SQL Tips:

**Checklist in the Tech Interveiw**

1. Understand the tables. Speak about your understanding with you intreviewer to confirm
2. Write down the table you need
3. Start with small
4. Specify table in select
5. Alias
6. Rename column
7. Order

**Database query**

1. Update set
2. Delete from
3. CREATE TABLE customers(id int, FirstName varchar(255));
4. INSERT INTO customers(FirstName, LastName, address, email)

VALUES ('Jason', 'Dsouza', 'McLaren Vale, South Australia', '[test@fakeGmail.com](mailto:test@fakeGmail.com)');

1. Alter table my table
2. Add default
3. Drop

**NULL:**

* SELECT NULL/0
* SELECT NULL+1
* SELECT NULL+'1'

**Error:**

* SELECT AVG (NULL)
* SELECT MAX (NULL)
* SELECT SUM (NULL)
* SELECT 0/0

**Tricky questions**

* SELECT Max ('TD') -TD
* SELECT MAX (1,3,8) - Error
* SELECT Max ('TD'+'AD') - TDAD
* select \* from 'Employee' - Syntax error
* 2 <> NULL -Null
* 3 NOT IN (1, 2, NULL) - False

**Data types/Special select**

CAST(ranks as signed)

COALESCE(NULL, NULL, NULL, 'W3Schools.com');

NVL()

**DATE Function**

* DATE\_ADD('1998-01-02', INTERVAL 31 DAY)
* DATE\_SUB("2017-06-15", INTERVAL 10 DAY)
* EXTRACT(YEAR FROM '1999-07-02')
* DATE\_TRUNC('month' , cleaned\_date)
* DATENAME(weekday, '2017/08/25')
* extract(year\_month from program\_date)="202006")
* DATEPART(year, '2017/08/25')
* DATEDIFF(year, '2017/08/25', '2011/08/25')
* CURRENT\_DATE()

**Numeric**

* CEILING: Upward
* FLOOR: Downward
* TRUNCATE(345.156, -2)
* ISNUMERIC(4567)

**String**

* POSITION('A' IN descript)
* SUBSTR(date, 4, 2)
* group\_concat

**Windows**

* NTILE(4) OVER (PARTITION BY start\_terminal ORDER BY duration\_seconds)
* LAG(duration\_seconds, 1) OVER(PARTITION BY start\_terminal ORDER BY duration\_seconds)

**Get index 1-10**

with recursive cte as

(select 1 as ids

union all

select ids+1 from cte

where ids<10)

select ids

from cte

**Moving average**

select \*,

avg(Price) OVER(ORDER BY Date

ROWS BETWEEN 2 PRECEDING AND CURRENT ROW )

as moving\_average

from stock\_price;

**User Cohort Analysis**

1. Find the first\_login of each user
2. Left join by user\_id and datediff=1
3. Group by first\_day and count user\_id

**User continuously use**

1. Get Row\_number order by date
2. Group by date-ranks and select max, min date
3. Results: user\_id, startdate, enddate (continusly visit)

**Continous seats**

select distinct c.seat\_id

from cinema c

join cinema c1

on abs(c1.seat\_id-c.seat\_id)=1

and c.free=1 and c1.free=1

order by 1

**Common Friends:**

WITH temp AS (

SELECT

user1\_id AS user\_id,

user2\_id AS friend\_id

FROM

Friendship

UNION

SELECT

user2\_id AS user\_id,

user1\_id AS friend\_id

FROM

Friendship)

SELECT

a.user1\_id,

a.user2\_id,

COUNT(\*) AS common\_friend

FROM

Friendship a

JOIN temp f1

ON a.user1\_id = f1.user\_id AND a.user2\_id != f1.friend\_id

JOIN temp f2

ON a.user2\_id = f2.user\_id AND a.user1\_id != f2.friend\_id

WHERE

a.user1\_id < a.user2\_id

AND f1.friend\_id = f2.friend\_id

GROUP BY 1,2

HAVING COUNT(\*) >= 3;