The Best SQL Interview Preparation Doc

By Deepanshu Kalra ([LinkedIn](https://www.linkedin.com/in/deepanshuk/), [Instagram](http://instagram.com/dkalra_/), [Twitter](https://twitter.com/dkalra_/))

[Read me!](#_heading=h.wmscqrxgkw90) **2**

[Some topics to revise before your interview](#_heading=h.ha12ivudcm2i) **3**

[Data pipeline should address](#_heading=h.u79a4yr5ilj) **4**

[SQL Questions from Top 50 Data Engineer Interview Questions and Answers](#_heading=h.tz029qxk8tob) **5**

[Important Internet Sources](#_heading=h.fykw6vlnu1uz) **6**

[Asked in Interviews](#_heading=h.ukexjioiyms) **8**

[Medium Hard Problems](#_heading=h.ravyqll9gxy2) **14**

[Ending note](#_heading=h.jpb43nz3ybq5) **30**

Expected time to read: 2 days - 3 days

(excluding practice on SQLStar, HackerRank, other resources shared below)

# Read me!

Thanks for all the support and feedback, everyone. You all encouraged me to create this document. I will try to come back and reiterate over it with time

Important: The purpose of this document is for you to brush up your SQL skills before the interview. If you are someone who wants to start from [here](https://www.linkedin.com/feed/update/urn:li:activity:6899568308950040576/). If you are a pro, directly jump to page 14, 8 to practice and read questions at page 5 after that.

Please drop a note over [LinkedIn](https://www.linkedin.com/in/deepanshuk/):

-if you like this document that would really encourage me

-or if you would like to contribute to this document and make it better.

Bonus Links:

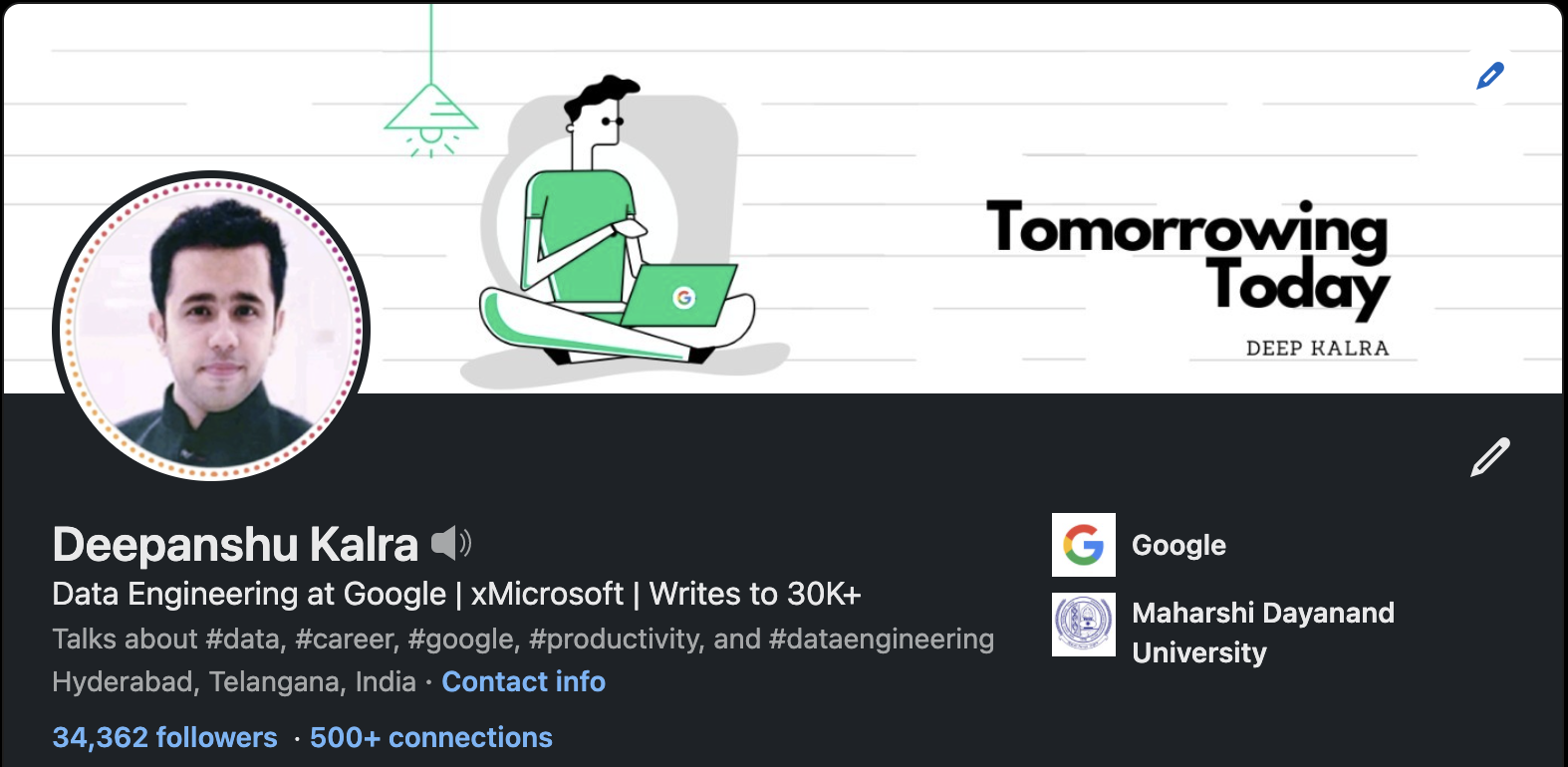
Resume Template: [rebrand.ly/deepTemplate](http://rebrand.ly/deepTemplate)

