Analyzing Clickstreams Using Subsessions

Jesper Andersen, Anders Giversen, Janne Skyt from cs.aau.dk
Torben Bach Pedersen from Analyze.dk
Allan H. Jensen, Rune S. Larsen from Nykredit Aalborg

ACM International Workshop on Data Warehousing and OLAP (DOLAP 2000)

Presented by Nicholas Tinggaard

Outline

- Motivation / The Case
- What is Clickstreams
- Existing models
- New combined solution
- Problems / solutions
- Evaluation
- Conclusion
- Relation to our work

Motivation / The Case

- Corporate web
  - Personalisation
  - Optimisation
  - Adaptive sites
- Analysing user behaviour
  - Sequences of clicks
  - SpeedTracer from IBM
    - No Data Warehouse
      - □ Inflexible
      - □ No OLAP
- Nykredit case (Danish mortage provider)
  - Banner effiency
  - Session kills (killer subsessions)
    - Parts of the website that could be badly written or structured

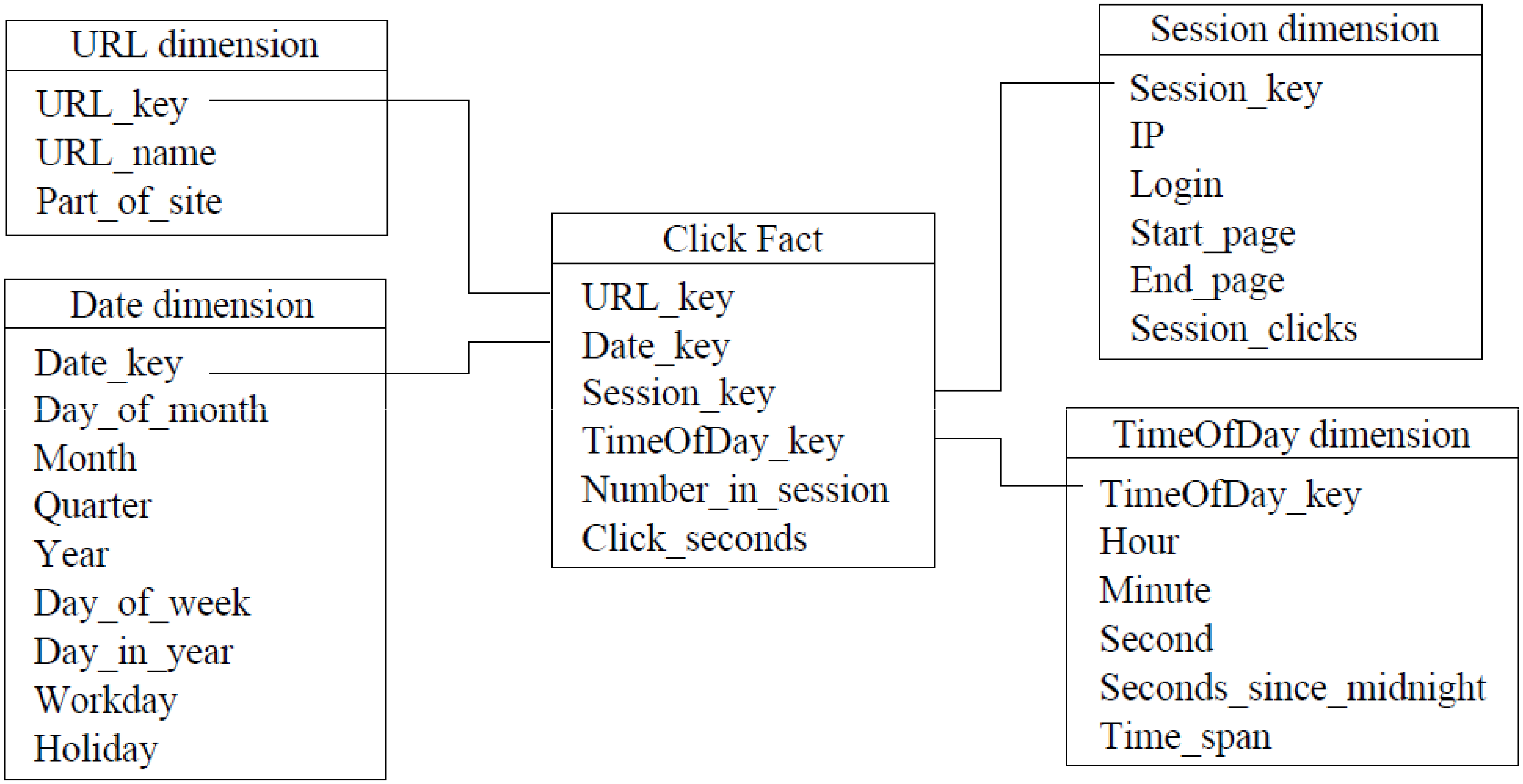
What is Clickstreams

- Sequences of clicks on a website
- User session
- Web server log file
  - IP Address, URL, Timestamp
    - Time on each page, Group users on different terms like geography
  - Cookie
- Existing modelling methods
  - Click Fact Table
  - Session Fact Table
  - Existing methods has limitations
- New Subsession model

Click Fact Table

- Click fact introduced by R. Kimball
- Star-Schema
  - Single clicks as facts
  - Good detail level
  - Hard to query sequences of clicks
    - Multiple self-joins on the fact table
  - Designed for queries on single clicks

Click Fact Star Schema



URL dimension
- URL_key
- URL_name
- Part_of_site

Date dimension
- Date_key
- Day_of_month
- Month
- Quarter
- Year
- Day_of_week
- Day_in_year
- Workday
- Holiday

Click Fact
- URL_key
- Date_key
- Session_key
- TimeOfDay_key
- Number_in_session
- Click_seconds

Session dimension
- Session_key
- IP
- Login
- Start_page
- End_page
- Session_clicks

