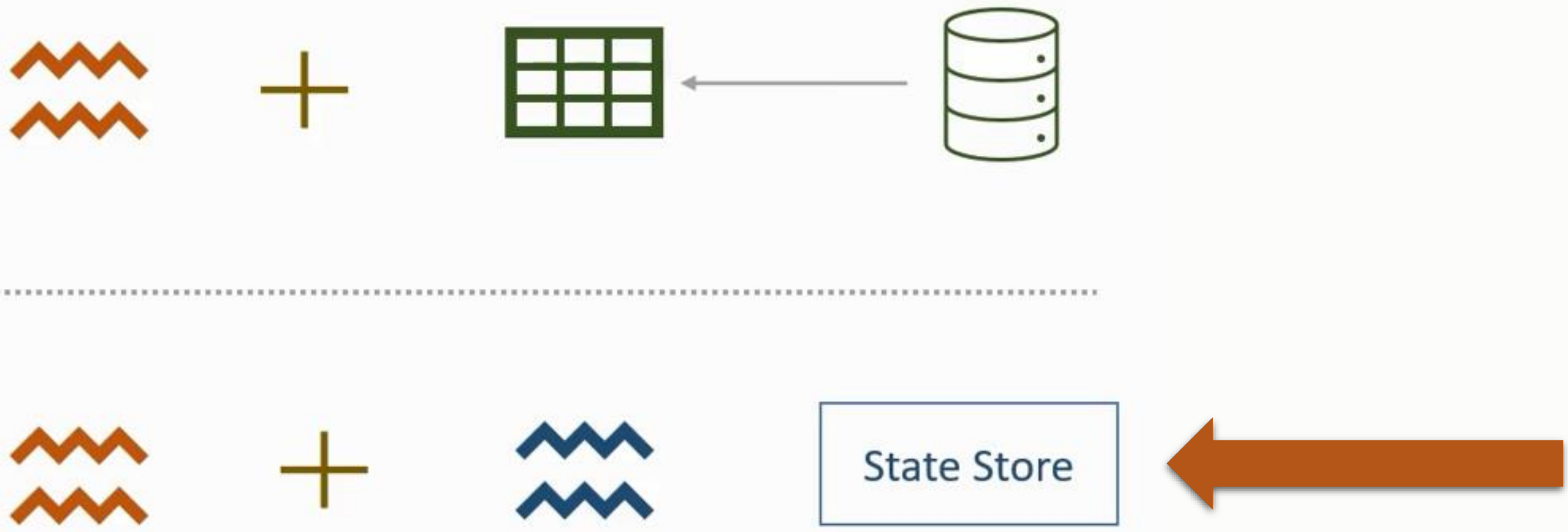
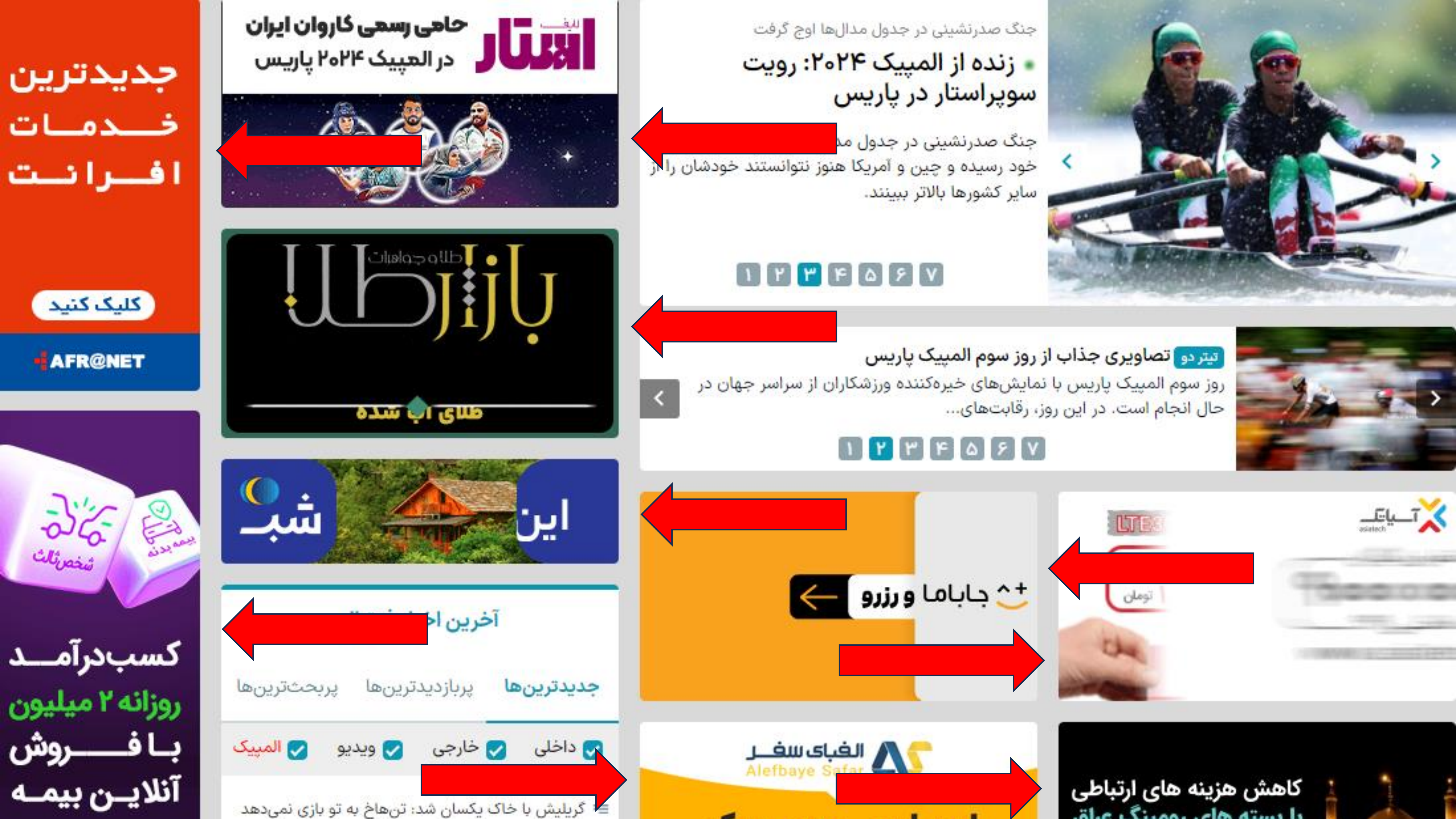
Regarding Stream Joins in Structured Streaming of Spark