Data Engineering Skills: [rebrand.ly/deepDEskills](http://rebrand.ly/deepDEskills)

Individual help (Paid): [rebrand.ly/deep1on1](http://rebrand.ly/deep1on1)

Regards and Happy Learning,

Deepanshu Kalra

[](https://www.linkedin.com/in/deepanshuk/)

# Some topics to revise before your interview

Here are a few important links for you to go through before you start your preparation:

| S.no | Topic | Links |
| --- | --- | --- |
| 0 | My Linkedin Posts | Top SQL Interview Topics-  <https://www.linkedin.com/posts/deepanshuk_sql-interview-data-activity-6843425149660758016-HZxA>  SQL practice and studying links, foundation to advance:  1. This doc: [rebrand.ly/deepSQL](http://rebrand.ly/deepSQL)  2. Select \* SQL: [selectstarsql.com/](http://selectstarsql.com/)  3. Leetcode: [lnkd.in/g3c5JGC](http://lnkd.in/g3c5JGC)  4. LinkedIn Learning: [lnkd.in/gQXFc4n](http://lnkd.in/gQXFc4n)  5. Window Functions: [lnkd.in/g3RtPCJ](http://lnkd.in/g3RtPCJ)  6. HackerRank: [lnkd.in/grv\_9sB](http://lnkd.in/grv_9sB)  7. W3 Schools: [lnkd.in/gJPfrrv](http://lnkd.in/gJPfrrv)  8. CodeAcademy: [lnkd.in/gT5xmpN](http://lnkd.in/gT5xmpN)  9. SQLZOO: [sqlzoo.net/](http://sqlzoo.net/)  10. SQL Bolt: [sqlbolt.com/](http://sqlbolt.com/)  11. Danny Ma's SQL: [8weeksqlchallenge.com/](http://8weeksqlchallenge.com/)  12: Interactive SQL: [sqlcourse.com/](http://sqlcourse.com/) |
| 1 | Slowly Changing Dimension | https://datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.html |
| 2 | Joins | <https://i.stack.imgur.com/4zjxm.png>  Also read [self joins](https://www.linkedin.com/posts/deepanshuk_sql-data-dataengineering-activity-6841264716157878272-aoxi)  Practice self join: <https://www.w3resource.com/sql/joins/perform-a-self-join.php> |
| 3 | Physical Joins | https://www.linkedin.com/pulse/loop-hash-merge-join-types-eitan-blumin/ |
| 4 | Star/Snowflake | https://www.guru99.com/star-snowflake-data-warehousing.html |
| 5 | Indexes | <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-2017>  <https://www.red-gate.com/simple-talk/sql/learn-sql-server/sql-server-index-basics/>  <https://www.red-gate.com/simple-talk/sql/database-administration/brads-sure-guide-to-indexes/> |
| 6 | Data warehouse concepts | <https://www.1keydata.com/datawarehousing/dimensional.html> |
| 7 | Practicing sql | HackerRank (SQL)  Leetcode (SQL) (Worth paying for premium for sql as many questions are in premium. I took premium for a month)  <https://pgexercises.com/> |
| 8 | Complex sql queries | <http://www.complexsql.com/complex-sql-queries-examples-with-answers/> |
| 9 | Dimension Types | https://www.edureka.co/blog/types-of-dimension-table/ |
| 10 | Cheat sheet | https://intellipaat.com/mediaFiles/2019/02/SQL-Basic-Cheat-Sheet-1.png |

# Data pipeline should address

* Partial loads (A scenarios where Partial processing of the files or records or any failures of ETL Jobs occurred; to clean up a few records and re-run the job)
* Restart-ability (You have to re-run from a previous successful run because a downstream dependent job failed or reprocess process some data from history. for e.g. We need to run since last Monday or a random date)
* Re-processing the same files (A source issue where they sent multiple files; We need to pick the right records)
* Catch-up loads (In case you missed executing jobs for specific runs and playing catch up; Batch Processing)