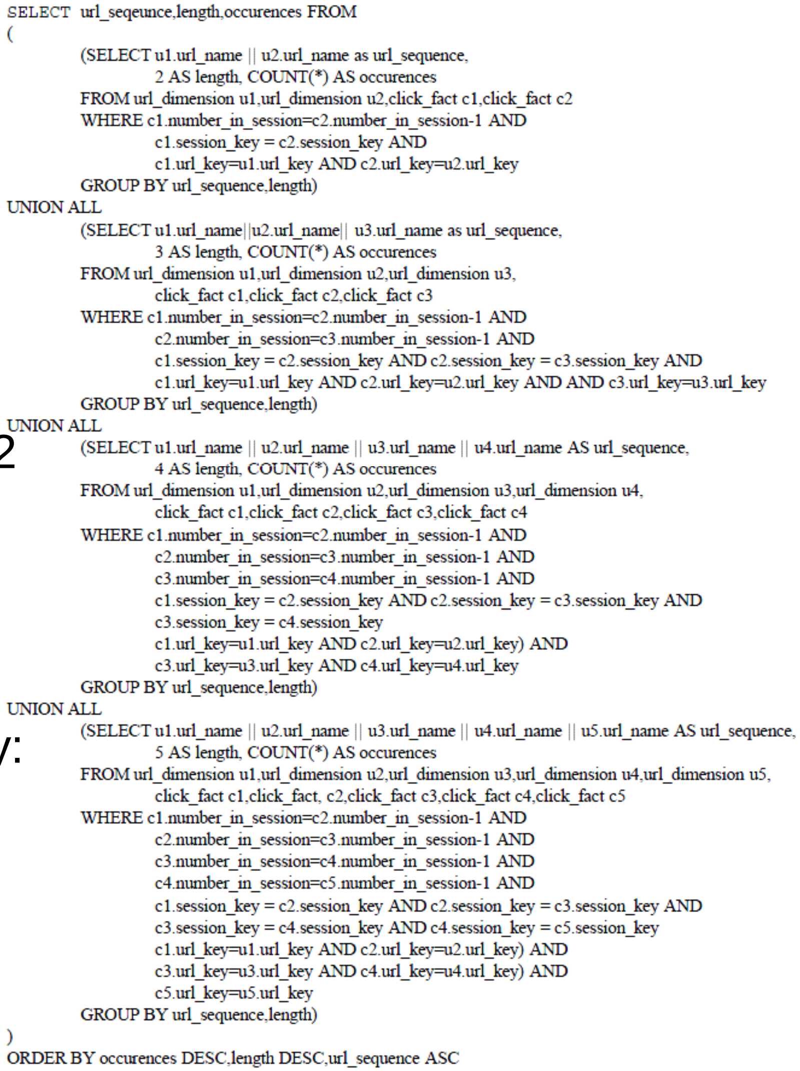
TimeOfDay dimension
- TimeOfDay_key
- Hour
- Minute
- Second
- Seconds_since_midnight
- Time_span

Figure 8: Click Fact Star Schema

Simple sequence query on a Click fact model

- Simple query that selects sequences of clicks up to 5 in length
- Arrange the results after longest and most frequent occurring sequences

- The query in short words:
  - Selects all sequences of length 2
  - Union all sequences of length 3
  - Union all sequences of length 4
  - Union all sequences of length 5
  - On the constructed set the query:
  - Group by occurrences, length and url_sequence

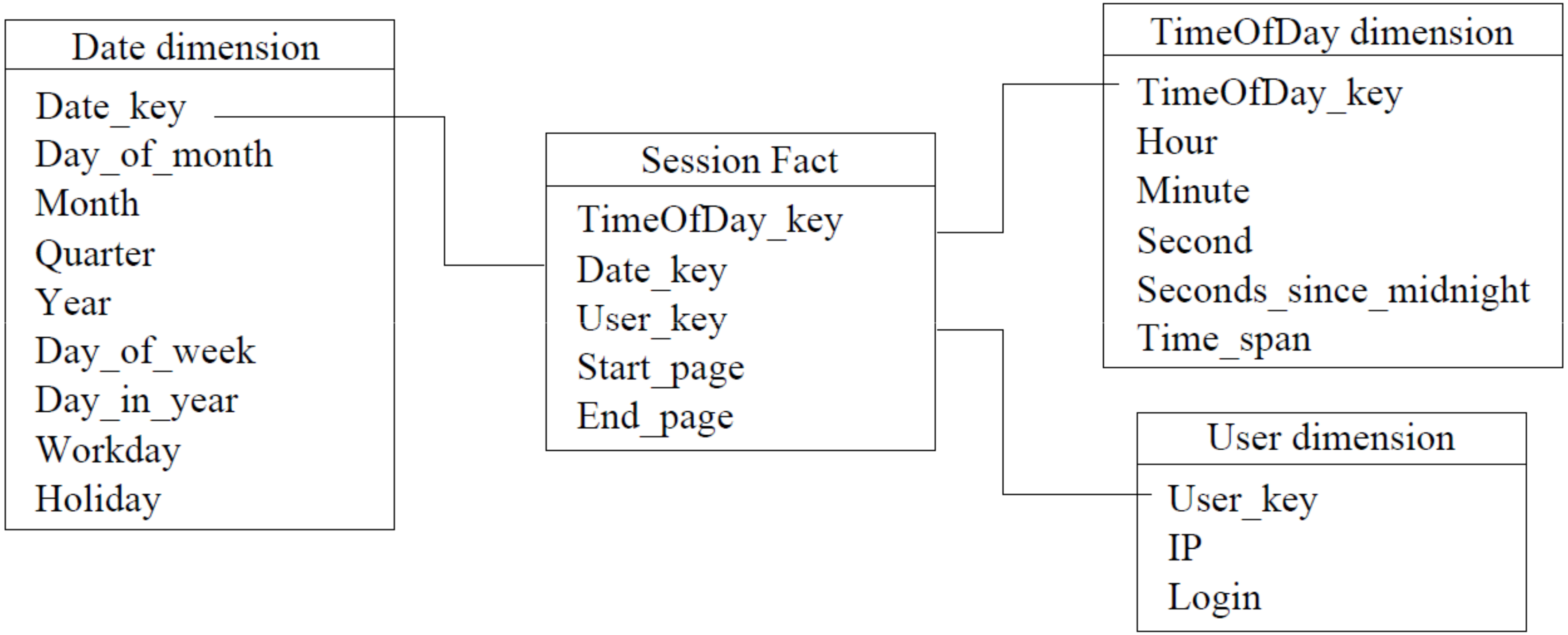


```
SELECT  url_seqeunce,length,occurences FROM
(
        (SELECT u1.url_name || u2.url_name as url_sequence,
                2 AS length, COUNT(*) AS occurences
        FROM url_dimension u1,url_dimension u2,click_fact c1,click_fact c2
        WHERE c1.number_in_session=c2.number_in_session-1 AND
                c1.session_key = c2.session_key AND
                c1.url_key=u1.url_key AND c2.url_key=u2.url_key
        GROUP BY url_sequence,length)
UNION ALL
        (SELECT u1.url_name||u2.url_name|| u3.url_name as url_sequence,
                3 AS length, COUNT(*) AS occurences
        FROM url_dimension u1,url_dimension u2,url_dimension u3,
                click_fact c1,click_fact c2,click_fact c3
        WHERE c1.number_in_session=c2.number_in_session-1 AND
                c2.number_in_session=c3.number_in_session-1 AND
                c1.session_key = c2.session_key AND c2.session_key = c3.session_key AND
                c1.url_key=u1.url_key AND c2.url_key=u2.url_key AND AND c3.url_key=u3.url_key
        GROUP BY url_sequence,length)
UNION ALL
        (SELECT u1.url_name || u2.url_name || u3.url_name || u4.url_name AS url_sequence,
                4 AS length, COUNT(*) AS occurences
        FROM url_dimension u1,url_dimension u2,url_dimension u3,url_dimension u4,
                click_fact c1,click_fact c2,click_fact c3,click_fact c4
        WHERE c1.number_in_session=c2.number_in_session-1 AND
                c2.number_in_session=c3.number_in_session-1 AND
                c3.number_in_session=c4.number_in_session-1 AND
                c1.session_key = c2.session_key AND c2.session_key = c3.session_key AND
                c3.session_key = c4.session_key
                c1.url_key=u1.url_key AND c2.url_key=u2.url_key) AND
                c3.url_key=u3.url_key AND c4.url_key=u4.url_key
        GROUP BY url_sequence,length)
UNION ALL
        (SELECT u1.url_name || u2.url_name || u3.url_name || u4.url_name || u5.url_name AS url_sequence,
                5 AS length, COUNT(*) AS occurences
        FROM url_dimension u1,url_dimension u2,url_dimension u3,url_dimension u4,url_dimension u5,
                click_fact c1,click_fact, c2,click_fact c3,click_fact c4,click_fact c5
        WHERE c1.number_in_session=c2.number_in_session-1 AND
                c2.number_in_session=c3.number_in_session-1 AND
                c3.number_in_session=c4.number_in_session-1 AND
                c4.number_in_session=c5.number_in_session-1 AND
                c1.session_key = c2.session_key AND c2.session_key = c3.session_key AND
                c3.session_key = c4.session_key AND c4.session_key = c5.session_key
                c1.url_key=u1.url_key AND c2.url_key=u2.url_key) AND
                c3.url_key=u3.url_key AND c4.url_key=u4.url_key) AND
                c5.url_key=u5.url_key
        GROUP BY url_sequence,length)
)
ORDER BY occurences DESC,length DESC,url_sequence ASC
```