THE ULTIMATE KING

Stream to static joins are stateless

Streaming Joins





جدیدترین خدمات افرانت

کلیک کنید

AFR@NET

شخص ثالث

بیمه بدنه

کسب درآمد روزانه ۲ میلیون با فروش آنلاین بیمه

لایف استار

حامی رسمی کاروان ایران در المپیک ۲۰۲۴ پاریس

بازار طلا

طلا و جواهرات

طلای آب شده

این شب

آخرین ا

جدیدترین‌ها　پربازدیدترین‌ها　پربحث‌ترین‌ها

☑ داخلی　☑ خارجی　☑ ویدیو　☑ المپیک

گریلیش با خاک یکسان شد: تن‌هاخ به تو بازی نمی‌دهد

جنگ صدرنشینی در جدول مدال‌ها اوج گرفت

زنده از المپیک ۲۰۲۴: رویت سوپراستار در پاریس

جنگ صدرنشینی در جدول مد
خود رسیده و چین و آمریکا هنوز نتوانستند خودشان را از
سایر کشورها بالاتر ببینند.

۱ ۲ ۳ ۴ ۵ ۶ ۷

تیتر دو تصاویری جذاب از روز سوم المپیک پاریس

روز سوم المپیک پاریس با نمایش‌های خیره‌کننده ورزشکاران از سراسر جهان در حال انجام است. در این روز، رقابت‌های...

