SERIOUS SQL LIVE WEEK 2: 27TH NOV

BY DANNY MA

DATA WITH DANNY

AGENDA:

- Intro (5 mins)
- Identifying Duplicates (25 mins)
- Summary Statistics (30 mins)

WHAT WE COVERED LAST WEEK

SELECT & SORT

- Basic SELECT ✓
- LIMIT result rows ✓
- ORDER BY ASC/DESC date numeric
- Multi column sort ✓ chars

RECORD COUNTS & DISTINCT VALUES

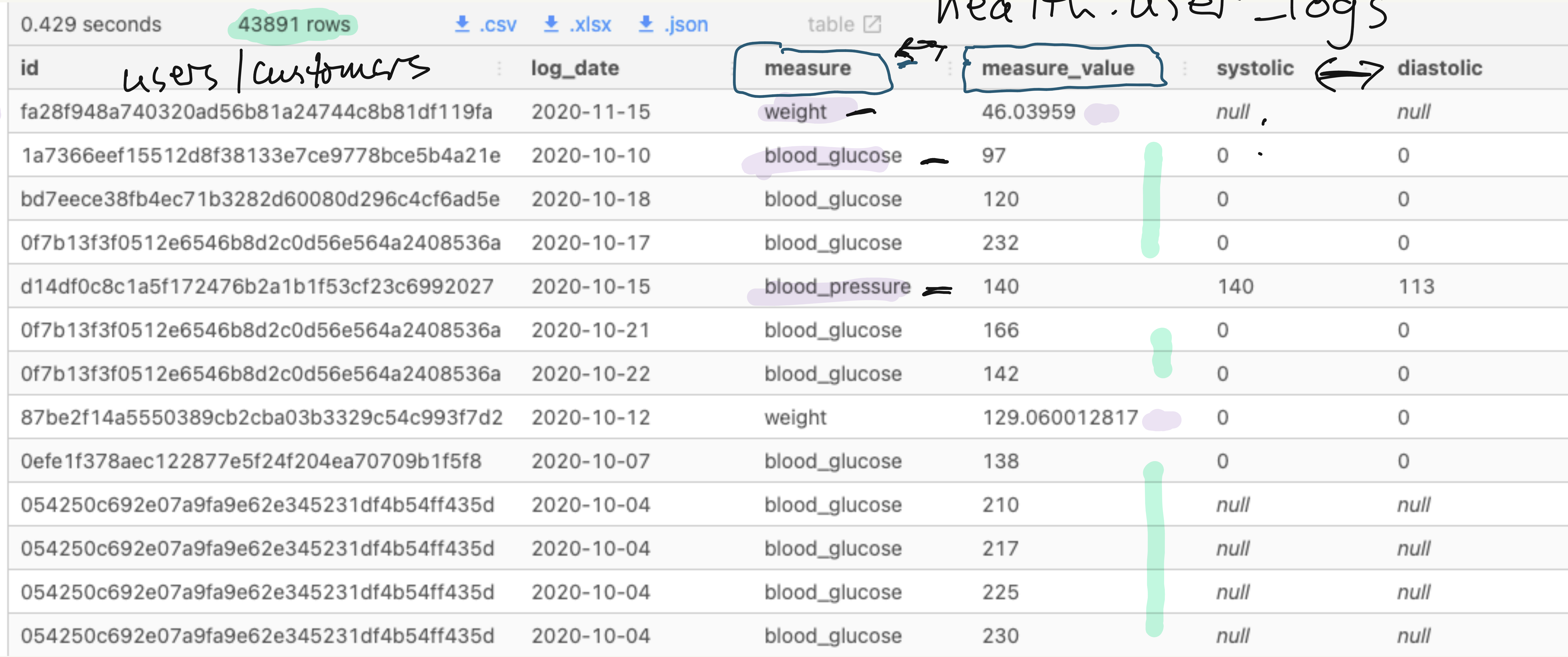
- Column Aliases – AS ...
- DISTINCT * and column(s)
- COUNT DISTINCT – select distinct col1 col2 ... from table
- GROUP BY basics –
- Percentage column