Session Fact Table

- Star-Schema
  - Sessions as facts
  - Session questions is easy to query
    - Queries that is about start and end page
  - Internal clicks gets lost
    - Not useful for behaviour analysis

Session Fact Schema



Date dimension

Date_key
Day_of_month
Month
Quarter
Year
Day_of_week
Day_in_year
Workday
Holiday

Session Fact

TimeOfDay_key
Date_key
User_key
Start_page
End_page

TimeOfDay dimension

TimeOfDay_key
Hour
Minute
Second
Seconds_since_midnight
Time_span

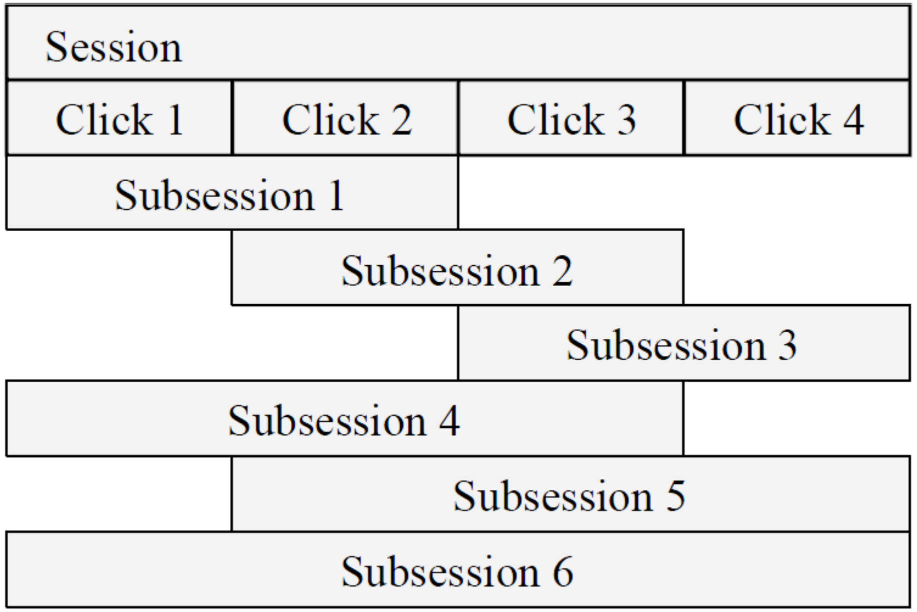
User dimension

User_key
IP
Login

Figure 9: Session fact star scheme

Subsession fact model

- Store all possible sequences of clicks from each session
  - Sessions span many subsession
  - Subsessions overlap
- URL sequence dimension
  - Stores summarised fact ir