۱ ۲ ۳ ۴ ۵ ۶ ۷

جاباما و رزرو

آسیاتک
asiatech

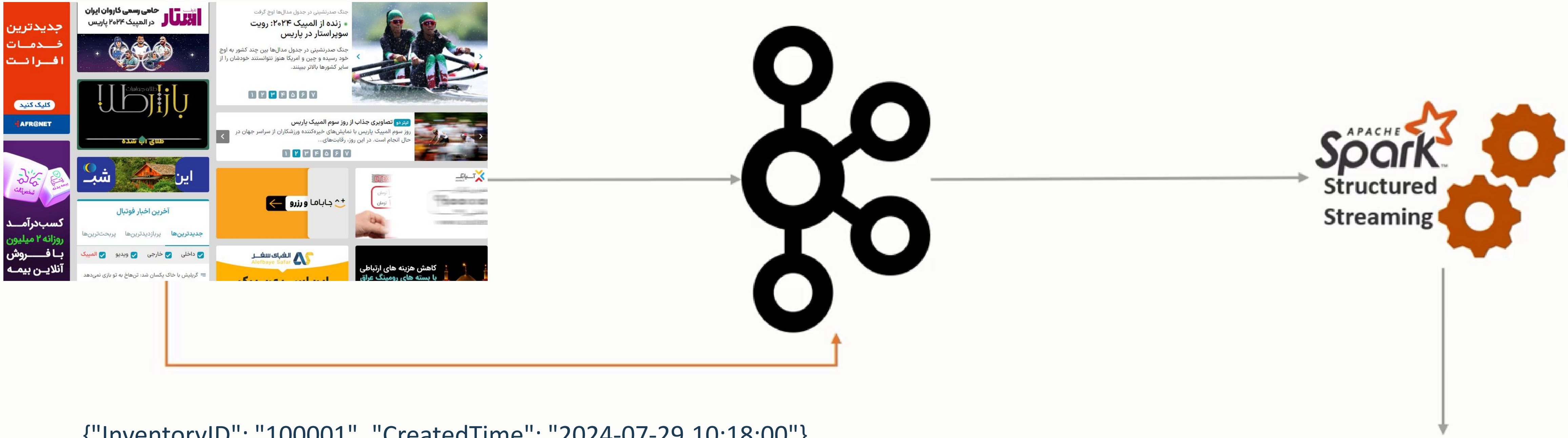
LTE

تومان

الفبای سفر
Alefbaye Safar

کاهش هزینه های ارتباطی

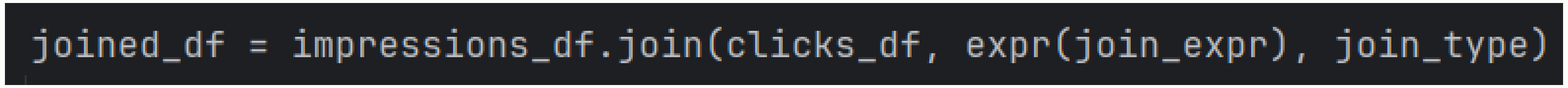
{"InventoryID": "100001", "CreatedTime": "2024-07-29 10:00:00", "Campaigner": "ABC Ltd"}



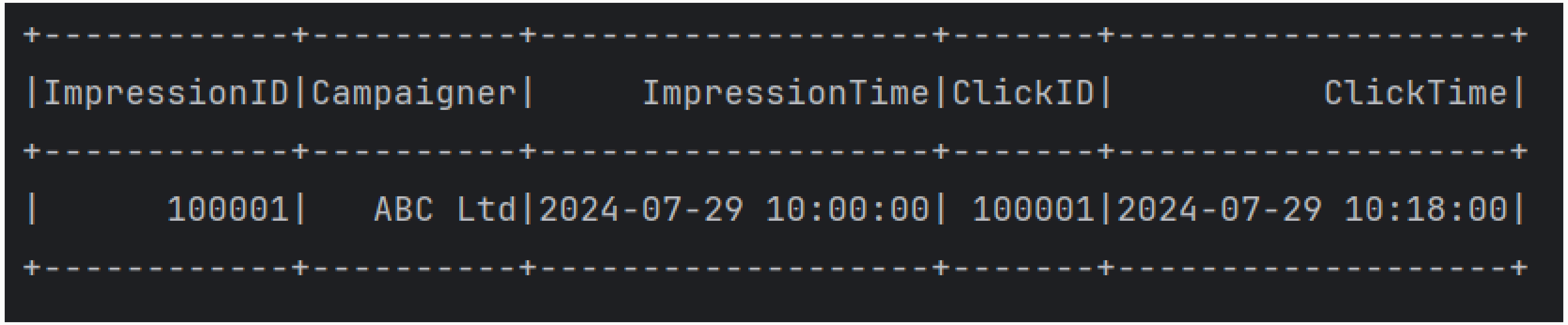
{"InventoryID": "100001", "CreatedTime": "2024-07-29 10:18:00"}

