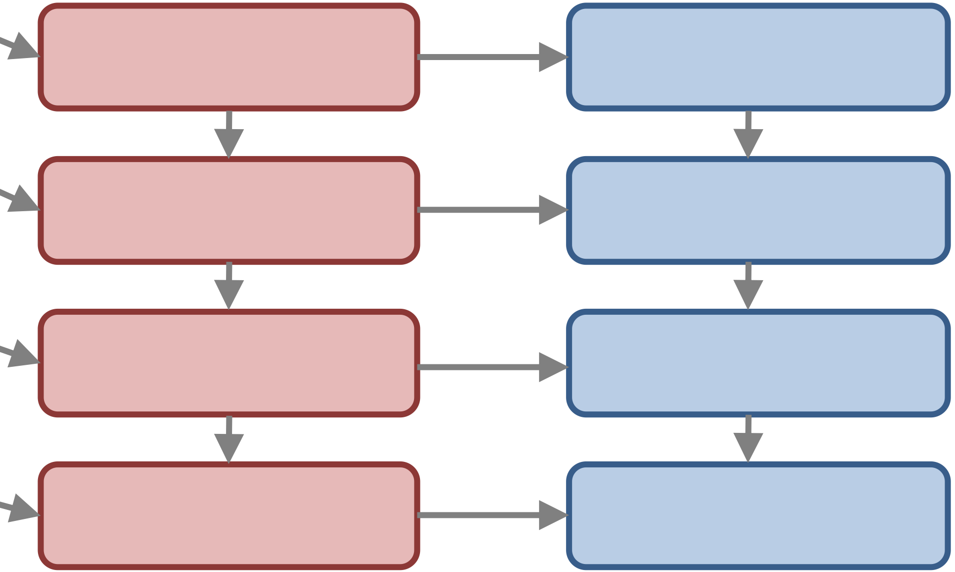


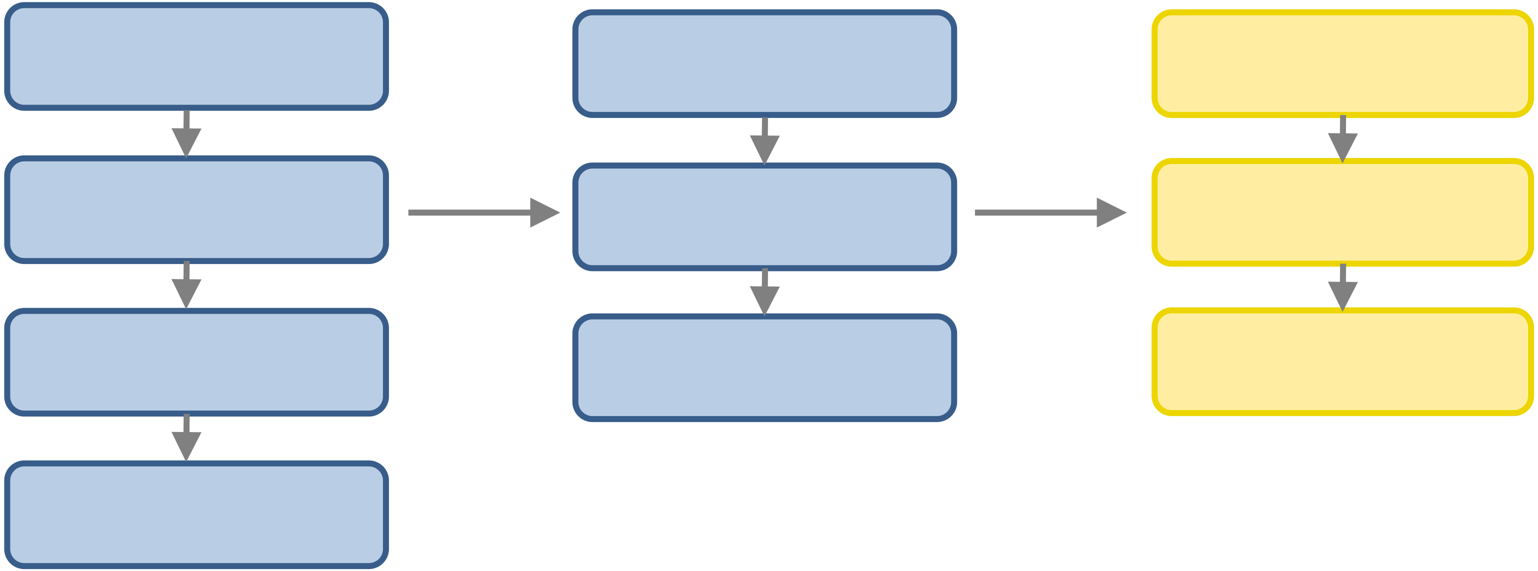


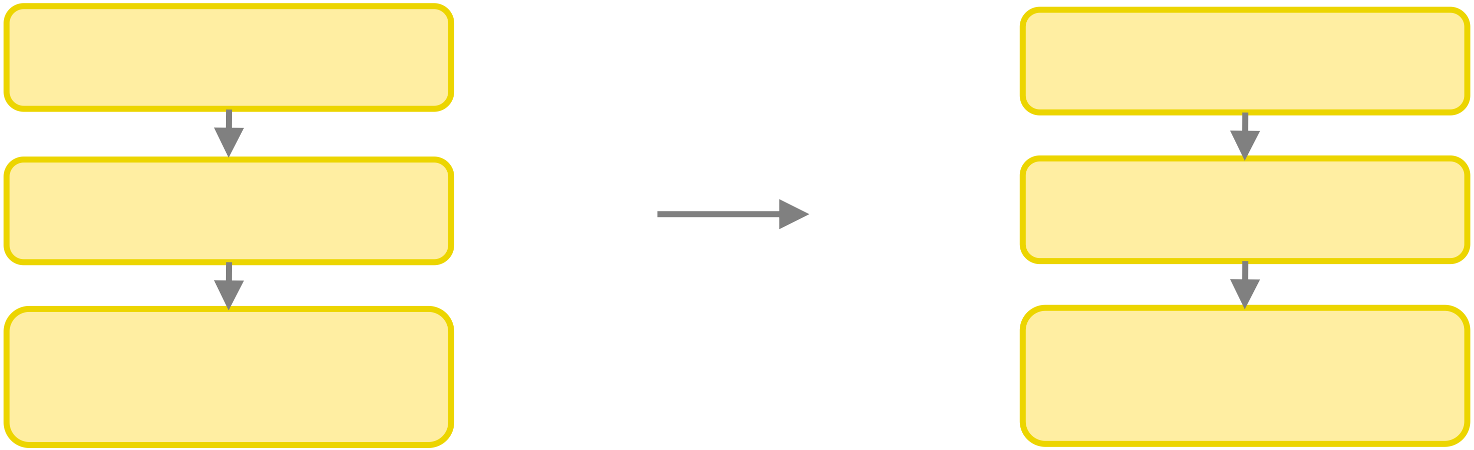