Session

Click 1

Click 2

Click 3

Click 4

Subsession 1

Subsession 2

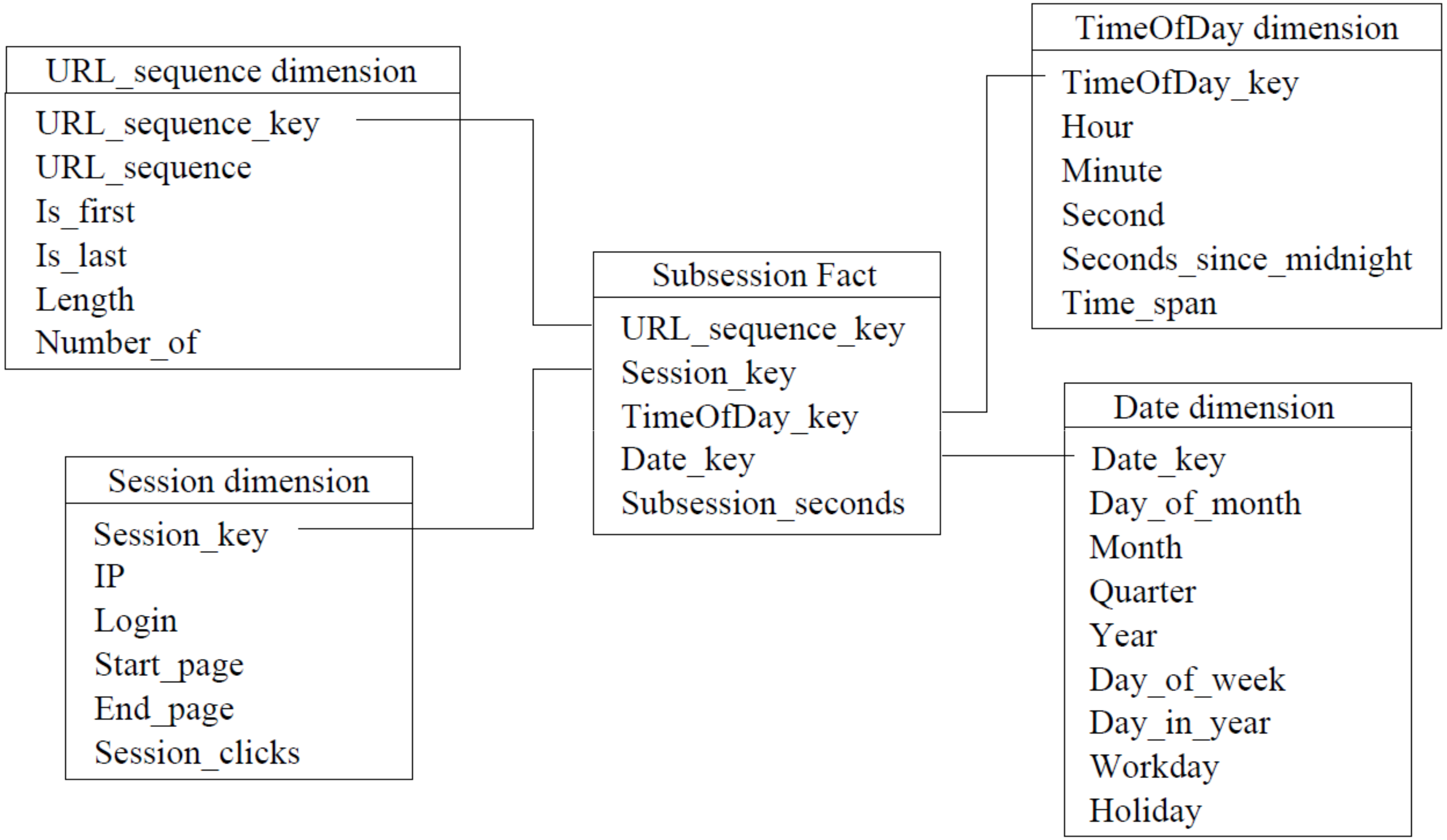
Subsession 3

Subsession 4

Subsession 5

Subsession 6

Subsession Fact Star Schema



URL_sequence dimension

URL_sequence_key
URL_sequence
Is_first
Is_last
Length
Number_of

Session dimension

Session_key
IP
Login
Start_page
End_page
Session_clicks

Subsession Fact

URL_sequence_key
Session_key
TimeOfDay_key
Date_key
Subsession_seconds

TimeOfDay dimension

TimeOfDay_key
Hour
Minute
Second
Seconds_since_midnight
Time_span

Date dimension

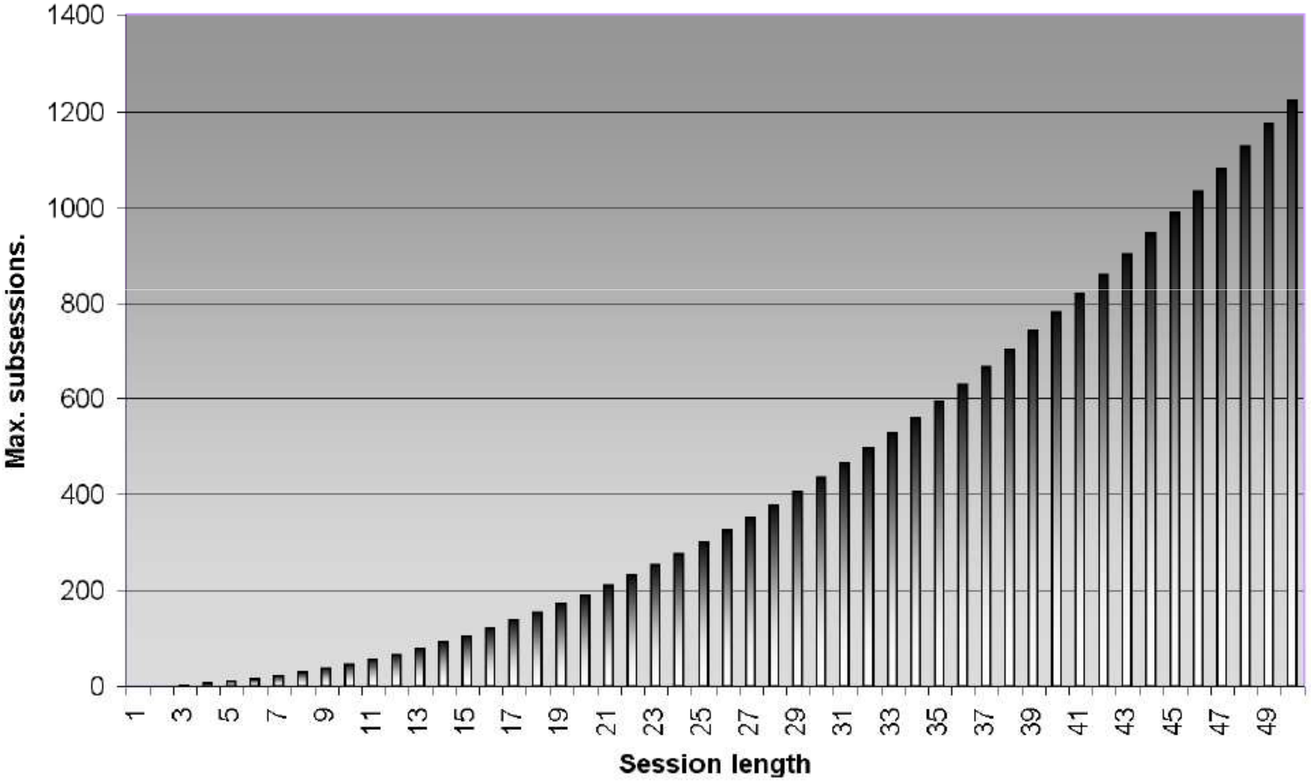
Date_key
Day_of_month
Month
Quarter
Year
Day_of_week
Day_in_year
Workday
Holiday

Figure 2: Subsession Star Schema

Subsession counts

sl = Length of sequence

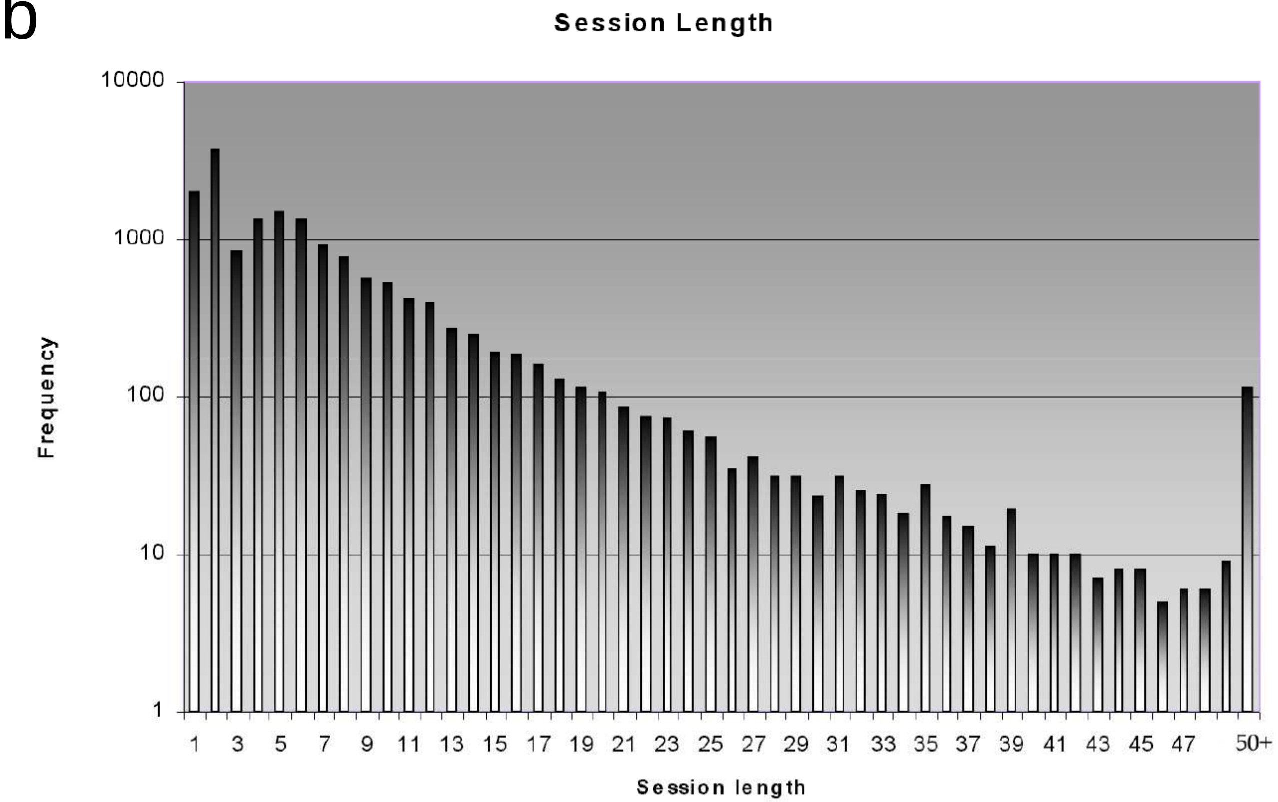
$$maxss(sl) = \sum_{i=1}^{sl}(i-1) \approx 0.5 \times (sl-1)^2$$