window function

IDENTIFYING DUPLICATES

HEALTH ANALYTICS DATA

health.user_logs



0.429 seconds 43891 rows .csv .xlsx .json table

users / customers

id	log_date	measure	measure_value	systolic	diastolic
fa28f948a740320ad56b81a24744c8b81df119fa	2020-11-15	weight	46.03959	*null*	*null*
1a7366eef15512d8f38133e7ce9778bce5b4a21e	2020-10-10	blood_glucose	97	0	0
bd7eece38fb4ec71b3282d60080d296c4cf6ad5e	2020-10-18	blood_glucose	120	0	0
0f7b13f3f0512e6546b8d2c0d56e564a2408536a	2020-10-17	blood_glucose	232	0	0
d14df0c8c1a5f172476b2a1b1f53cf23c6992027	2020-10-15	blood_pressure	140	140	113
0f7b13f3f0512e6546b8d2c0d56e564a2408536a	2020-10-21	blood_glucose	166	0	0
0f7b13f3f0512e6546b8d2c0d56e564a2408536a	2020-10-22	blood_glucose	142	0	0
87be2f14a5550389cb2cba03b3329c54c993f7d2	2020-10-12	weight	129.060012817	0	0
0efe1f378aec122877e5f24f204ea70709b1f5f8	2020-10-07	blood_glucose	138	0	0
054250c692e07a9fa9e62e345231df4b54ff435d	2020-10-04	blood_glucose	210	*null*	*null*
054250c692e07a9fa9e62e345231df4b54ff435d	2020-10-04	blood_glucose	217	*null*	*null*
054250c692e07a9fa9e62e345231df4b54ff435d	2020-10-04	blood_glucose	225	*null*	*null*
054250c692e07a9fa9e62e345231df4b54ff435d	2020-10-04	blood_glucose	230	*null*	*null*

group by count(*)

EXPLORING A NEW DATASET

limit

select *

- Show first few rows and all cols
- How many records are there?

count(*]

- Any columns of interest?

FURTHER ANALYSIS

- COUNT(*) & COUNT DISTINCT
- Percentage calculations
- Investigate specific values

colums of interest

window function

count(*) percentages

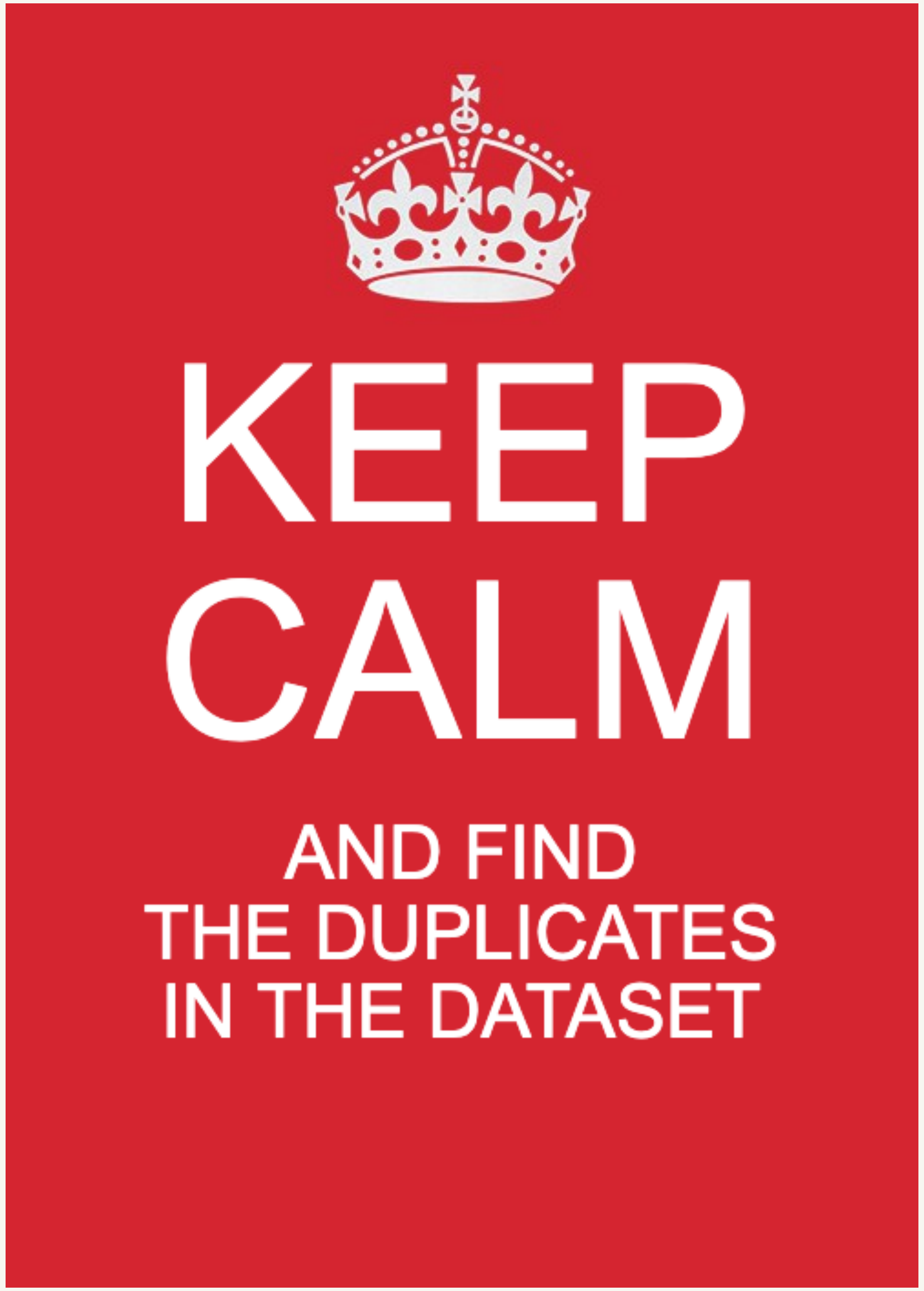
(count(*) / sum(count(*) over ()))

