Problem Statement → Early Search Signal(implicit behavioral signals of relevance and ranking quality): **Time to first click**

TTFC is a **leading metric** (micro-behavior, within session).

QRR is a **session-level aggregate metric** (macro outcome).

Solution:

Reg. Time-to-first-click/session

```
Select
Session_id,
MIN(c.click_time - s.search_time) AS time_to_first_click
From search_events s
JOIN search_events c
On s.session_id = c.session_id
AND s.event_type = 'search'
And c.event_type = 'click'
AND c.timestamp > s.timestamp
GROUP BY session_id
```

Reg. Time-to-first-click/session-user-query

```
With searches AS(
Select
Session_id,
User_id,
Query,
Timestamp AS search_time
From search_events
Where event_type = 'search' ),
Clicks AS(
Select
Session_id,
Timestamp AS click_time
From search_events
Where event_type = 'click' )
Select
s.session_id,
s.user_id,
s.query,
MIN(c.click_time - s.search_time) AS time_to_first_click
From searches s
JOIN clicks c
```

On s.session_id = c.session_id → Same Session
And c.click_time > s.search_time
Group by s.session_id, s.user_id, s.query;

Explanation:

1. We join searches and clicks within the same session
2. Only keep clicks after the search timestamp (c.click_time > s.search_time)
3. MIN() ensures we only take the first click after the query

---

No Reformulation Rate (%) = Sessions with only 1 unique or semantically the same query/Total sessions *100

```
Select
Session_id,
User_id,
Query as current_query,
lag(query) over(partition by timestamp) as previous_query,
CASE
WHEN LAG(query) OVER(PARTITION BY session_id ORDER BY timestamp) is null then ‘first_query’
WHEN query = LAG(query) OVER(PARTITION BY session_id ORDER BY timestamp) then ‘same_query’
ELSE ‘potential_reformulation’
END AS query_status
FROM ordered_queries
```