Subsessions starting pr click at length 50 is average 24,5

Session length - Frequency

- Average length of click sequences from Nykredits web



When considering session length frequency, the amount of subsession starting pr click is in average 3,56

Subsession Amount Optimisation (1/2)

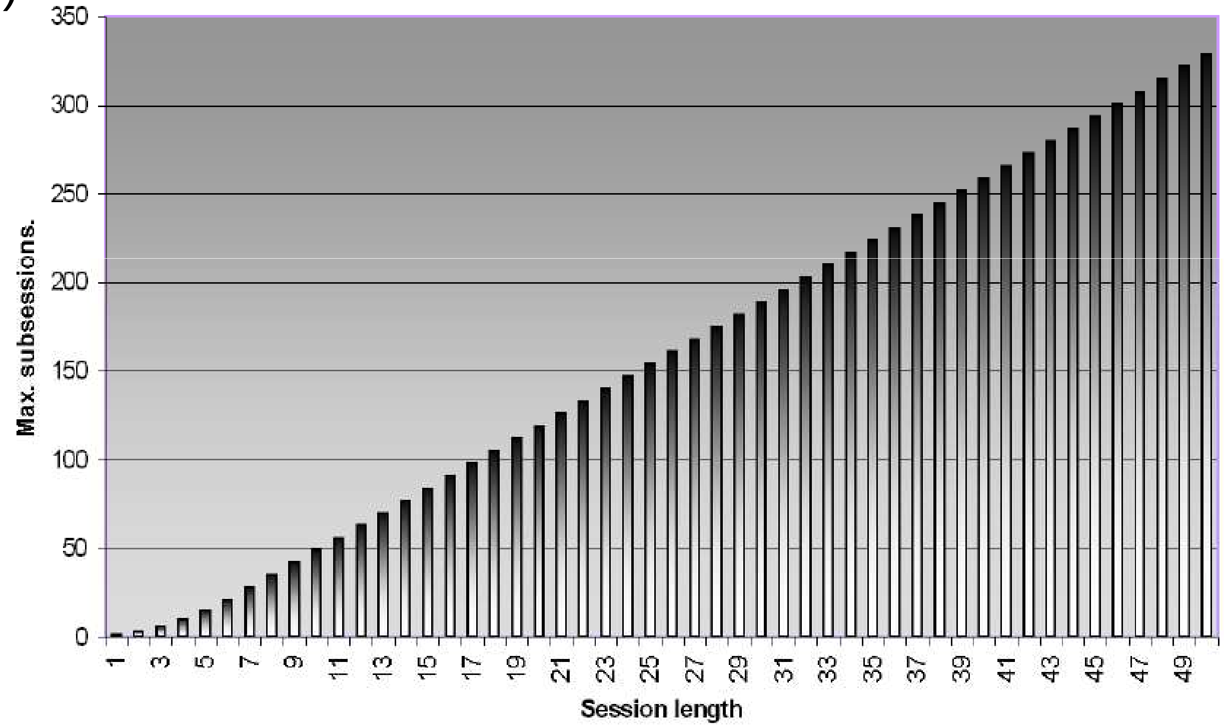
- Optimisation
  - Excluding sessions with 1-2 clicks.
    - Could be users entering the site by mistake.
    - 1 clicks sessions are already excluded by the subsession model
    - 2 clicks sessions are needed in the case
  - Setting a minimum subsession length
    - Not useful in the Nykredit case
    - Cannot answer the banner effiency question
  - Setting a max subsession length
  - Data quality cost
  - Likeliness of a given click sequence, decrease with length

$$ss(sl) = \left(\sum_{i=1}^{min(sl,maxSSL)} (i-1) \right) + (sl - min(sl, maxSSL)) \times (maxSSL - 1)$$

maxSSL = Max length of subssession, sl = length of click sequence

Subsession Amount Optimisation (2/2)

- Up to a 73% reduction in the number of subsessions
- Average amount of subsessions starting pr click is now: 2,78 (22,2% reduction)