denominator

DATA INSPECTION

WHERE filter

- measure_value = 0
- measure = 'blood_pressue'
- measure & measure_value
- NULL values


KEEP
CALM
AND FIND
THE DUPLICATES
IN THE DATASET

DEAL WITH DUPLICATES

- How can we identify duplicates?
- Should we remove all of them? distinct
- How can we inspect our duplicates?
- Do we actually want to keep them?

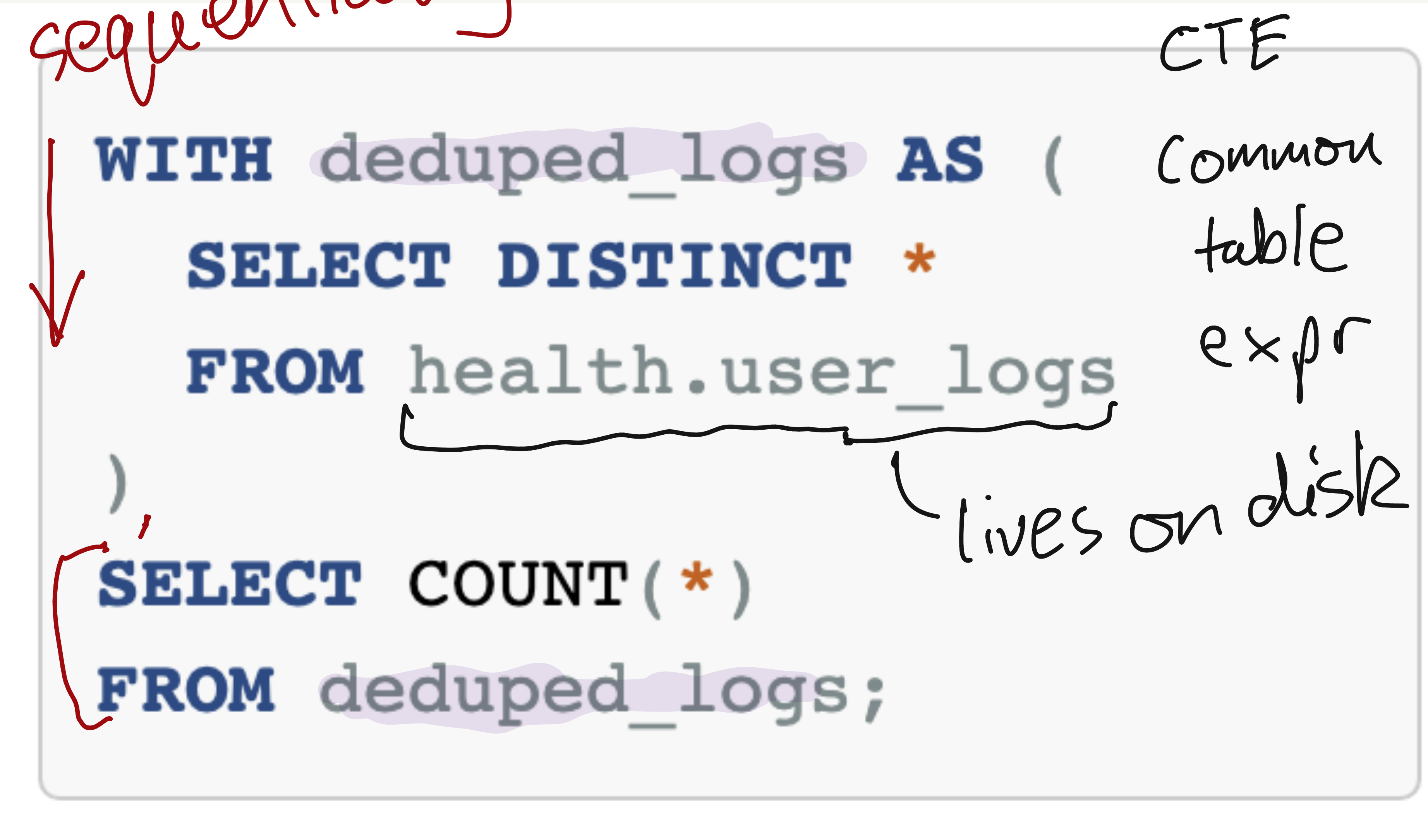
IDENTIFICATION

select distinct *

- Row counts vs distinct row counts
- COUNT(*) VS COUNT(DISTINCT <col>)
- COUNT(*) vs COUNT (DISTINCT *)

CTEs vs SUBQUERY

sequentially



```
WITH deduped_logs AS (
  SELECT DISTINCT *
  FROM health.user_logs
)
SELECT COUNT(*)
FROM deduped_logs;
```