# 

# SQL Questions from Top 50 Data Engineer Interview Questions and Answers

1. Write a SQL Query to find Max salary and Department name from each department.
2. Write a SQL query to find records in Table A that are not in Table B without using NOT IN operator.
3. Write SQL Query to find employees that have same name and email.
4. Write a SQL Query to find Max salary from each department.
5. Write SQL query to get the nth highest salary among all Employees.
6. How can you find 10 employees with Odd number as Employee ID?
7. Write a SQL Query to get the names of employees whose date of birth is between 01/01/1990 to 31/12/2000.
8. Write a SQL Query to get the Quarter from date.
9. Write Query to find employees with duplicate email.
10. Is it safe to use ROWID to locate a record in Oracle SQL queries?
11. What is a Pseudocolumn?
12. What are the reasons for de-normalizing the data?
13. What is the feature in SQL for writing If/Else statements?
14. What is the difference between DELETE and TRUNCATE in SQL?
15. What is the difference between DDL and DML commands in SQL?
16. Why do we use Escape characters in SQL queries?
17. What is the difference between Primary key and Unique key in SQL?
18. What is the difference between INNER join and OUTER join in SQL?
19. What is the difference between Left OUTER Join and Right OUTER Join?
20. What is the datatype of ROWID?
21. What is the difference between where clause and having clause?
22. How will you calculate the number of days between two dates in MySQL?
23. What are the different types of Triggers in MySQL?
24. What are the differences between Heap table and temporary table in MySQL?
25. What is a Heap table in MySQL?
26. What is the difference between BLOB and TEXT data type in MySQL?
27. What will happen when AUTO\_INCREMENT on an INTEGER column reaches MAX\_VALUE in MySQL?
28. What are the advantages of MySQL as compared with Oracle DB?
29. What are the disadvantages of MySQL?
30. What is the difference between CHAR and VARCHAR datatype in MySQL?
31. What is the use of 'i\_am\_a\_dummy flag' in MySQL?
32. How can we get current date and time in MySQL?
33. What is the difference between timestamp in Unix and MySQL?
34. How will you limit a MySQL query to display only top 10 rows?
35. What is automatic initialization and updating for TIMESTAMP in a MySQL table?
36. How can we get the list of all the indexes on a table?
37. What is SAVEPOINT in MySQL?
38. What is the difference between ROLLBACK TO SAVEPOINT and RELEASE SAVEPOINT?
39. How will you search for a String in MySQL column?
40. How can we find the version of the MySQL server and the name of the current database by SELECT query?
41. What is the use of IFNULL() operator in MySQL?
42. How will you check if a table exists in MySQL?
43. How will you see the structure of a table in MySQL?
44. What are the objects that can be created by CREATE statement in MySQL?
45. How will you see the current user logged into MySQL connection?
46. How can you copy the structure of a table into another table without copying the data?
47. What is the difference between Batch and Interactive modes of MySQL?
48. How can we get a random number between 1 and 100 in MySQL?

# Important Internet Sources

* Pivot: <https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot?view=sql-server-ver15>
* <https://www.teamblind.com/post/Facebook-DE-decision-wzQRWoCS> (Do topics from here as well)
* Analytical function
  + <https://www.red-gate.com/simple-talk/sql/oracle/introduction-to-analytic-functions-part-1-2/>
  + <https://www.red-gate.com/simple-talk/sql/oracle/introduction-to-analytic-functions-part-2/>
* Windows function
  + <https://www.red-gate.com/simple-talk/sql/learn-sql-server/window-functions-in-sql-server/>
  + <https://www.red-gate.com/simple-talk/sql/learn-sql-server/window-functions-in-sql-server-part-2-the-frame/>
* Indexes
* Columnstore indexes
* Datawarehouse
  + Star, snowflake
  + Types of dimension
  + Types of facts
  + Modeling of databases
  + OLAP vs OLTP - <https://academy.vertabelo.com/blog/oltp-vs-olap-whats-difference/>
  + <https://www.imaginarycloud.com/blog/oltp-vs-olap/>
  + <https://www.vertabelo.com/blog/a-unified-view-on-database-normal-forms-from-the-boyce-codd-normal-form-to-the-second-normal-form-2nf-3nf-bcnf/>
* Basics of Redshift
  + <https://s3-eu-west-1.amazonaws.com/cdn.jefclaes.be/amazon-redshift-fundamentals/aws-redshift-fundamentals.html>
  + <https://www.youtube.com/watch?v=TFLoCLXulU0>
  + <https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-techniques-for-amazon-redshift/>

# 

# Asked in Interviews

| Glassdoor: <https://www.glassdoor.co.in/Interview/Facebook-Data-Engineer-Interview-Questions-EI_IE40772.0,8_KO9,22_IP3.htm?filter.jobTitleFTS=Data+Engineer>  [Must do]  products sales  +------------------+---------+ +------------------+---------+  | product\_id | int |------->| product\_id | int |  | product\_class\_id | int | +---->| store\_id | int |  | brand\_name | varchar | | +->| customer\_id | int |  | product\_name | varchar | | | | promotion\_id | int |  | price | int | | | | store\_sales | decimal |  +------------------+---------+ | | | store\_cost | decimal |  | | | units\_sold | decimal |  | | | transaction\_date | date |  | | +------------------+---------+  | |  stores | | customers  +-------------------+---------+ | | +---------------------+---------+  | store\_id | int |-+ +--| customer\_id | int |  | type | varchar | | first\_name | varchar |  | name | varchar | | last\_name | varchar |  | state | varchar | | state | varchar |  | first\_opened\_date | datetime| | birthdate | date |  | last\_remodel\_date | datetime| | education | varchar |  | area\_sqft | int | | gender | varchar |  +-------------------+---------+ | date\_account\_opened | date |  +---------------------+---------+  Question 1:  What brands have an average price above $3 and contain at least 2 different products?  Question 2:  To improve sales, the marketing department runs various types of promotions.  The marketing manager would like to analyze the effectiveness of these promotion campaigns.  In particular, what percent of our sales transactions had a valid promotion applied?  Question 3:  We want to run a new promotion for our most successful category of products  (we call these categories “product classes”).  Can you find out what are the top 3 selling product classes by total sales?  Question 4:  We are considering running a promo across brands. We want to target  customers who have bought products from two specific brands.  Can you find out which customers have bought products from both the  “Fort West" and the "Golden" brands? |
| --- |

* + One table has date and salesamount. Output a table which has both the above columns with cumulative month s sales amount as an additional column
  + Relational data modeling and dimensional data modeling diff
  + how to distribute storage while creating the table
  + if I have a data model which has a lot of dimension how can I simplify it <https://stackoverflow.com/questions/27690617/star-schema-structure-to-many-dimensions>
  + SCD types. if I have a table which has a lot of attributes column but only few changes frequently how can I capture these changes
  + Diff between oltp and master data   
    <https://metamug.com/article/difference-between-master-and-transaction-table.html>
  + how can we implement normalization
  + Table Questions
    - Find cumulative sum of values from a table of dept, item and value
    - From same table, find item with maximum value in each dept?
  + Create table of fixtures from below table of countries