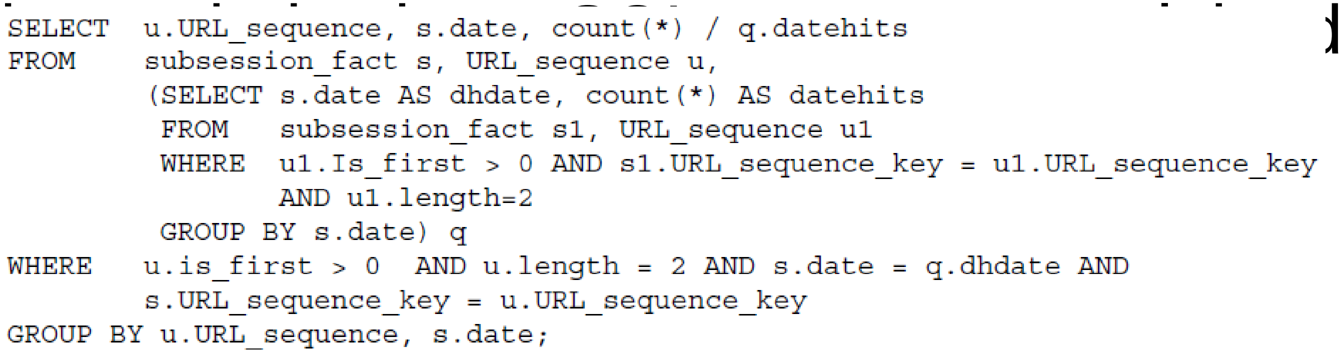


Max subsession length set to 8

Analysis - Banner optimisation

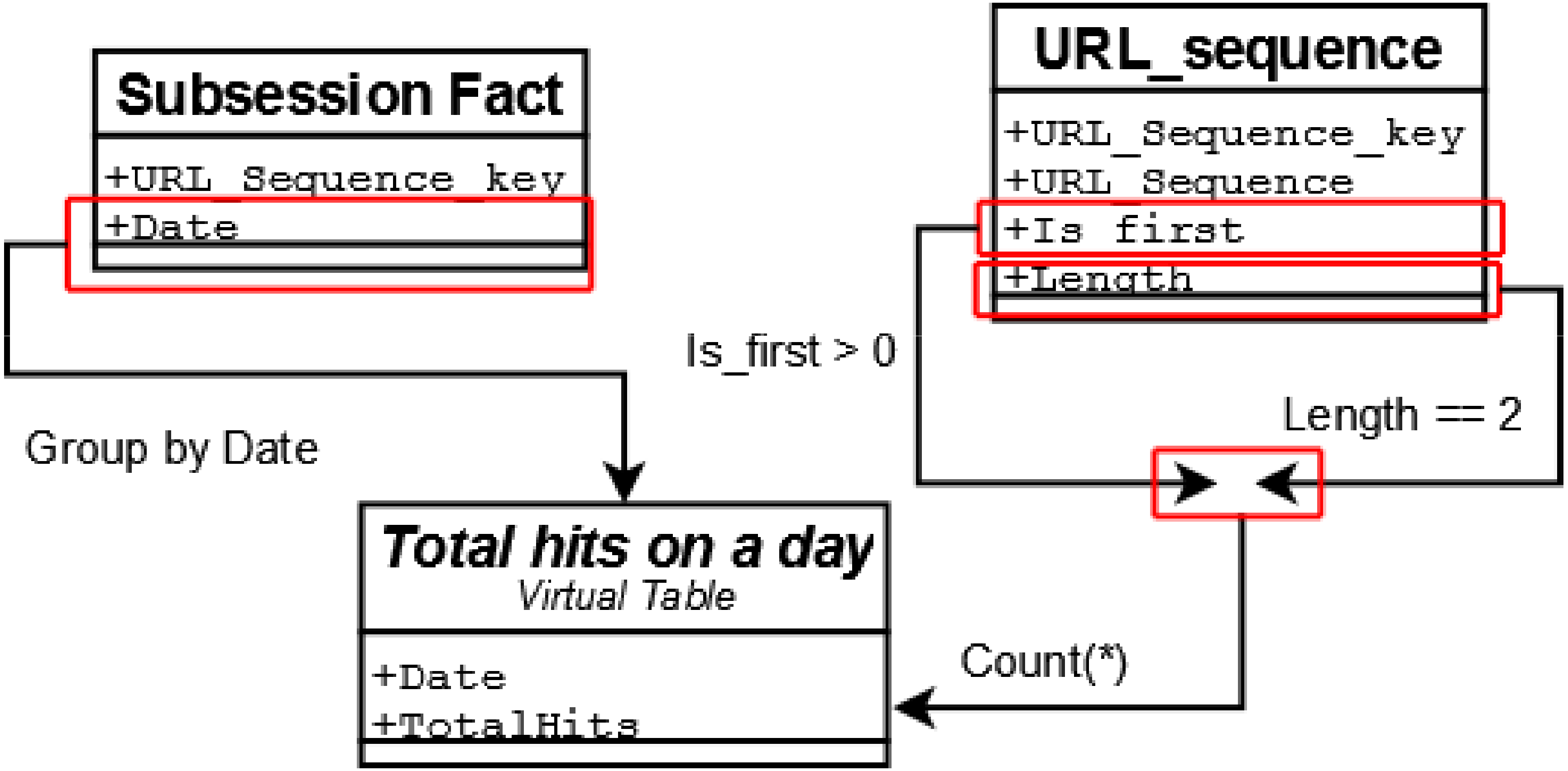
$$\mathit{HitsFromFrontPage}(page, t+1) >> \mathit{HitsFromFrontPage}(page, t-1) \Leftrightarrow$$
$$\text{A banner is set up on the front page, at the time } t\text{, pointing to } page$$

- Banner optimisation
  - Page hits from front page at a given day
  - Hit count has to be larger that a given threshold
  - Frequency of hits on a page at a given day (page, day)
- This d next