CTE
common
table
expr

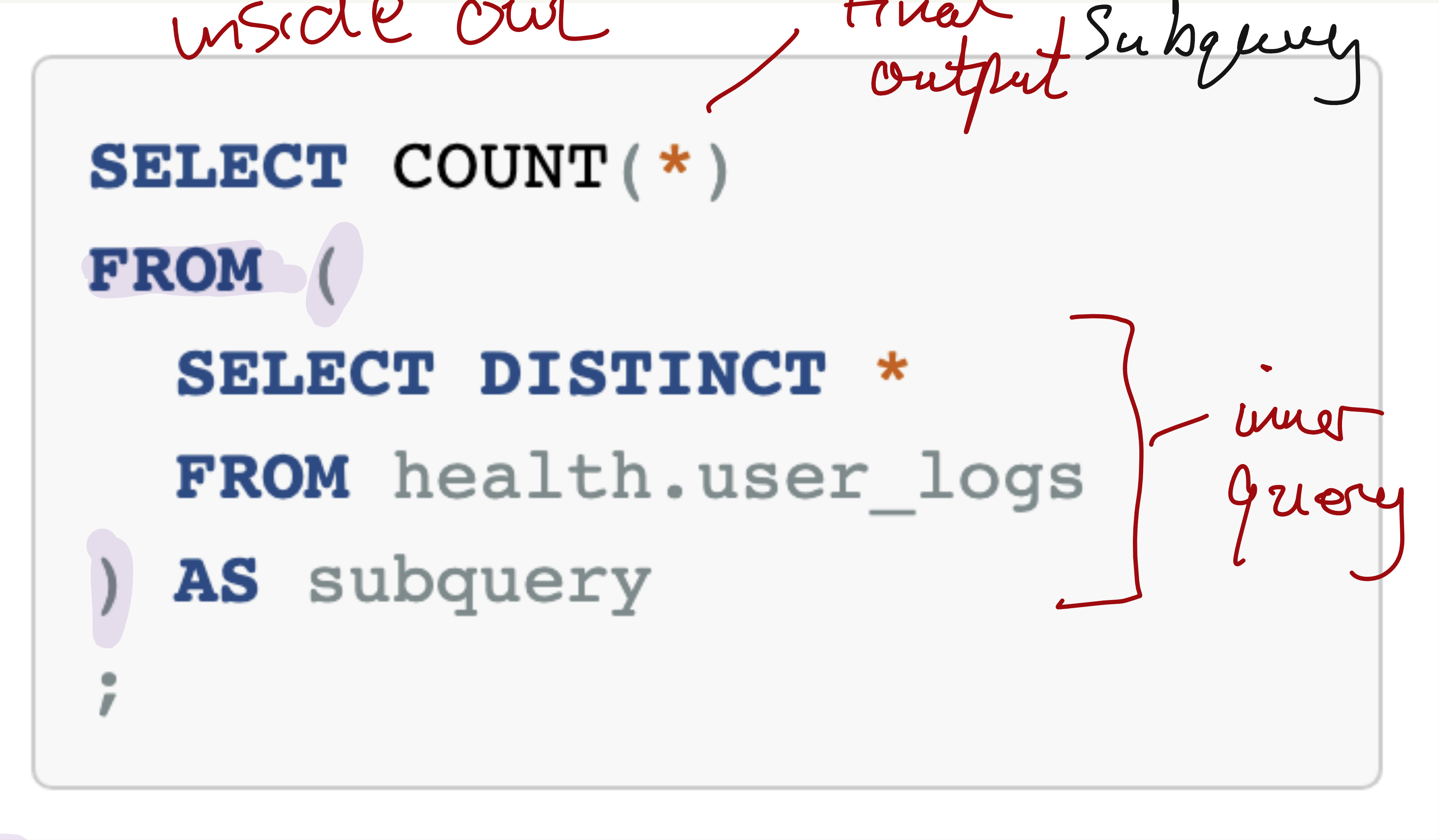
lives on disk

in-memory

inside out

final output

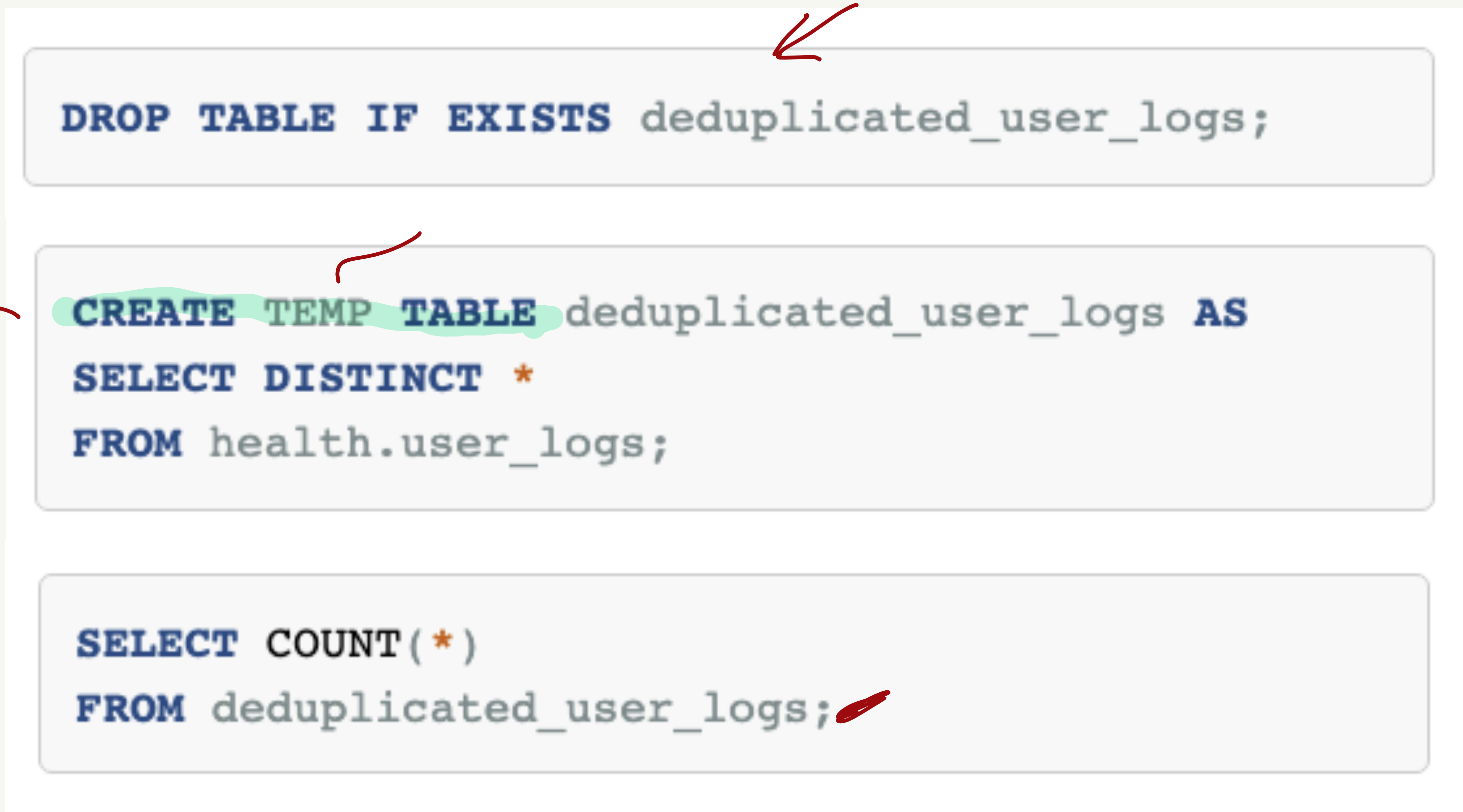
Subquery



```
SELECT COUNT(*)
FROM (
  SELECT DISTINCT *
  FROM health.user_logs
) AS subquery
;
```

inner query

TEMPORARY TABLE



```sql
DROP TABLE IF EXISTS deduplicated_user_logs;
```

```sql
CREATE TEMP TABLE deduplicated_user_logs AS
SELECT DISTINCT *
FROM health.user_logs;
```

```sql
SELECT COUNT(*)
FROM deduplicated_user_logs;
```