Clicked Impressions

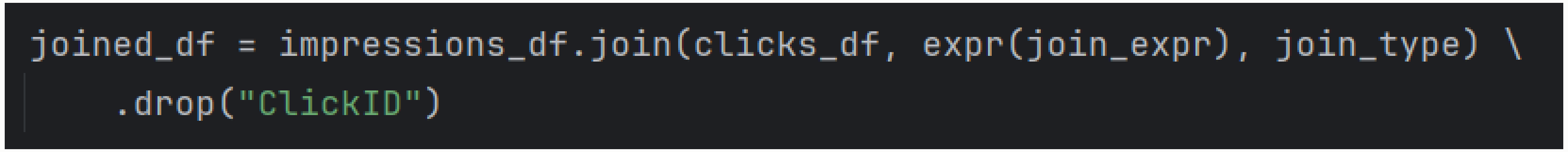
If an user clicked on an advertisement, we can join these two dataframes for further analysis.



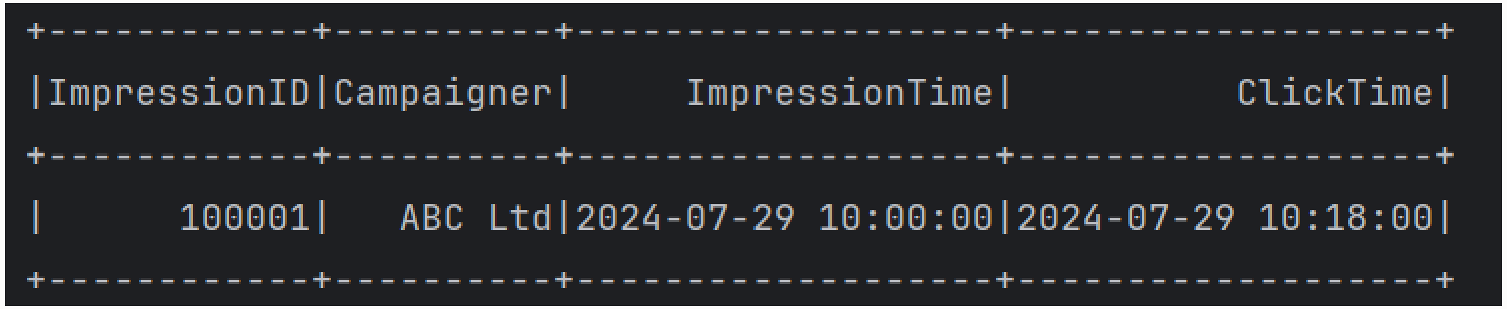
```
joined_df = impressions_df.join(clicks_df, expr(join_expr), join_type)
```



```
+------------+----------+-------------------+-------+-------------------+
|ImpressionID|Campaigner|     ImpressionTime|ClickID|          ClickTime|
+------------+----------+-------------------+-------+-------------------+
|      100001|   ABC Ltd|2024-07-29 10:00:00| 100001|2024-07-29 10:18:00|
+------------+----------+-------------------+-------+-------------------+
```