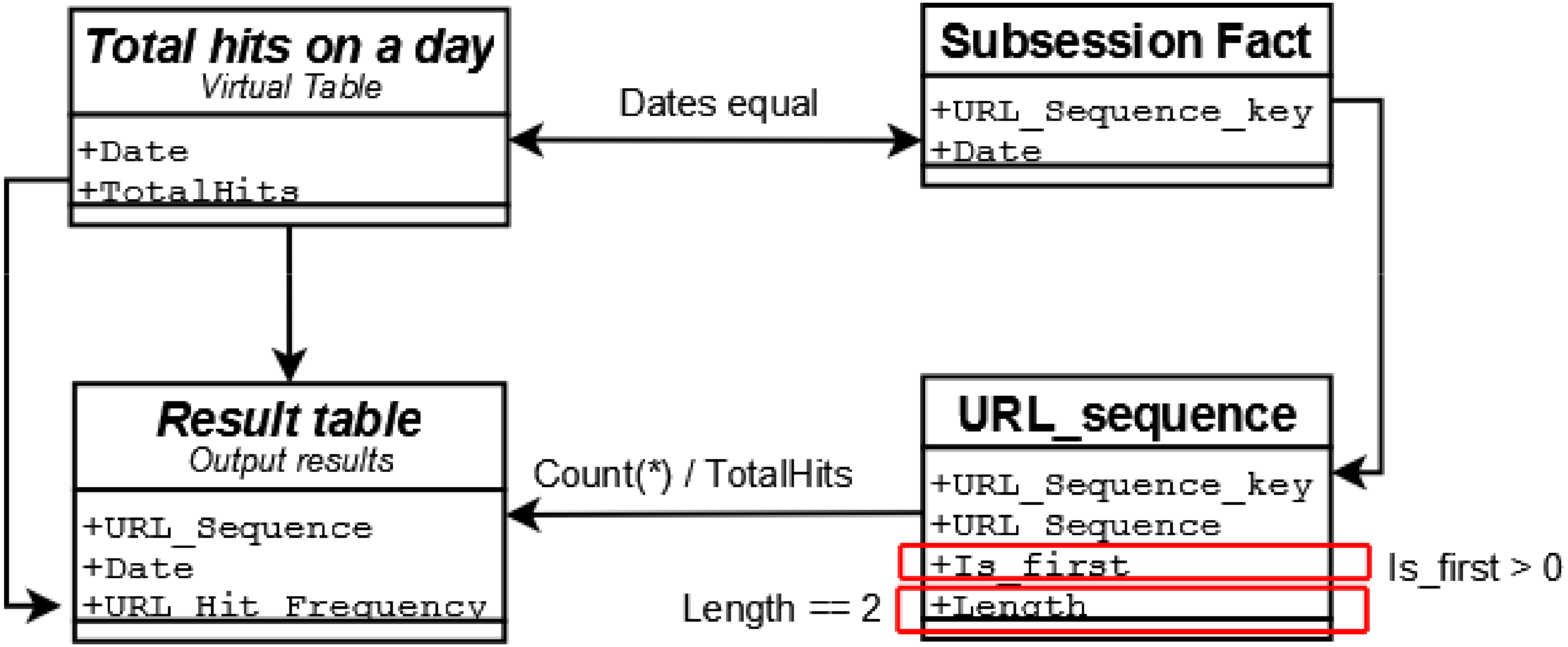


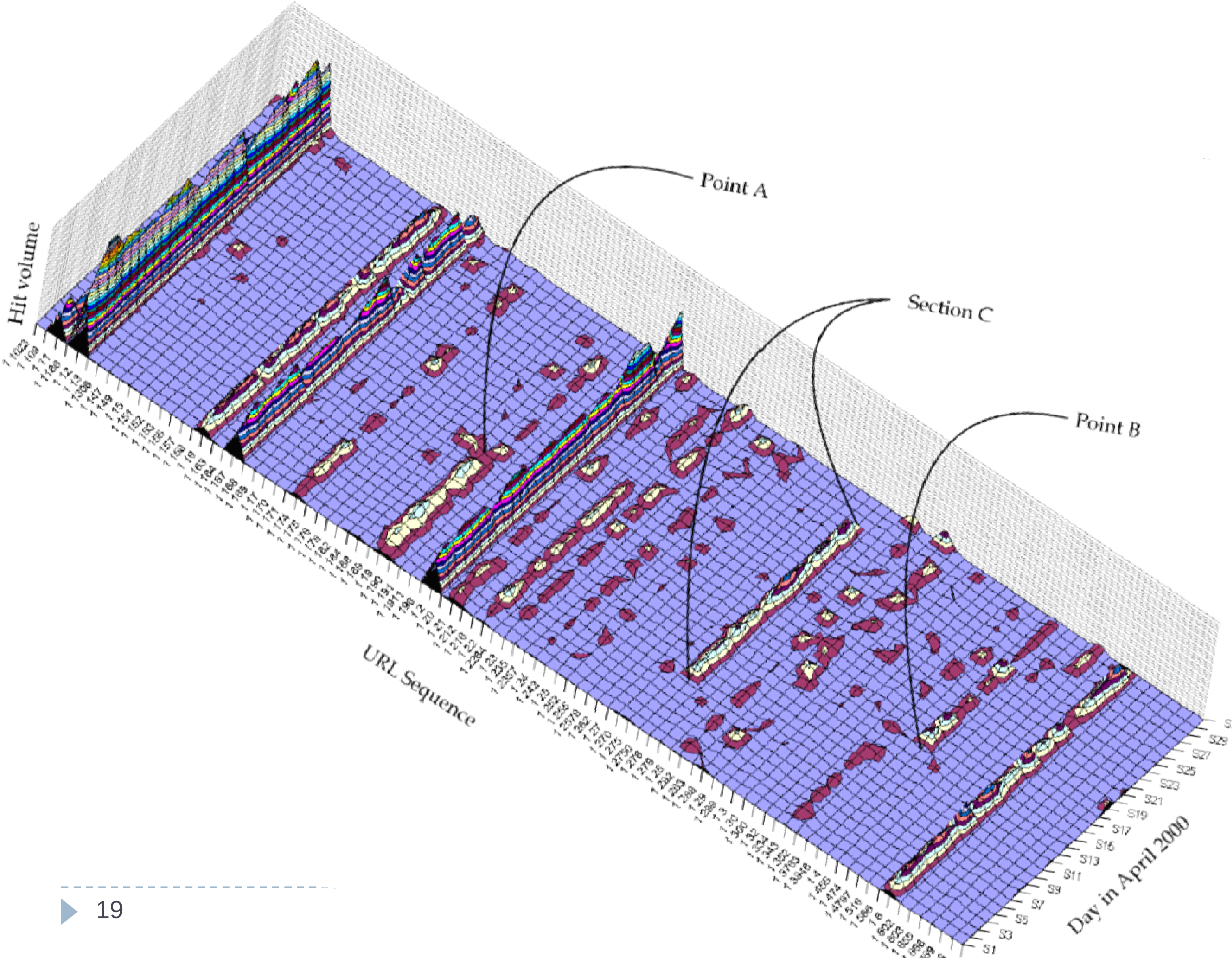
```
SELECT  u.URL_sequence, s.date, count(*) / q.datehits
FROM    subsession_fact s, URL_sequence u,
        (SELECT s.date AS dhdate, count(*) AS datehits
         FROM   subsession_fact s1, URL_sequence u1
         WHERE  u1.Is_first > 0 AND s1.URL_sequence_key = u1.URL_sequence_key
                AND u1.length=2
         GROUP BY s.date) q
WHERE   u.is_first > 0  AND u.length = 2 AND s.date = q.dhdate AND
        s.URL_sequence_key = u.URL_sequence_key
GROUP BY u.URL_sequence, s.date;
```

Example (1/2)