Country

Ind

Aus

SA

Result:

c1 | c2

ind | aus

aus | sa

sa | ind

* + INPUT:

Asin day is\_instock

A1 1 0

A1 2 0

A1 3 1

A1 4 1

A1 5 0

Output:

asin start\_day end\_day is\_instock

a1 1 2 0

a1 3 4 1

a1 5 5 0

* + There is a list of countries say IND, PAK, CHN, AFG, SRI, BNG. Create a combination of countries with the help of this list using one query  
    How about IND-PAK & PAK-IND duplicate, this is where people get stuck? Could not arrive at the solution or approach
  + Which range has most visitors
    - TBL1: <start\_dt> <end\_dt>
    - TBL2: <date> <num\_of\_visitors>
  + How to delete Duplicate Records from a table considering there is no primary key. For example, consider the table below

id

1

1

1

2

2

* + You have two tables:

A

id

1

1

1

1

1

B

id

1

1

* + - Select count(\*) from A INNER JOIN B On A.id = B.Id [ans] 2 correct is 10
    - Select count(\*) from A LEFT OUTER JOIN B On A.id = B.Id [ans] 5 correct is 10
    - Select count(\*) from A RIGHT OUTER JOIN B On A.id = B.Id [ans]2 correct is 10
  + You have table i.e. customer with details

cust\_id | mem\_start\_date | mem\_end\_date |

-------|-----------------|---------------------|

| 114 | 2015-01-01 | 2015-02-15 |

| 116 | 2014-12-01 | 2015-03-15 |

| 120 | 2015-02-15 | 2015-04-01 |

| 221 | 2015-01-15 | 2015-10-01 |

| 120 | 2015-05-15 | 2015-07-01 |

-------------|-----------------------|--------------------|

* + - Give me SQL QUERY that can produce list of active customers till date?
    - Give me SQL Query that can Produce list of active customers for month of January 2015?
  + You have a table i.e shipments\_details

Shipments Table:

shipment\_id| shipment\_date | delvry\_date |

114 | 2015-01-01 | 2015-01-02 |

116 | 2015-02-01 | 2015-02-01 |

120 | 2015-02-15 | 2015-02-16 |

221 | 2015-03-15 | 2015-03-18 |

120 | 2015-05-15 | 2015-06-01 |

+---------------+--------------------+-----------------+

* + - Give me SQL QUERY that can give produce output to draw graph between DeliveredShipment v/s ShippedShipment for last 7 Days?
  + Write a SQL query that can give following output in two columns.
    - Count of negative numbers || Count of the positive numbers

id

1

-1

1

-1

1

1

-1

1

-1

1

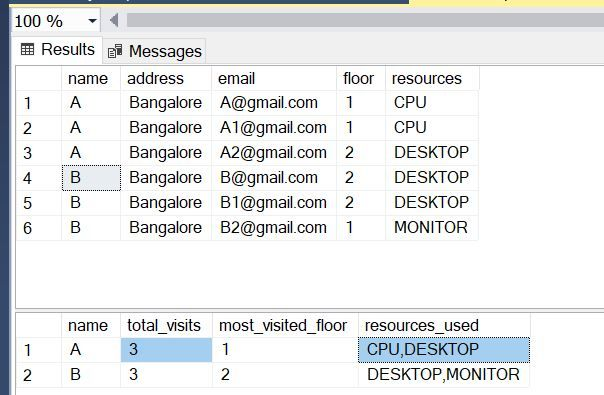
* + Sum of salaries per department for current and previous month

Dept1 PreviousMonthTotal CurrentMonthTotal

1 100 2000

2 ..

* Complex queries example
  + Second highest salaried person in each dept – Done
  + Backfilling problem
  + Rank – Done
  + Dense rank – Done
  + Row number – Done
  + Running sum – Done
  + Delete rows in table so that out of duplicate rows only singled value rows are left
  + DML DDL DQL
  + Diff between truncate delete and drop
  + Fragmentation
  + Types of constraints
  + Acid property
  + Diff between temp table and cte, table variables
  + Which is more efficient? CTE or temp tables?
  + Recursive CTE – To find the hierarchy levels
* Partitioning of table
  + <https://www.cathrinewilhelmsen.net/2015/04/12/table-partitioning-in-sql-server/>
* Normalization
  + Normalization of OLTP
  + Normalization of star and snowflake schema
  + <http://www.sqa.org.uk/e-learning/MDBS01CD/page_01.htm>
* <https://mindmajix.com/data-modeling-interview-questions>
* <https://mindmajix.com/sql-server-interview-questions>
* <https://www.softwaretestinghelp.com/data-modeling-interview-questions-answers/>
* Output Clause
  + <http://www.sqlservercentral.com/articles/T-SQL/156204/>
* <https://www.codeproject.com/Articles/34372/Top-10-steps-to-optimize-data-access-in-SQL-Server>
* <https://biginterview.com/blog/2014/09/sql-interview-questions.html>
* <https://www.upwork.com/i/interview-questions/sql/>
* General architectural questions around Data Pipelines
  + <https://medium.com/@mrashish/design-strategies-for-building-big-data-pipelines-4c11affd47f3>
* <https://www.agent.media/grow/sql-interview-questions/>
* <https://www.toptal.com/sql/interview-questions>
* <http://www.java67.com/2013/04/10-frequently-asked-sql-query-interview-questions-answers-database.html>
* <https://begriffs.com/posts/2018-01-01-sql-keys-in-depth.html>
* <https://www.youtube.com/watch?v=9gOw3joU4a8&list=PL9ooVrP1hQOEDSc5QEbI8WYVV_EbWKJwX>
* <https://docs.aws.amazon.com/redshift/latest/dg/c-the-query-plan.html>
* <https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-techniques-for-amazon-redshift/>
* <https://aws.amazon.com/blogs/big-data/amazon-redshift-engineerings-advanced-table-design-playbook-preamble-prerequisites-and-prioritization/>
* <https://365datascience.com/data-architect-interview-questions/>