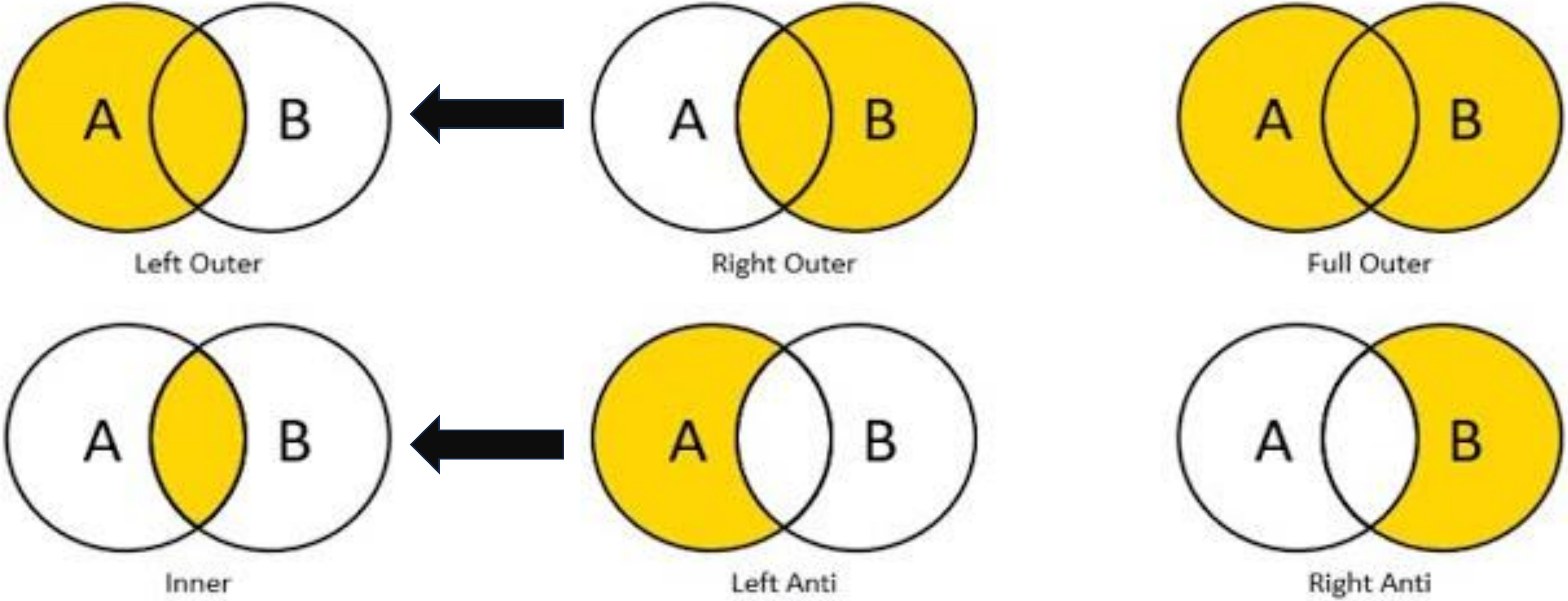


```
joined_df = impressions_df.join(clicks_df, expr(join_expr), join_type) \
    .drop("ClickID")
```



```
+------------+----------+-------------------+-------------------+
|ImpressionID|Campaigner|     ImpressionTime|          ClickTime|
+------------+----------+-------------------+-------------------+
|      100001|   ABC Ltd|2024-07-29 10:00:00|2024-07-29 10:18:00|
+------------+----------+-------------------+-------------------+
```

JOIN Types



Left Outer

Right Outer

Full Outer

Inner

Left Anti

Right Anti

Left Input	Right Input	Join Type	description
Static	Static	All types	Supported, since its not on streaming data even though it can be present in a streaming query
Stream	Static	Inner	Supported, not stateful
		Left Outer	Supported, not stateful
		Right Outer	Not supported
		Full Outer	Not supported
		Left Semi	Supported, not stateful

Left Input	Right Input	Join Type	description
Static	Stream	Inner	Supported, not stateful
		Left Outer	Not supported
		Right Outer	Supported, not stateful
		Full Outer	Not supported
		Left Semi	Not supported