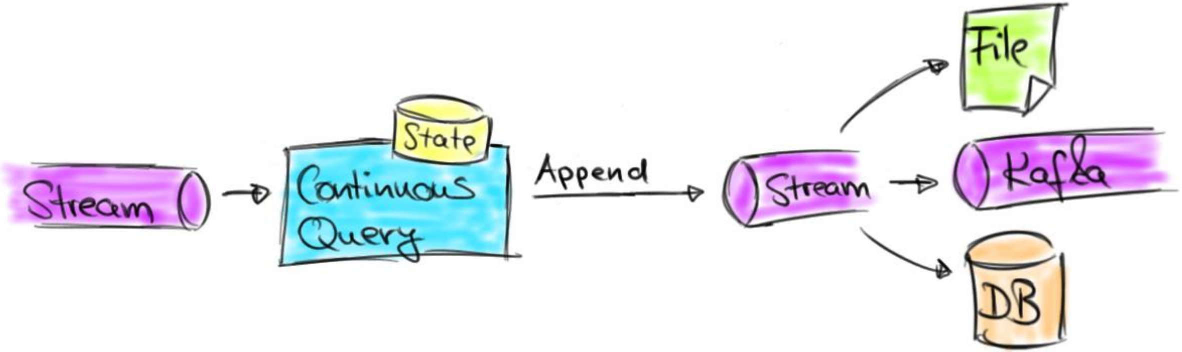


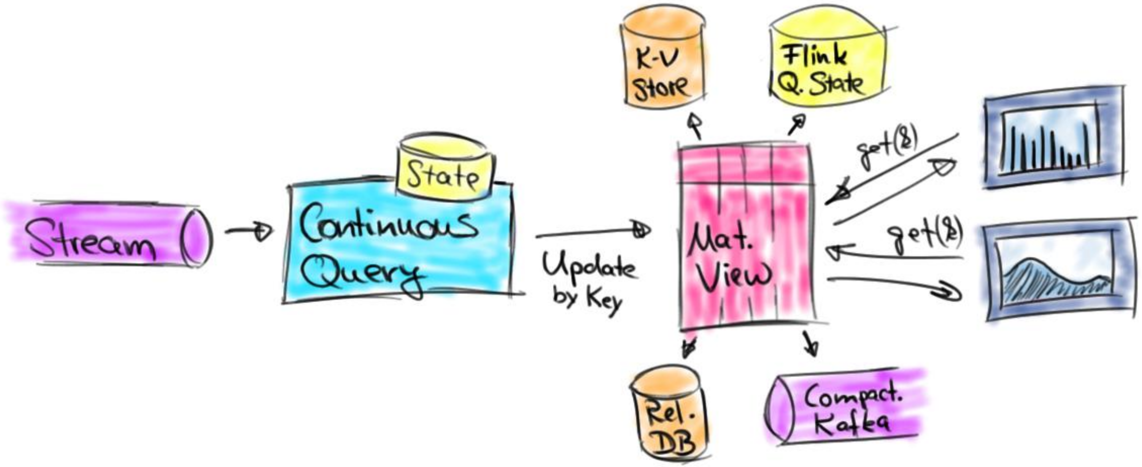
```
sensorTable

  .window(Tumble over 1.day on 'rowtime as 'w)
  .groupBy('location, ’w)

  .select(
    'w.start as 'day,
    'location,
    (('tempF.avg - 32) *
        0.556) as 'avgTempC)

  .where('location like "room%")
```