Image up: input, image down: output. Write SQL

# 

# 

# Medium Hard Problems

Self-Join Practice Problems (Credits: [Zachary](https://www.linkedin.com/in/thomaszi/) Thomas’s [Doc](https://quip.com/2gwZArKuWk7))

#1: MoM Percent Change

Context: Oftentimes it's useful to know how much a key metric, such as monthly active users, changes between months. Say we have a table logins in the form:

| user\_id | date |

|---------|------------|

| 1 | 2018-07-01 |

| 234 | 2018-07-02 |

| 3 | 2018-07-02 |

| 1 | 2018-07-02 |

| ... | ... |

| 234 | 2018-10-04 |

Task: Find the month-over-month percentage change for monthly active users (MAU).

*Solution:*

*(This solution, like other solution code blocks you will see in this doc, contains comments about SQL syntax that may differ between flavors of SQL or other comments about the solutions as listed)*

WITH mau AS

(

SELECT

/\*

\* Typically, interviewers allow you to write psuedocode for date functions

\* i.e. will NOT be checking if you have memorized date functions.

\* Just explain what your function does as you whiteboard

\*

\* DATE\_TRUNC() is available in Postgres, but other SQL date functions or

\* combinations of date functions can give you a identical results

\* See<https://www.postgresql.org/docs/9.0/functions-datetime.html#FUNCTIONS-DATETIME-TRUNC>

\*/

DATE\_TRUNC('month', date) month\_timestamp,

COUNT(DISTINCT user\_id) mau

FROM

logins

GROUP BY

DATE\_TRUNC('month', date)

)

SELECT

/\*

\* You don't literally need to include the previous month in this SELECT statement.

\*

\* However, as mentioned in the "Tips" section of this guide, it can be helpful

\* to at least sketch out self-joins to avoid getting confused which table

\* represents the prior month vs current month, etc.

\*/

a.month\_timestamp previous\_month,

a.mau previous\_mau,

b.month\_timestamp current\_month,

b.mau current\_mau,

ROUND(100.0\*(b.mau - a.mau)/a.mau,2) AS percent\_change

FROM

mau a

JOIN

/\*

\* Could also have done `ON b.month\_timestamp = a.month\_timestamp + interval '1 month'`

\*/

mau b ON a.month\_timestamp = b.month\_timestamp - interval '1 month'

#2: Tree Structure Labeling

Context: Say you have a table tree with a column of nodes and a column corresponding parent nodes

node parent

1 2

2 5

3 5

4 3

5 NULL

Task: Write SQL such that we label each node as a “leaf”, “inner” or “Root” node, such that for the nodes above we get:

node label

1 Leaf

2 Inner

3 Inner

4 Leaf

5 Root

A solution which works for the above example will receive full credit, although you can receive extra credit for providing a solution that is generalizable to a tree of any depth (not just depth = 2, as is the case in the example above).

(Side note: [this link](http://ceadserv1.nku.edu/longa//classes/mat385_resources/docs/trees.html) has more details on Tree data structure terminology. Not needed to solve the problem though!)

*Solution:*

Note: This solution works for the example above with tree depth = 2, but is not generalizable beyond that.

WITH join\_table AS

(

SELECT

a.node a\_node,

a.parent a\_parent,

b.node b\_node,

b.parent b\_parent

FROM

tree a

LEFT JOIN

tree b ON a.parent = b.node

)

SELECT

a\_node node,

CASE

WHEN b\_node IS NULL and b\_parent IS NULL THEN 'Root'

WHEN b\_node IS NOT NULL and b\_parent IS NOT NULL THEN 'Leaf'

ELSE 'Inner'

END AS label

FROM

join\_table

An alternate solution, that is generalizable to any tree depth:

Acknowledgement: this more generalizable solution was contributed by Fabian Hofmann on 5/2/20. Thank, FH!

WITH join\_table AS

(

SELECT

cur.node,

cur.parent,

COUNT(next.node) AS num\_children

FROM

tree cur

LEFT JOIN

tree next ON (next.parent = cur.node)

GROUP BY

cur.node,

cur.parent

)

SELECT

node,

CASE

WHEN parent IS NULL THEN "Root"

WHEN num\_children = 0 THEN "Leaf"

ELSE "Inner"

END AS label

FROM

join\_table

An alternate solution, without explicit joins:

Acknowledgement: William Chargin on 5/2/20 noted that WHERE parent IS NOT NULL is needed to make this solution return Leaf instead of NULL. Thanks, WC!