write out
to disk
partitions
subfolders
indexes

sequentially

CTEs, SUBQUERIES & TEMP TABLES

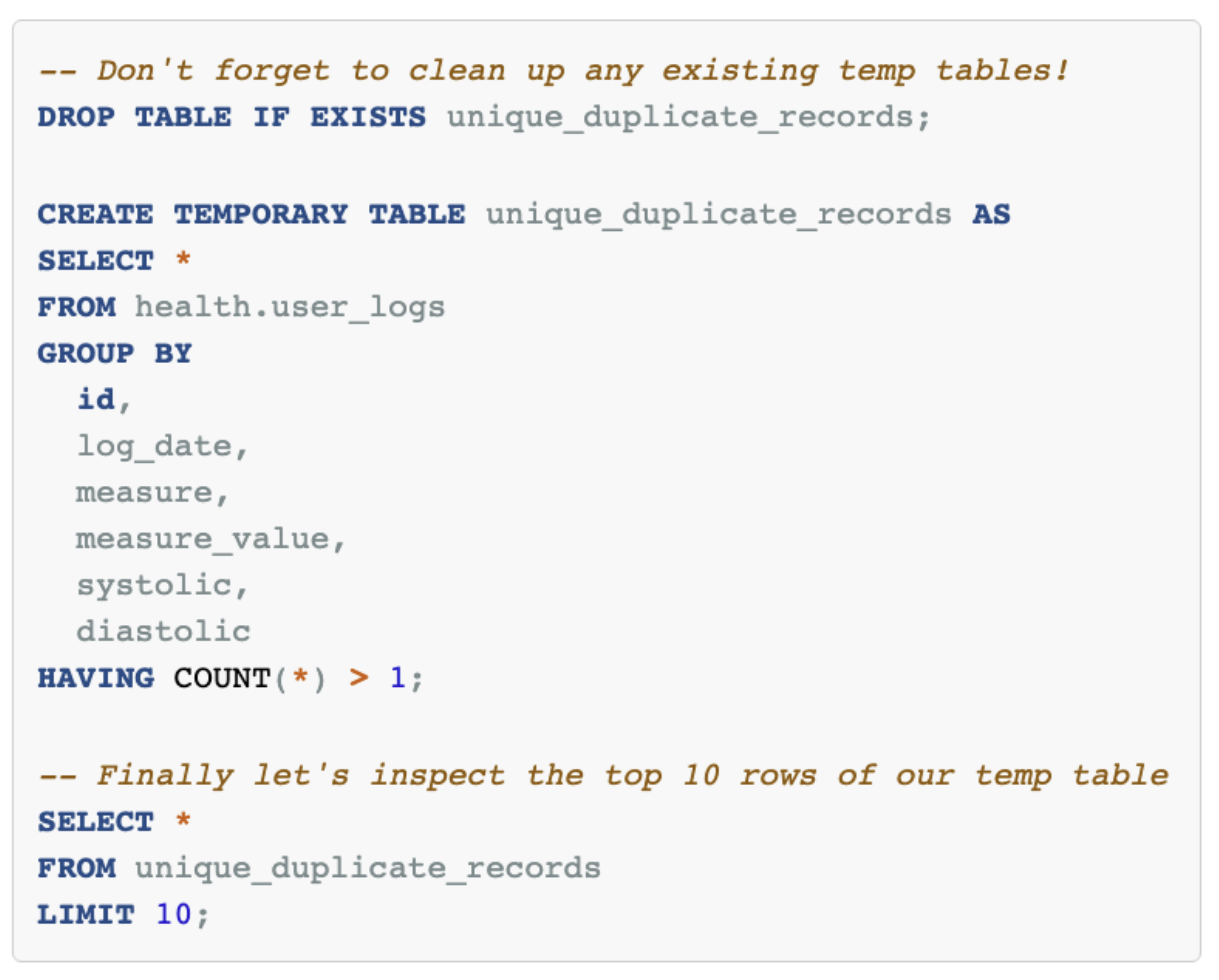
- CTEs : sequential (in-memory)
- Subqueries: inside out (in-memory)
- Temp Tables: sequential (write/read to disk) — control how it's written

→ index, partitions

KEEPING DUPLICATES

- Why do we want to keep duplicates?
- GROUP BY COUNT(*) with all columns
- GROUP BY vs HAVING