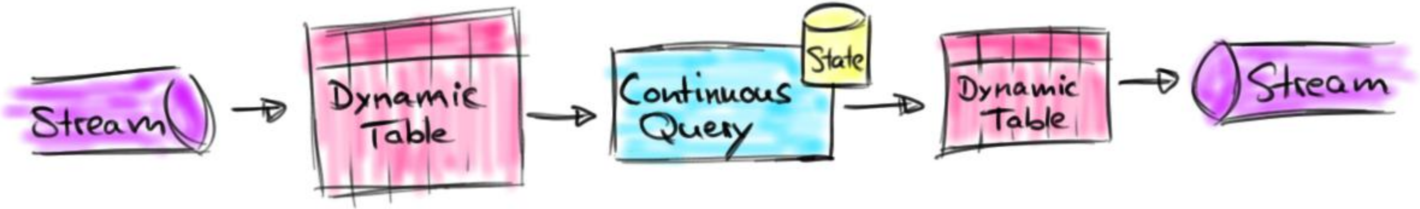


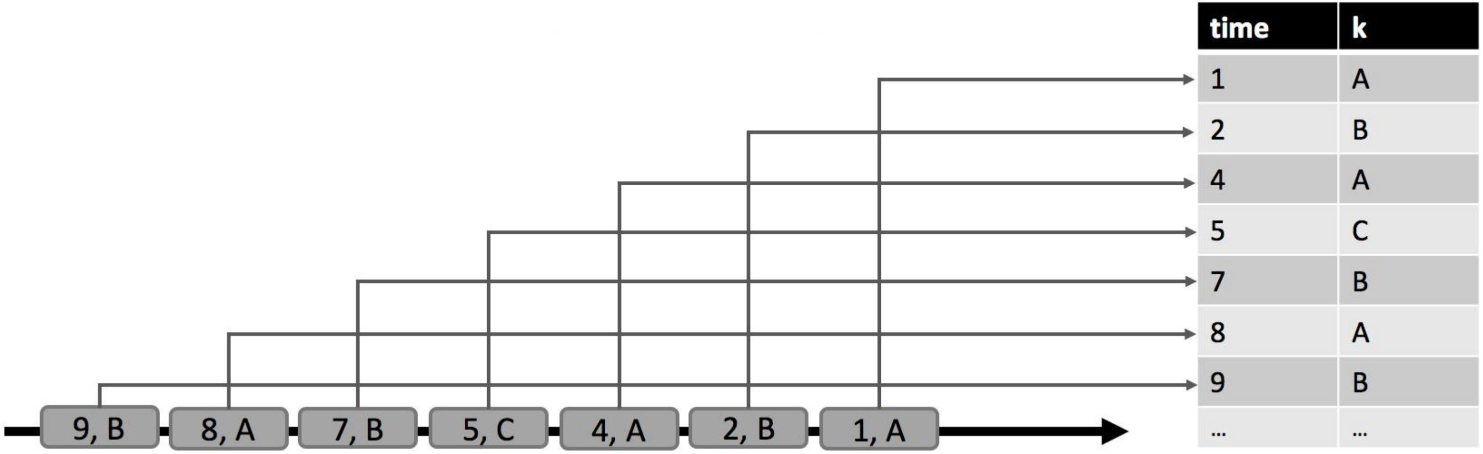



Stream
State
Continuous Query
Append
Stream
File
Kafka
DB

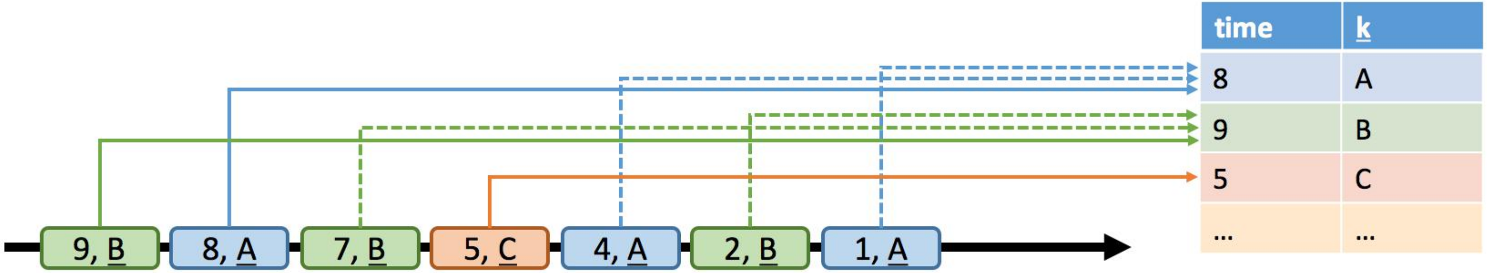

Stream
State
Continuous Query
Update by Key
K-V Store
Flink Q. State
Mat. View
get(k)
get(k)
Rel. DB
Compact. Kafka

← →


Stream
Dynamic Table
Continuous Query
State
Dynamic Table
Stream



time	k
1	A
2	B
4	A
5	C
7	B
8	A
9	B
...	...



time	k
8	A
9	B
5	C
...	...