SELECT

node,

CASE

WHEN parent IS NULL THEN 'Root'

WHEN node NOT IN

(SELECT parent FROM tree WHERE parent IS NOT NULL) THEN 'Leaf'

WHEN node IN (SELECT parent FROM tree) AND parent IS NOT NULL THEN 'Inner'

END AS label

from

tree

#3: Retained Users Per Month (multi-part)

Acknowledgement: this problem is adapted from SiSense’s [“Using Self Joins to Calculate Your Retention, Churn, and Reactivation Metrics”](https://www.sisense.com/blog/use-self-joins-to-calculate-your-retention-churn-and-reactivation-metrics/) blog post

Part 1:

Context: Say we have login data in the table logins:

| user\_id | date |

|---------|------------|

| 1 | 2018-07-01 |

| 234 | 2018-07-02 |

| 3 | 2018-07-02 |

| 1 | 2018-07-02 |

| ... | ... |

| 234 | 2018-10-04 |

Task: Write a query that gets the number of retained users per month. In this case, retention for a given month is defined as the number of users who logged in that month who also logged in the immediately previous month.

*Solution:*

SELECT

DATE\_TRUNC('month', a.date) month\_timestamp,

COUNT(DISTINCT a.user\_id) retained\_users

FROM

logins a

JOIN

logins b ON a.user\_id = b.user\_id

AND DATE\_TRUNC('month', a.date) = DATE\_TRUNC('month', b.date) +

interval '1 month'

GROUP BY

date\_trunc('month', a.date)

Acknowledgement: Tom Moertel pointed out de-duping user-login pairs before the self-join would make the solution more efficient and contributed the alternate solution below. Thanks, TM!

Note: De-duping logins would also make the given solutions to Parts 2 and 3 of this problem more efficient as well.

Alternate solution:

WITH DistinctMonthlyUsers AS (

/\*

\* For each month, compute the \*set\* of users having logins.

\*/

SELECT DISTINCT

DATE\_TRUNC('MONTH', date) AS month\_timestamp,

user\_id

FROM logins

)

SELECT

CurrentMonth.month\_timestamp month\_timestamp,

COUNT(PriorMonth.user\_id) AS retained\_user\_count

FROM

DistinctMonthlyUsers AS CurrentMonth

LEFT JOIN

DistinctMonthlyUsers AS PriorMonth

ON

CurrentMonth.month\_timestamp = PriorMonth.month\_timestamp + INTERVAL '1 MONTH'

AND

CurrentMonth.user\_id = PriorMonth.user\_id

GROUP BY CurrentMonth.month\_timestamp

Part 2:

Task: Now we’ll take retention and turn it on its head: Write a query to find many users last month *did not* come back this month. i.e. the number of churned users.

*Solution:*

SELECT

DATE\_TRUNC('month', a.date) month\_timestamp,

COUNT(DISTINCT b.user\_id) churned\_users

FROM

logins a

FULL OUTER JOIN

logins b ON a.user\_id = b.user\_id

AND DATE\_TRUNC('month', a.date) = DATE\_TRUNC('month', b.date) +

interval '1 month'

WHERE

a.user\_id IS NULL

GROUP BY

DATE\_TRUNC('month', a.date)

Note that there are solutions to this problem that can use LEFT or RIGHT joins.

Part 3:

Context: You now want to see the number of active users this month *who have been reactivated* — in other words, users who have churned but this month they became active again. Keep in mind a user can reactivate after churning *before* the previous month. An example of this could be a user active in February (appears in logins), no activity in March and April, but then active again in May (appears in logins), so they count as a reactivated user for May .

Task: Create a table that contains the number of reactivated users per month.

*Solution:*

SELECT

DATE\_TRUNC('month', a.date) month\_timestamp,

COUNT(DISTINCT a.user\_id) reactivated\_users,

/\*

\* At least in the flavors of SQL I have used, you don't need to

\* include the columns used in HAVING in the SELECT statement.

\* I have written them out for clarity here.

\*/

MAX(DATE\_TRUNC('month', b.date)) most\_recent\_active\_previously

FROM

logins a

JOIN

logins b ON a.user\_id = b.user\_id

AND

DATE\_TRUNC('month', a.date) > DATE\_TRUNC('month', b.date)

GROUP BY

DATE\_TRUNC('month', a.date)

HAVING

month\_timestamp > most\_recent\_active\_previously + interval '1 month'

#4: Cumulative Sums

Acknowledgement: This problem was inspired by Sisense’s [“Cash Flow modeling in SQL”](https://www.sisense.com/blog/cash-flow-modeling-in-sql/) blog post

Context: Say we have a table transactions in the form:

| date | cash\_flow |

|------------|-----------|

| 2018-01-01 | -1000 |

| 2018-01-02 | -100 |

| 2018-01-03 | 50 |

| ... | ... |

Where cash\_flow is the revenues minus costs for each day.

Task: Write a query to get *cumulative* cash flow for each day such that we end up with a table in the form below:

| date | cumulative\_cf |

|------------|---------------|

| 2018-01-01 | -1000 |

| 2018-01-02 | -1100 |

| 2018-01-03 | -1050 |

| ... | ... |

*Solution:*

SELECT

a.date date,

SUM(b.cash\_flow) as cumulative\_cf

FROM

transactions a

JOIN b

transactions b ON a.date >= b.date

GROUP BY

a.date

ORDER BY

date ASC

Alternate solution using a window function (more efficient!):

SELECT

date,

SUM(cash\_flow) OVER (ORDER BY date ASC) as cumulative\_cf

FROM

transactions

ORDER BY

date ASC

#5: Rolling Averages

Acknowledgement: This problem is adapted from Sisense’s [“Rolling Averages in MySQL and SQL Server”](https://www.sisense.com/blog/rolling-average/) blog post

Note: there are different ways to compute rolling/moving averages. Here we'll use a preceding average which means that the metric for the 7th day of the month would be the average of the preceding 6 days and that day itself.