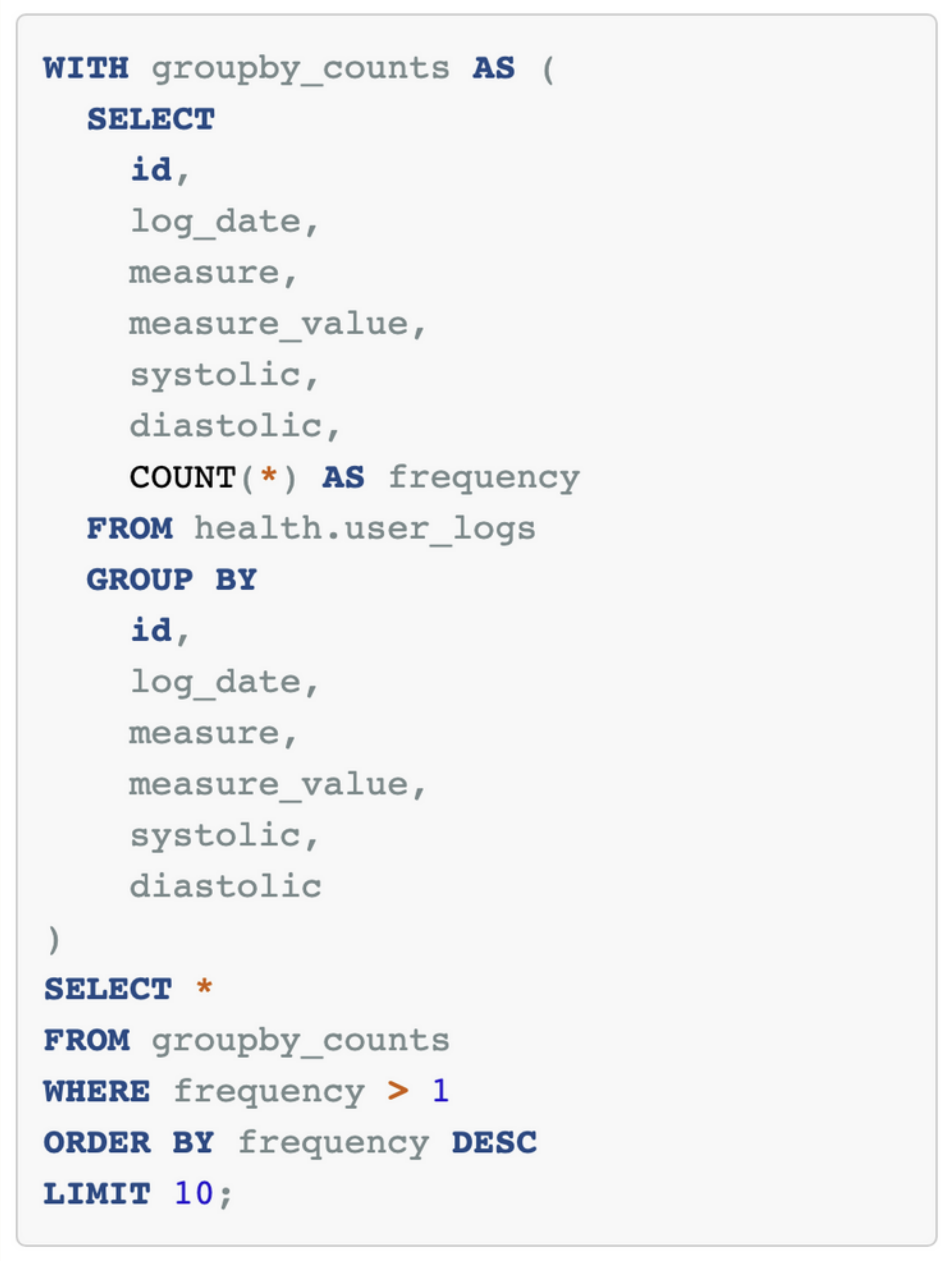
TEMP TABLE VS CTE



```sql
-- Don't forget to clean up any existing temp tables!
DROP TABLE IF EXISTS unique_duplicate_records;

CREATE TEMPORARY TABLE unique_duplicate_records AS
SELECT *
FROM health.user_logs
GROUP BY
  id,
  log_date,
  measure,
  measure_value,
  systolic,
  diastolic
HAVING COUNT(*) > 1;

-- Finally let's inspect the top 10 rows of our temp table
SELECT *
FROM unique_duplicate_records
LIMIT 10;
```



```sql
WITH groupby_counts AS (
  SELECT
    id,
    log_date,
    measure,
    measure_value,
    systolic,
    diastolic,
    COUNT(*) AS frequency
  FROM health.user_logs
  GROUP BY
    id,
    log_date,
    measure,
    measure_value,
    systolic,
    diastolic
)
SELECT *
FROM groupby_counts
WHERE frequency > 1
ORDER BY frequency DESC
LIMIT 10;
```

EXERCISE QUESTION

- Which id value has the most duplicate records in the health.user_logs table?

EXERCISE QUESTION

- Which log_date value had the most duplicate records after removing the max duplicate id value from the previous question?

DUPLICATES SUMMARY

- Remove all duplicates
- Identify and count duplicates
- Keep only duplicates for checking
- WHERE and HAVING clauses
- CTEs vs Subqueries vs Temp Tables

group by
all columns