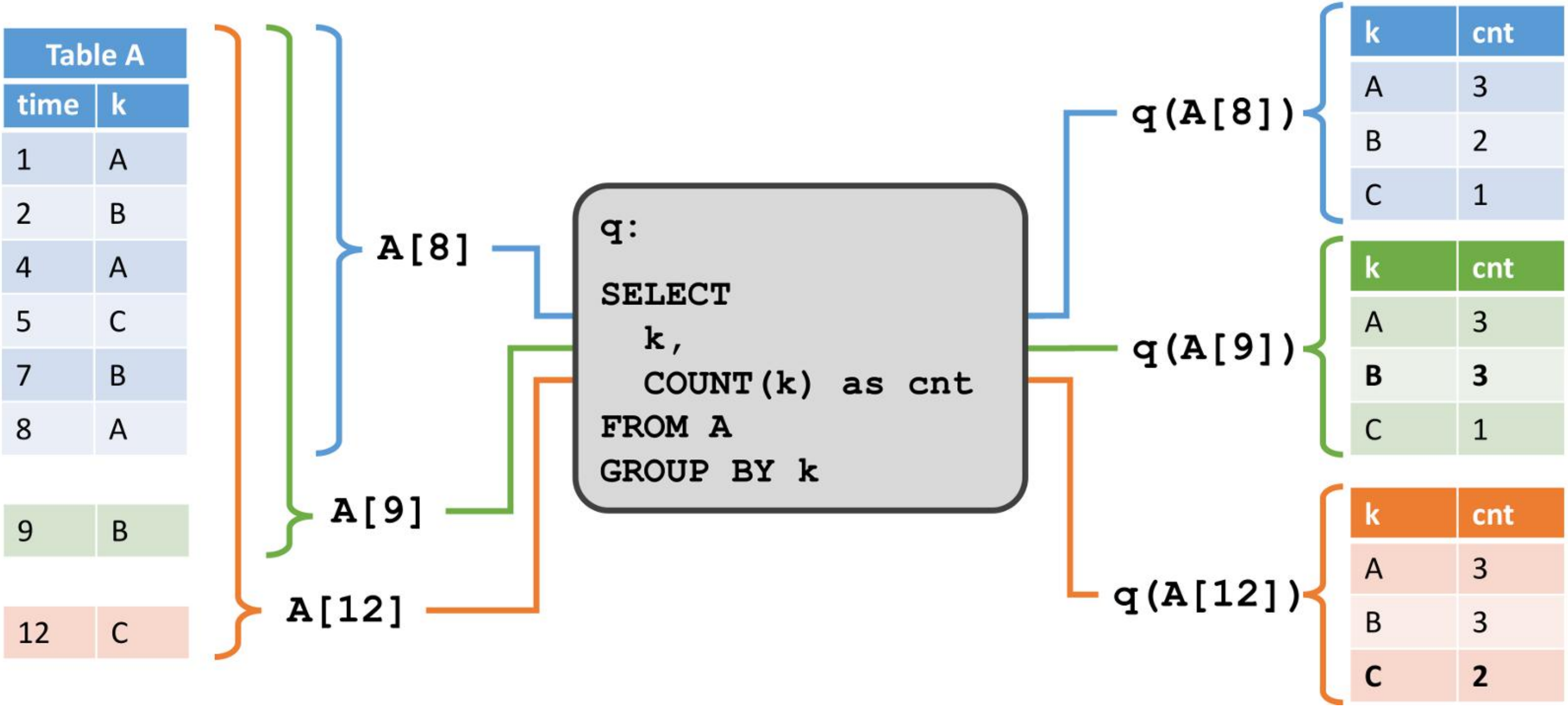


Table A

time	k
1	A
2	B
4	A
5	C
7	B
8	A
9	B
12	C

```
q:

SELECT
  k,
  COUNT(k) as cnt
FROM A
GROUP BY k
```

q(A[8])

k	cnt
A	3
B	2
C	1

q(A[9])

k	cnt
A	3
B	**3**
C	1

q(A[12])