Context: Say we have table signups in the form:

| date | sign\_ups |

|------------|----------|

| 2018-01-01 | 10 |

| 2018-01-02 | 20 |

| 2018-01-03 | 50 |

| ... | ... |

| 2018-10-01 | 35 |

Task: Write a query to get 7-day rolling (preceding) average of daily sign ups.

*Solution:*

SELECT

a.date,

AVG(b.sign\_ups) average\_sign\_ups

FROM

signups a

JOIN

signups b ON a.date <= b.date + interval '6 days' AND a.date >= b.date

GROUP BY

a.date

Acknowledgement: Shay Halfon pointed out that using a window function would produce an identical and more efficient solution. Thanks, SH!

Alternate solution:

SELECT

date,

AVG(sign\_ups) OVER(ORDER BY date ROWS BETWEEN 6 PRECEDING AND 0 PRECEDING)

from

sign\_ups

#6: Multiple Join Conditions

Acknowledgement: This problem was inspired by Sisense’s [“Analyzing Your Email with SQL”](https://www.sisense.com/blog/analyzing-your-email-with-sql/) blog post

Context: Say we have a table emails that includes emails sent to and from zach@g.com:

| id | subject | from | to | timestamp |

|----|----------|--------------|--------------|---------------------|

| 1 | Yosemite | zach@g.com | thomas@g.com | 2018-01-02 12:45:03 |

| 2 | Big Sur | sarah@g.com | thomas@g.com | 2018-01-02 16:30:01 |

| 3 | Yosemite | thomas@g.com | zach@g.com | 2018-01-02 16:35:04 |

| 4 | Running | jill@g.com | zach@g.com | 2018-01-03 08:12:45 |

| 5 | Yosemite | zach@g.com | thomas@g.com | 2018-01-03 14:02:01 |

| 6 | Yosemite | thomas@g.com | zach@g.com | 2018-01-03 15:01:05 |

| .. | .. | .. | .. | .. |

Task: Write a query to get the response time per email (id) sent to zach@g.com . Do not include ids that did not receive a response from zach@g.com. Assume each email thread has a unique subject. Keep in mind a thread may have multiple responses back-and-forth between zach@g.com and another email address.

*Solution:*

SELECT

a.id,

MIN(b.timestamp) - a.timestamp as time\_to\_respond

FROM

emails a

JOIN

emails b

ON

b.subject = a.subject

AND

a.to = b.from

AND

a.from = b.to

AND

a.timestamp < b.timestamp

WHERE

a.to = 'zach@g.com'

GROUP BY

a.id

Window Function Practice Problems

#1: Get the ID with the highest value

Context: Say we have a table salaries with data on employee salary and department in the following format:

depname | empno | salary |

-----------+-------+--------+

develop | 11 | 5200 |

develop | 7 | 4200 |

develop | 9 | 4500 |

develop | 8 | 6000 |

develop | 10 | 5200 |

personnel | 5 | 3500 |

personnel | 2 | 3900 |

sales | 3 | 4800 |

sales | 1 | 5000 |

sales | 4 | 4800 |

Task: Write a query to get the empno with the highest salary. Make sure your solution can handle ties!

*Solution:*

WITH max\_salary AS (

SELECT

MAX(salary) max\_salary

FROM

salaries

)

SELECT

s.empno

FROM

salaries s

JOIN

max\_salary ms ON s.salary = ms.max\_salary

Alternate solution using RANK():

WITH sal\_rank AS

(SELECT

empno,

RANK() OVER(ORDER BY salary DESC) rnk

FROM

salaries)

SELECT

empno

FROM

sal\_rank

WHERE

rnk = 1;

#2: Average and rank with a window function (multi-part)

Part 1:

Context: Say we have a table salaries in the format:

depname | empno | salary |

-----------+-------+--------+

develop | 11 | 5200 |

develop | 7 | 4200 |

develop | 9 | 4500 |

develop | 8 | 6000 |

develop | 10 | 5200 |

personnel | 5 | 3500 |

personnel | 2 | 3900 |

sales | 3 | 4800 |

sales | 1 | 5000 |

sales | 4 | 4800 |

Task: Write a query that returns the same table, but with a new column that has average salary per depname. We would expect a table in the form:

depname | empno | salary | avg\_salary |

-----------+-------+--------+------------+

develop | 11 | 5200 | 5020 |

develop | 7 | 4200 | 5020 |

develop | 9 | 4500 | 5020 |

develop | 8 | 6000 | 5020 |

develop | 10 | 5200 | 5020 |

personnel | 5 | 3500 | 3700 |

personnel | 2 | 3900 | 3700 |

sales | 3 | 4800 | 4867 |

sales | 1 | 5000 | 4867 |

sales | 4 | 4800 | 4867 |

*Solution:*

SELECT

\*,

/\*

\* AVG() is a Postgres command, but other SQL flavors like BigQuery use

\* AVERAGE()

\*/

ROUND(AVG(salary),0) OVER (PARTITION BY depname) avg\_salary

FROM

salaries

Part 2:

Task: Write a query that adds a column with the rank of each employee based on their salary within their department, where the employee with the highest salary gets the rank of 1. We would expect a table in the form:

depname | empno | salary | salary\_rank |

-----------+-------+--------+-------------+

develop | 11 | 5200 | 2 |

develop | 7 | 4200 | 5 |

develop | 9 | 4500 | 4 |

develop | 8 | 6000 | 1 |

develop | 10 | 5200 | 2 |

personnel | 5 | 3500 | 2 |

personnel | 2 | 3900 | 1 |

sales | 3 | 4800 | 2 |

sales | 1 | 5000 | 1 |

sales | 4 | 4800 | 2 |

*Solution:*

SELECT

\*,

RANK() OVER(PARTITION BY depname ORDER BY salary DESC) salary\_rank

FROM

salaries

Other Medium/Hard SQL Practice Problems

#1: Histograms

Context: Say we have a table sessions where each row is a video streaming session with length in seconds:

| session\_id | length\_seconds |

|------------|----------------|

| 1 | 23 |

| 2 | 453 |

| 3 | 27 |

| .. | .. |

Task: Write a query to count the number of sessions that fall into bands of size 5, i.e. for the above snippet, produce something akin to:

| bucket | count |

|---------|-------|

| 20-25 | 2 |

| 450-455 | 1 |

Get complete credit for the proper string labels (“5-10”, etc.) but near complete credit for something that is communicable as the bin.

*Solution:*

WITH bin\_label AS

(SELECT

session\_id,

FLOOR(length\_seconds/5) as bin\_label

FROM

sessions

)

SELECT

CONCATENTATE(STR(bin\_label\*5), '-', STR(bin\_label\*5+5)) bucket,

COUNT(DISTINCT session\_id) count

GROUP BY

bin\_label

ORDER BY

bin\_label ASC

#2: CROSS JOIN (multi-part)

Part 1:

Context: Say we have a table state\_streams where each row is a state and the total number of hours of streaming from a video hosting service:

| state | total\_streams |

|-------|---------------|

| NC | 34569 |

| SC | 33999 |

| CA | 98324 |

| MA | 19345 |

| .. | .. |

(In reality these kinds of aggregate tables would normally have a date column, but we’ll exclude that component in this problem)

Task: Write a query to get the pairs of states with total streaming amounts within 1000 of each other. For the snippet above, we would want to see something like:

| state\_a | state\_b |

|---------|---------|

| NC | SC |

| SC | NC |

*Solution:*

SELECT

a.state as state\_a,

b.state as state\_b

FROM

state\_streams a

CROSS JOIN

state\_streams b

WHERE

ABS(a.total\_streams - b.total\_streams) < 1000

AND

a.state <> b.state

FYI, CROSS JOIN s can also be written without explicitly specifying a join:

SELECT

a.state as state\_a,

b.state as state\_b

FROM

state\_streams a, state\_streams b

WHERE

ABS(a.total\_streams - b.total\_streams) < 1000

AND

a.state <> b.state

Part 2:

Note: This question is considered more of a bonus problem than an actual SQL pattern. Feel free to skip it!

Task: How could you modify the SQL from the solution to Part 1 of this question so that duplicates are removed? For example, if we used the sample table from Part 1, the pair NC and SC should only appear in one row instead of two.

*Solution:*

SELECT

a.state as state\_a,

b.state as state\_b

FROM

state\_streams a, state\_streams b

WHERE

ABS(a.total\_streams - b.total\_streams) < 1000

AND

a.state > b.state

#3: Advancing Counting

Acknowledgement: This question is adapted from [this Stack Overflow question](https://stackoverflow.com/questions/54488894/using-case-to-properly-count-items-with-if-else-logic-in-sql) by me (zthomas.nc)

Note: this question is probably more complex than the kind you would encounter in an interview. Consider it a challenge problem, or feel free to skip it!

Context: Say I have a table table in the following form, where a user can be mapped to multiple values of class:

| user | class |

|------|-------|

| 1 | a |

| 1 | b |

| 1 | b |

| 2 | b |

| 3 | a |

Task: Assume there are only two possible values for class. Write a query to count the number of users in each class such that any user who has label a and b gets sorted into b, any user with just a gets sorted into a and any user with just b gets into b.

For table that would result in the following table:

| class | count |

|-------|-------|

| a | 1 |

| b | 2 |

*Solution:*

WITH usr\_b\_sum AS

(

SELECT

user,

SUM(CASE WHEN class = 'b' THEN 1 ELSE 0 END) num\_b

FROM

table

GROUP BY

user

),

usr\_class\_label AS

(

SELECT

user,

CASE WHEN num\_b > 0 THEN 'b' ELSE 'a' END class

FROM

usr\_b\_sum

)

SELECT

class,

COUNT(DISTINCT user) count

FROM

usr\_class\_label

GROUP BY

class

ORDER BY

class ASC

Alternate solution: Using SELECTs in the SELECT statement and UNION:

SELECT

"a" class,

COUNT(DISTINCT user\_id) -

(SELECT COUNT(DISTINCT user\_id) FROM table WHERE class = 'b') count

UNION

SELECT

"b" class,

(SELECT COUNT(DISTINCT user\_id) FROM table WHERE class = 'b') count

Alternate solution: Since the problem as stated didn’t ask for generalizable solution, you can leverage the fact that b > a to produce this straightforward solution:

Acknowledgement: Thanks to Karan Gadiya for contributing this solution. Thanks, KG!

WITH max\_class AS (

SELECT

user,

MAX(class) as class

FROM

table

GROUP BY

user

)

SELECT

class,

COUNT(user)

FROM

max\_class

GROUP BY

class

# 

# Ending note

Thank you for reading this. Hope this was helpful.