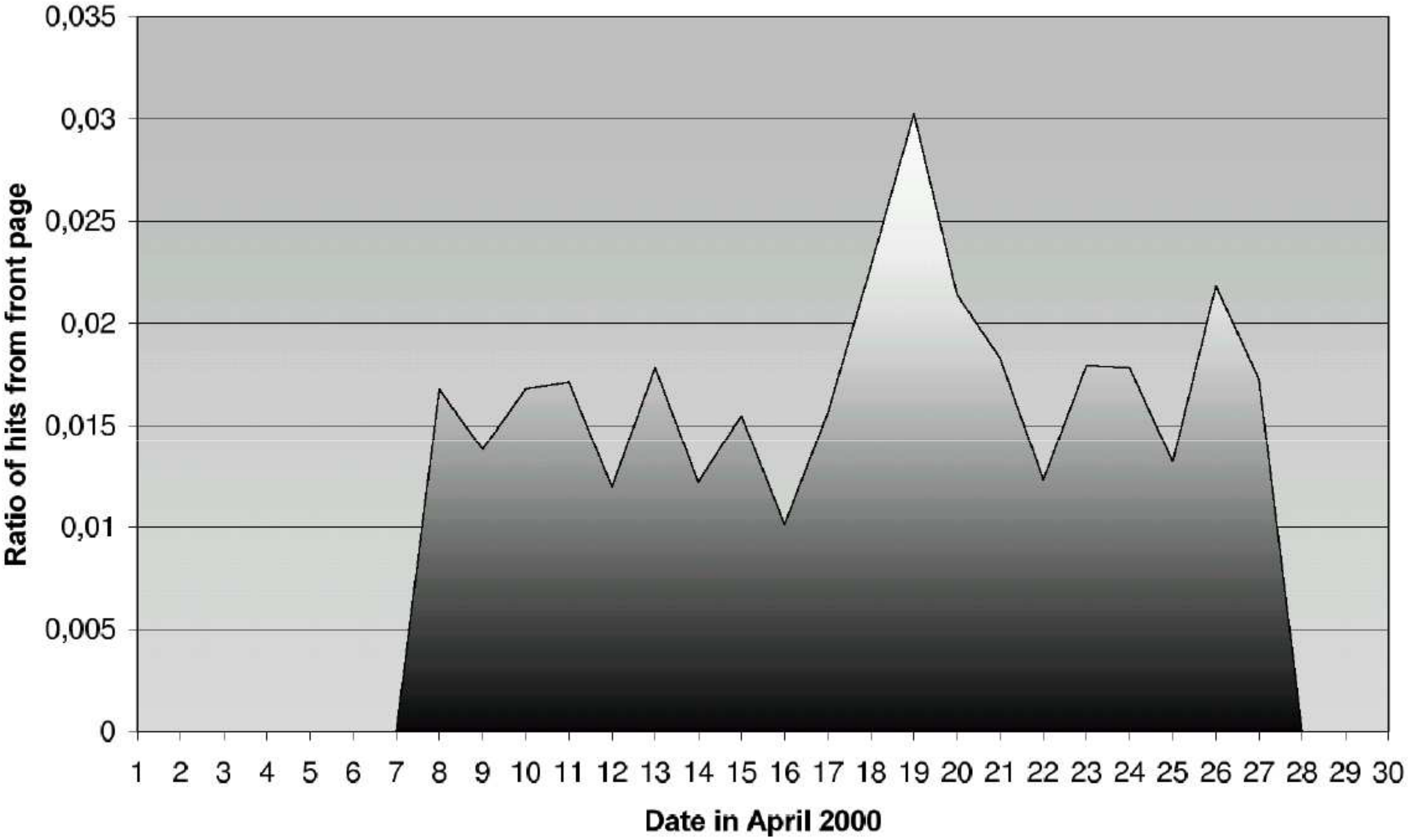

Subsession Fact
+URL_Sequence_key
+Date
URL_sequence
+URL_Sequence_key
+URL_Sequence
+Is_first
+Length
Is_first > 0
Length == 2
Group by Date
Total hits on a day
Virtual Table
+Date
+TotalHits
Count(*)

Example (2/2)


Total hits on a day
Virtual Table
+Date
+TotalHits
Dates equal
Subsession Fact
+URL_Sequence_key
+Date
Result table
Output results
+URL_Sequence
+Date
+URL_Hit_Frequency
Count(*) / TotalHits
URL_sequence
+URL_Sequence_key
+URL_Sequence
+Is_first
+Length
Is_first > 0
Length == 2


Point A
Section C
Point B
Hit volume
URL Sequence
Day in April 2000
S31
S29
S27
S25
S23
S21
S19
S17
S15
S13
S11
S9
S7
S5
S3
S1

Banner analysis 2 – Section C

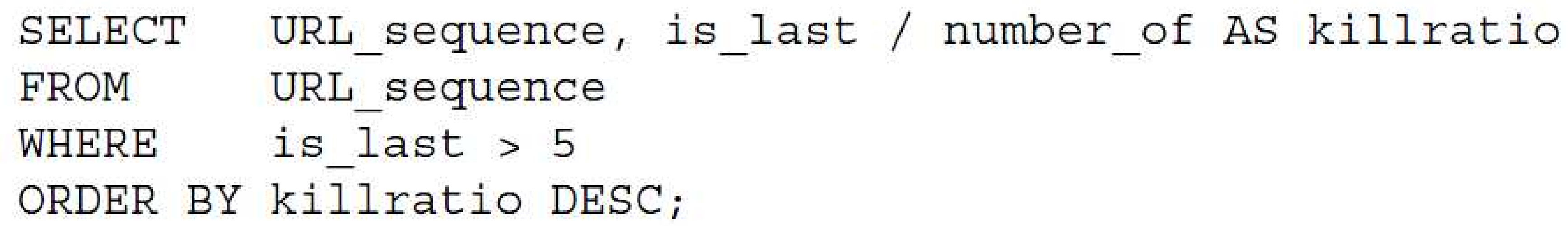


Front page banner advertisement of a online stock trading product.

Analysis - Identifying killer session

$$Sessionkills(ss)/hits(ss) > threshold \Leftrightarrow \text{The subsession } ss \text{ can be a killer subsession.}$$

- Types of killer sessions
  - Pages that have fulfilled its purpose (like a links page)
  - Pages where users switch to encrypted connection
  - Pages where users leave without fulfilling its purpose. True killer
- Can be calculated directly of aggregated data



```
SELECT   URL_sequence, is_last / number_of AS killratio
FROM     URL_sequence
WHERE    is_last > 5
ORDER BY killratio DESC;
```

Killer session graph

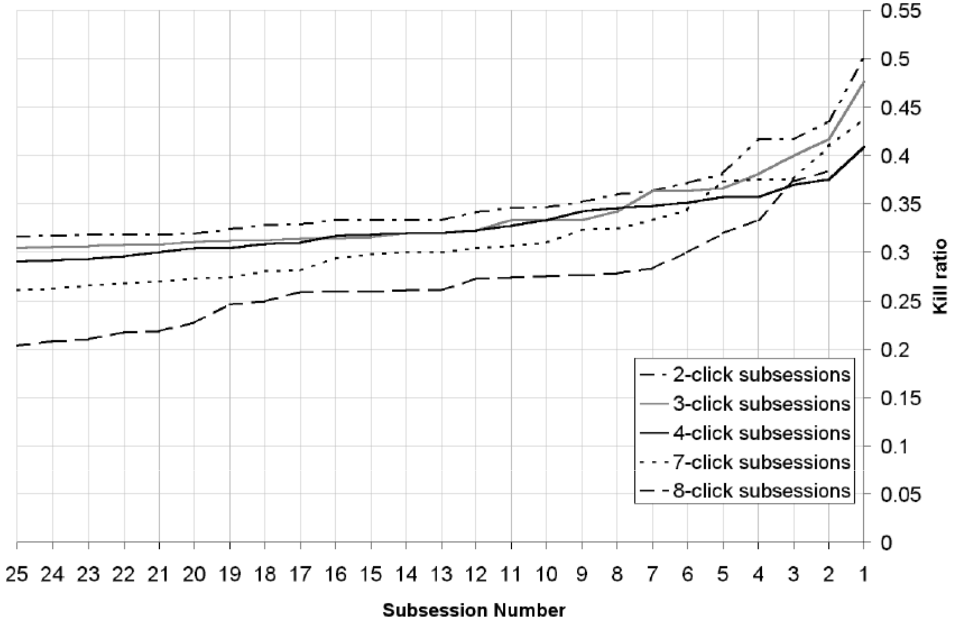


Figure 7: Killer Subsessions

- Subsessions with high kill ratio
  - URL: 13 51 520, Kill ratio 0,47 - Link page to real-estate companies
  - URL: 41 46 … 41 Kill ratio 0,44 - Circular sequence, Loan Calculator
  - URL: 3678 3679 Kill ratio 0,43 - Ends in a gigantic form

Conclusion

- They have proposed a new model
- Successfully shown it can solve the case problems
- Nykredit implemented this model after their were done

- Evaluation
  - Easy to read
  - Good flow
  - Interesting paper

Related to our work

- Our project is about tracking people in an airport
  - At the moment we analyse:
    - Where people spend time (dwell time)
    - We track the dwell time over time
    - The distribution of dwell time over tracking locations.
  - We have not looked that much into sequence analysis
  - But this model could be used for this.