k	cnt
A	3
B	3
C	**2**

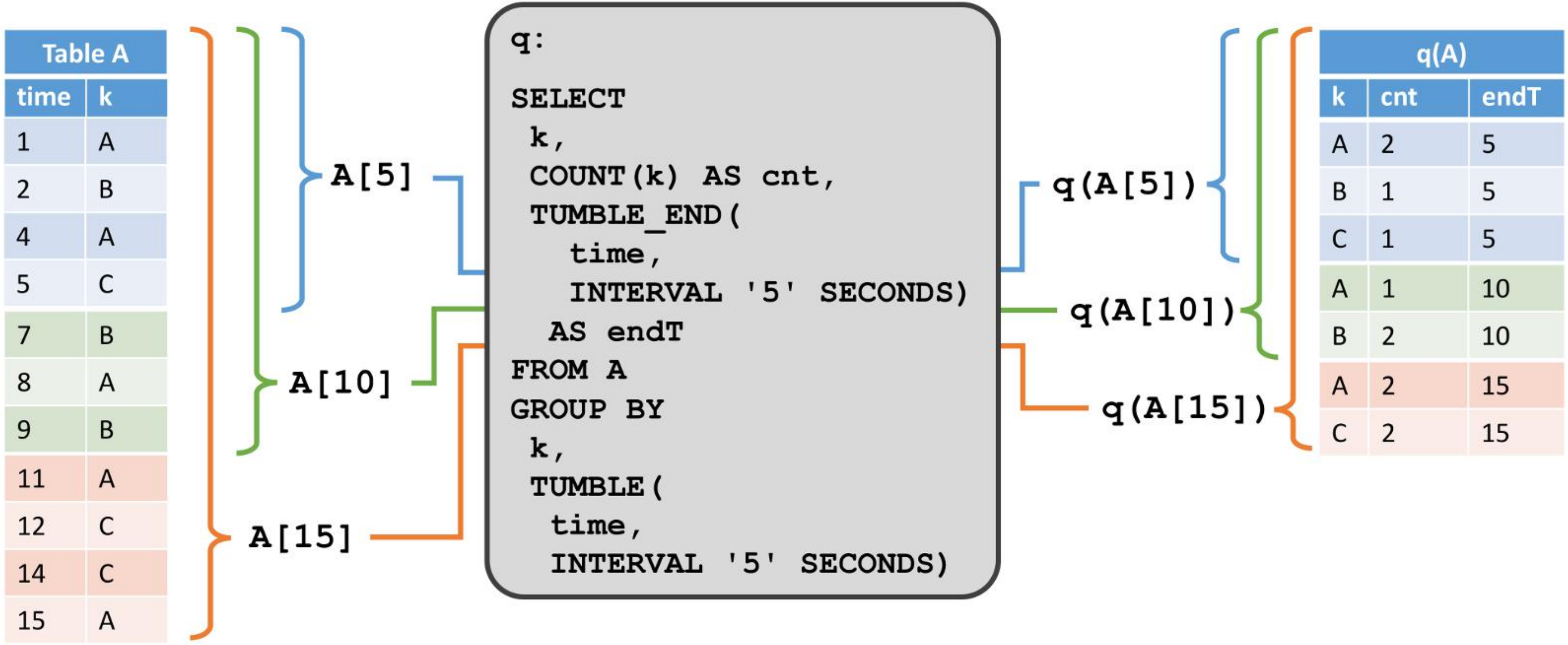


Table A	
time	k
1	A
2	B
4	A
5	C
7	B
8	A
9	B
11	A
12	C
14	C
15	A

```
q:

SELECT
 k,
 COUNT(k) AS cnt,
 TUMBLE_END(
   time,
   INTERVAL '5' SECONDS)
  AS endT
FROM A
GROUP BY
 k,
 TUMBLE(
  time,
  INTERVAL '5' SECONDS)
```