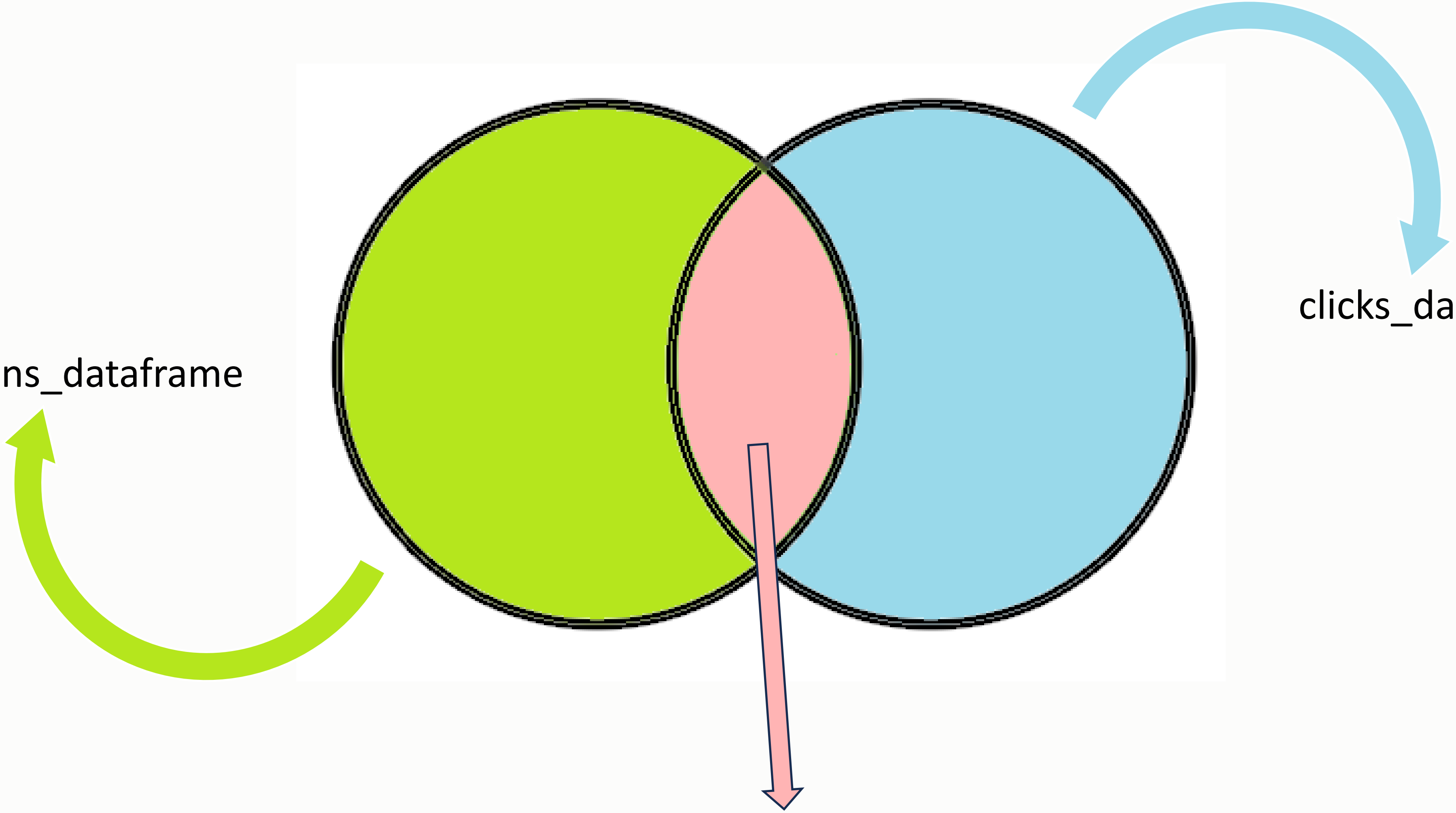
Left Input	Right Input	Join Type	description
Stream	Stream	Inner	Supported, optionally specify watermark on both sides + time constraints for state cleanup
		Left Outer	Conditionally supported, must specify watermark on right + time constraints for correct results, optionally specify watermark on left for all state cleanup
		Right Outer	Conditionally supported, must specify watermark on left + time constraints for correct results, optionally specify watermark on right for all state cleanup
		Full Outer	Conditionally supported, must specify watermark on one side + time constraints for correct results, optionally specify watermark on the other side for all state cleanup
		Left Semi	Conditionally supported, must specify watermark on right + time constraints for correct results, optionally specify watermark on left for all state cleanup