q(A)		
k	cnt	endT
A	2	5
B	1	5
C	1	5
A	1	10
B	2	10
A	2	15
C	2	15

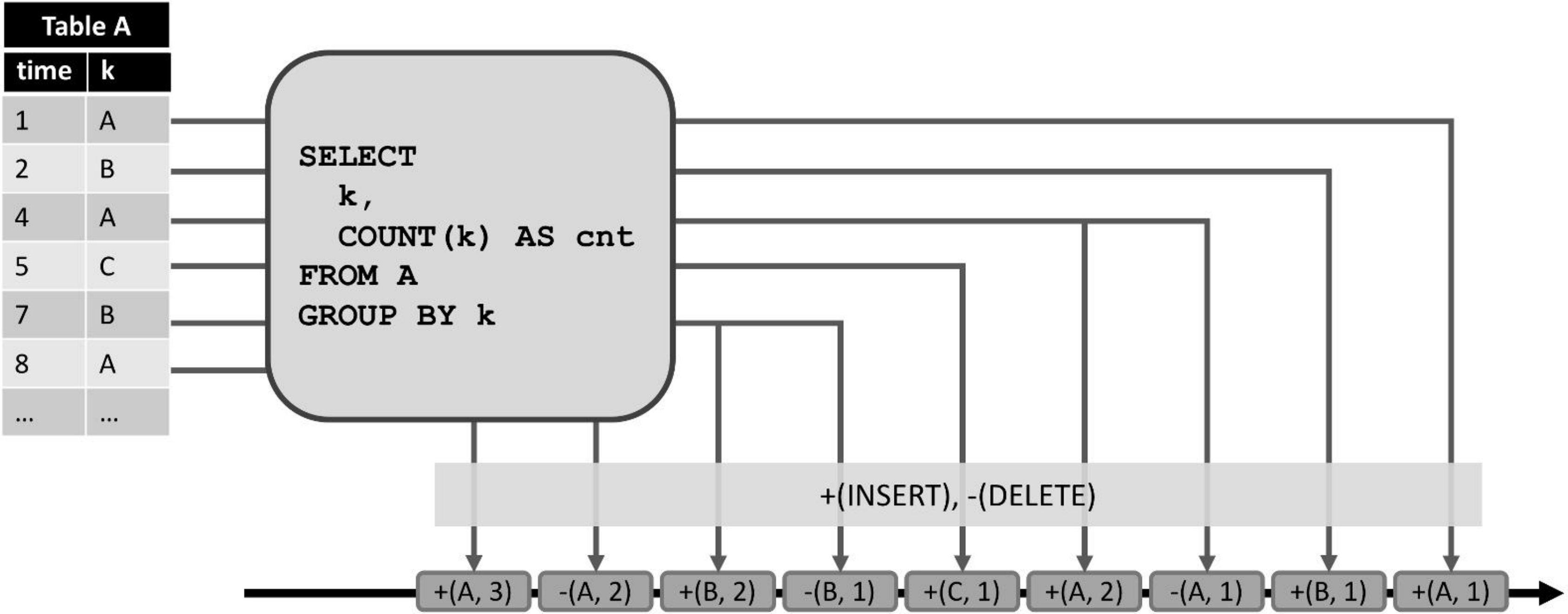


Table A

time	k
1	A
2	B
4	A
5	C
7	B
8	A
...	...

```
SELECT
  k,
  COUNT(k) AS cnt
FROM A
GROUP BY k
```