APACHE Spark™
Structured
Streaming

A watermark delay of “2 hours” guarantees that the engine will never drop any data that is less than 2 hours delayed.

BUT data delayed by more than 2 hours **may or may not get processed** 🤔.

It’s ok. **In distributed frameworks everything is strange** 😎.

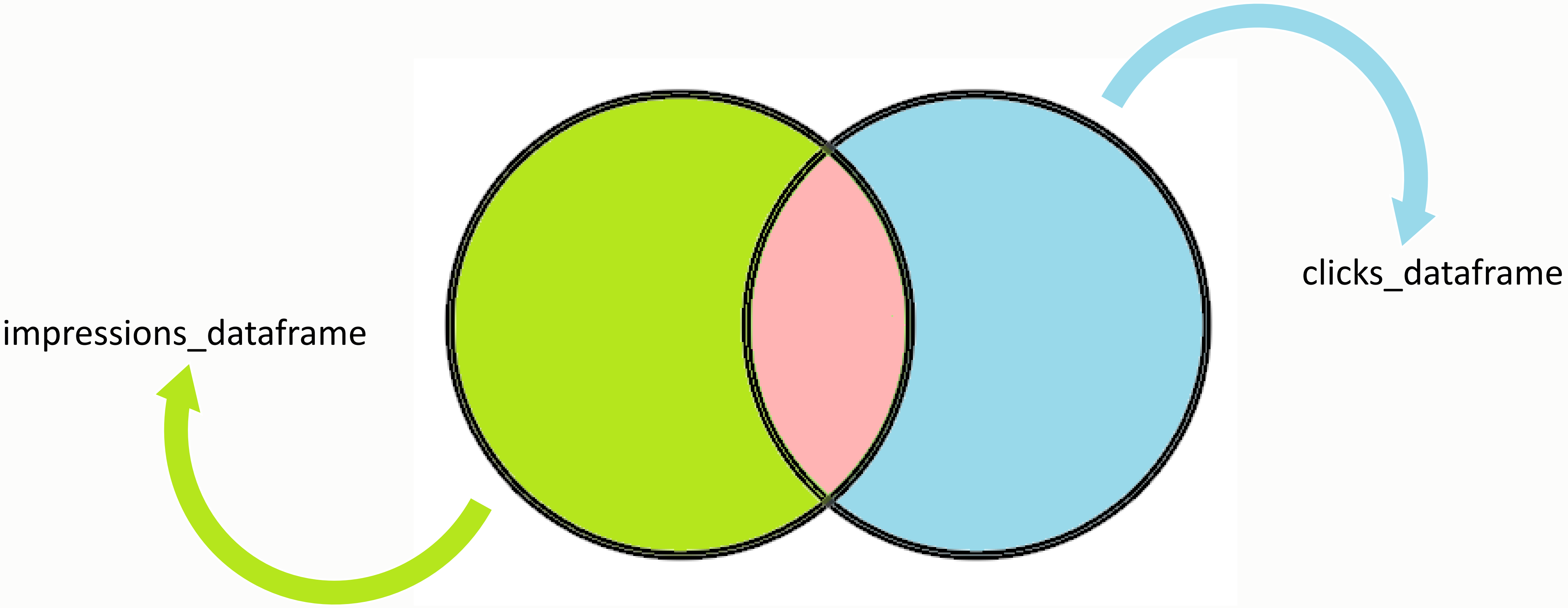


impressions_dataframe

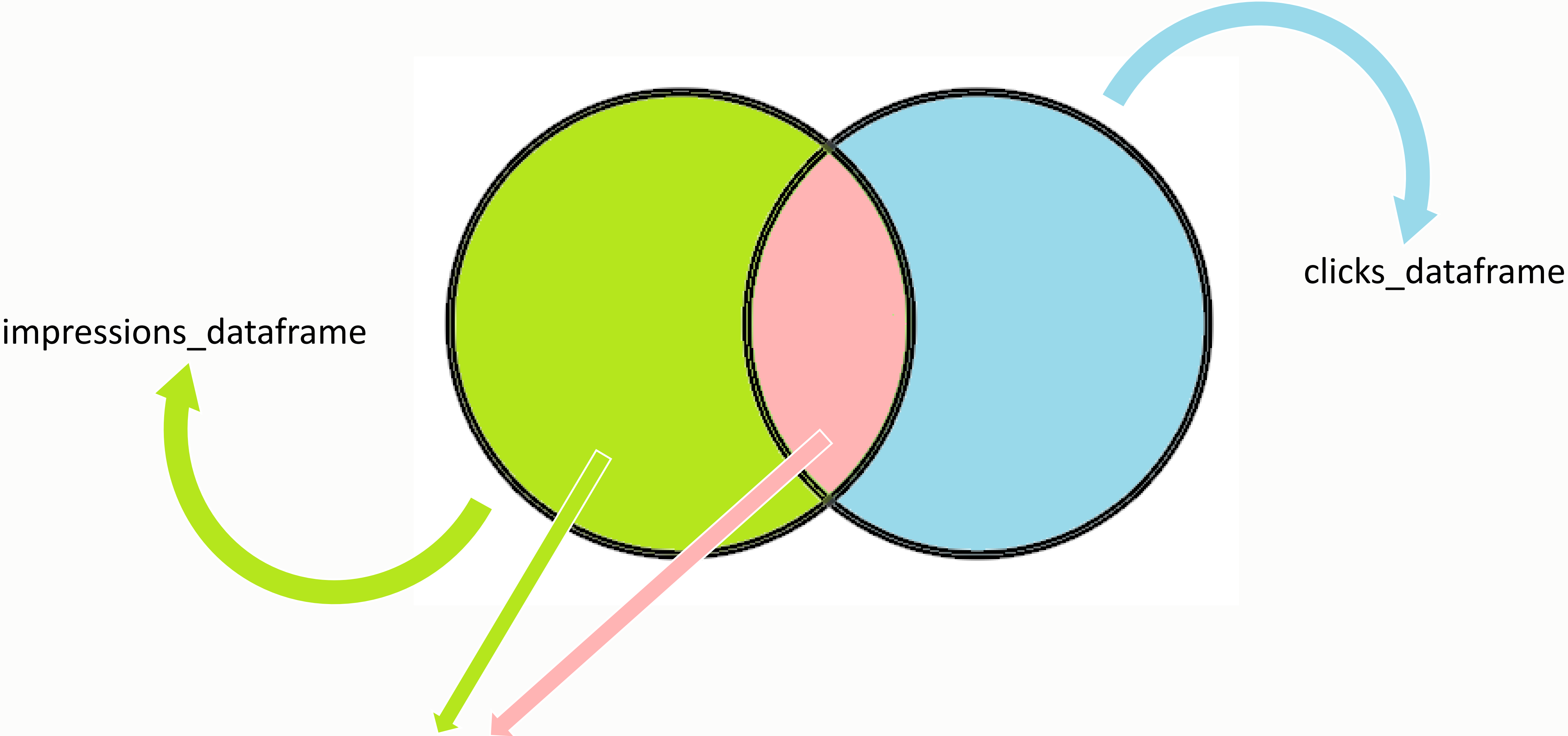
clicks_dataframe

In the first example, we are interested in the **INNER join** between impressions_df and clicks_df.

We will see **duplicate values** in this case 🤔.



In the second example, we will add **watermarking** to manage the state store and avoid duplicate values.



impressions_dataframe

clicks_dataframe

Finally we will do **OUTER join** between these two dataframes. Outer join with watermarking is very complex. It takes a lot of our attention!