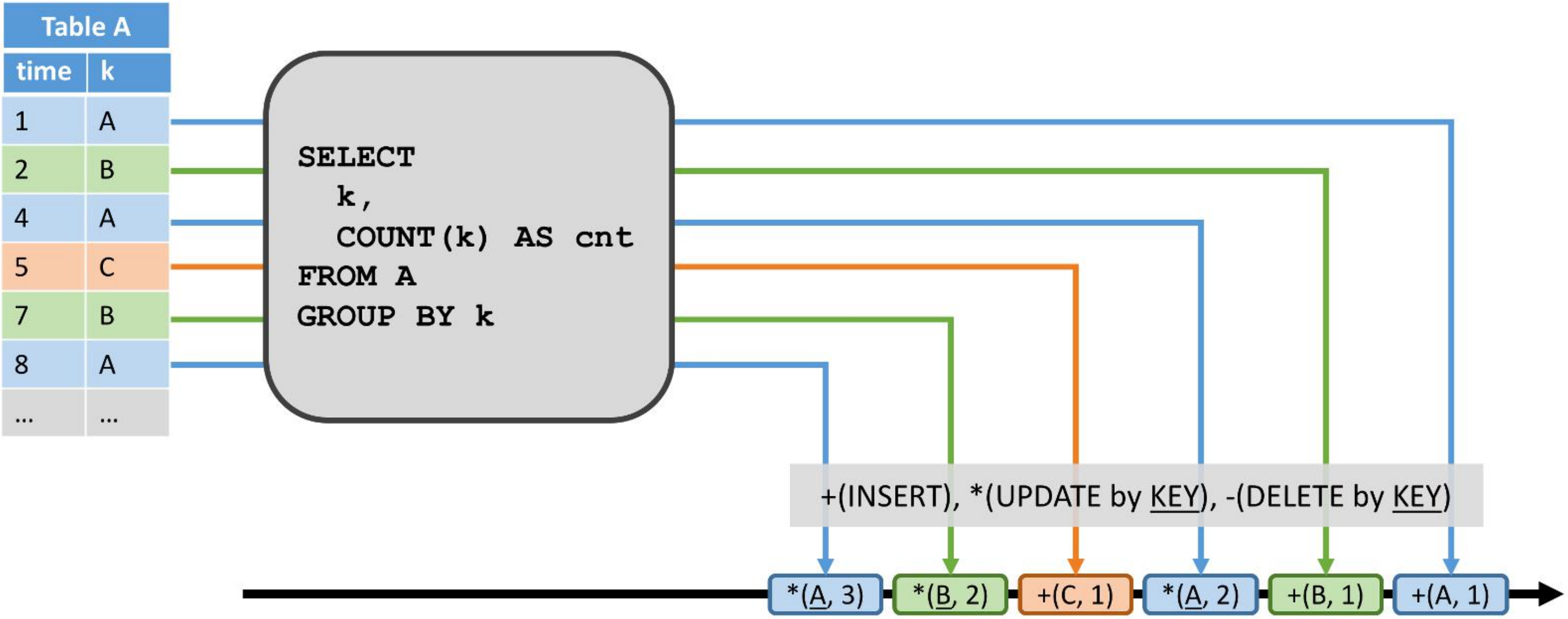
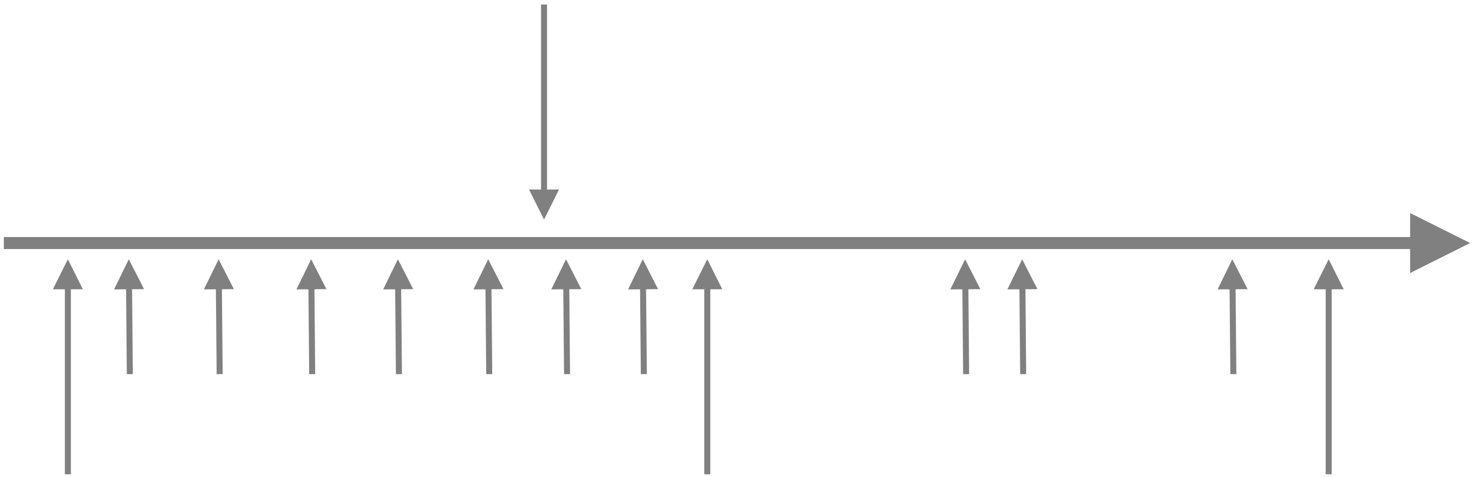


Table A	
time	k
1	A
2	B
4	A
5	C
7	B
8	A
...	...

```
SELECT
  k,
  COUNT(k) AS cnt
FROM A
GROUP BY k
```

+(INSERT), *(UPDATE by KEY), -(DELETE by KEY)



UBER

Alibaba Group

HUAWEI