SQL Server Interview Questions & Answers

www.questpond.com

Contents

Question 1 :- Explain normalization ?

→ Normalization is a database design technique to remove redundant data.

Question 2 :- How to implement normalization ?

→ Normalization is implemented by splitting tables in to two, one with reference data (master table) and other transaction data.

Question 3 :- What is denormalization ?

→ Denormalization is a database design technique to improve search performance. In denormalization we merge table so that we need to fetch from less tables and thus increase search performance.

Question 4 :- Explain OLTP vs OLAP ?

→ OLTP Normalization avoids redundancy and we follow normalization design(1st,2nd and 3rd normal).

OLAP Denormalization improve search performance and we follow denormalization design.

Question 5 :- Explain 1st,2nd and 3rd Normal form ?

→ **1st normal form :-** A table is in first normal form when the columns have Atomic values. It should not have repeating groups.

2nd Normal form :- First normal form should be satisfied. All non-key columns should be fully dependent on the Primary key.

3rd Normal :- All 1st and 2nd normal form should be satisfied. No transient dependency should be present.

Question 6 :- Primary Key vs Unique key ?

→ Remember the 2 Ns.

1st N NULLS :- Unique can have NULLS , but primary key can not have NULLS.

2nd N Numbers :- Many unique keys but ony ONE Primary key.

Question 7 :- Differentiate between Char vs Varchar ?

→ Char is Fixed length while varchar is variable length.

Question 8 :- Differentiate between Char vs NChar ?

→ If you want to just store english characters then use Char , For multilingual language (Non-English) use NChar.

Question 9 :- Whats the size of Char vs NChar ?

→ Char 1 Character = 1 byte , For NChar 1 Character = 2 bytes.

Question 10 :- What is the use of Index ?

→ Indexes increases search performance.

Question 11 :- How does it make search faster?

→ Search becomes faster because of Balance tree structure. Internally it creates Node and Leaf nodes to reach to the data quick.

Question 13 :- Clustered vs Non-Clustered index

→ In Clustered index leaf node will point to actual data. While in case of non-clustered index leaf node takes help of clustered index

Question 14 :- Function vs Stored Procedures

	Function	**Stored procedure**
Goal	Computed values But will not make any permanent changes to the environment. Only Selects allowed, insert/update/deletes not allowed.	Mini Batch program. Can change the environment. Insert, Updates and Deletes allowed.
Execution	Can be called from select/where/call from other Stored procedure.	Stored procedures can not be executed from Select/Where or from other functions.
Output	Mostly Scalar value, Table valued functions.	Can have single or multiple outputs.

Function returns Computed Scalar values. You cannot make permanent changes (insert, update, delete) inside function.

Stored procedure is a mini-programs which can do anything , make changes , backup database.

Question 15 :- What are triggers and why do you need it ?

→ Triggers are logics which can be executed when events like insert, update, delete etc happens.

Question 16 :- What are types of triggers ?

→ There are two types After trigger and Instead OF trigger.

Question 17 :- Differentiate between After trigger vs Instead Of ?

→ After trigger :- After event has happened logic is executed.

Instead Of trigger:- Instead of the event the logic is executed.

Question 18 :- What is need of Identity ?

→ Identity helps to define auto-incremented column.

Question 19 :- Explain transactions and how to implement it ?

→ Transaction treats series of activity as one single unit. Either everything is successful and or everything rollbacks.

Question 20 :- What are inner joins ?

→ Inner join selects matching records from both tables.

Question 21 :- Explain Left join ?

→ All data from left table selected and only matching records from right table.

Question 22 :- Explain Right join ?

→ All data from right table selected and only matching records from left table.

Question 23 :- Explain Full outer joins ?

→ All matching and unmatching records from both left and right table are selected.

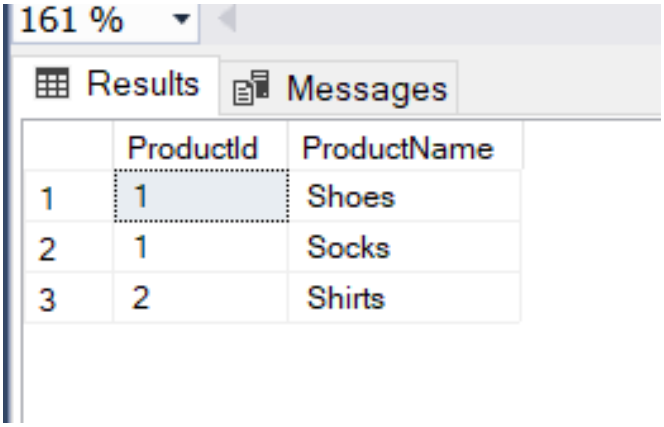
Question 24 :- Explain Cross joins ?

→ Cross join is cartesian. Every record of one table is joined with other table records.

Question 25:-Why do we need UNION ?

→ Union combines two result sets.

```
select [ProductId], [ProductName] from [LearnSql].[dbo].[mst_Products]
union
select [ProductId], [ProductName] from [LearnSql].[dbo].[mst_ExpiredProducts]
```

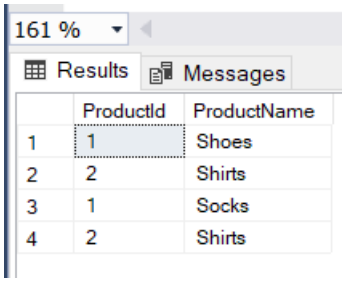


	ProductId	ProductName
1	1	Shoes
2	1	Socks
3	2	Shirts

Question 26:-Differentiate between Union vs Union All ?

→ Union combines result sets and excludes duplicates while Union all also combines result set but includes duplicates.

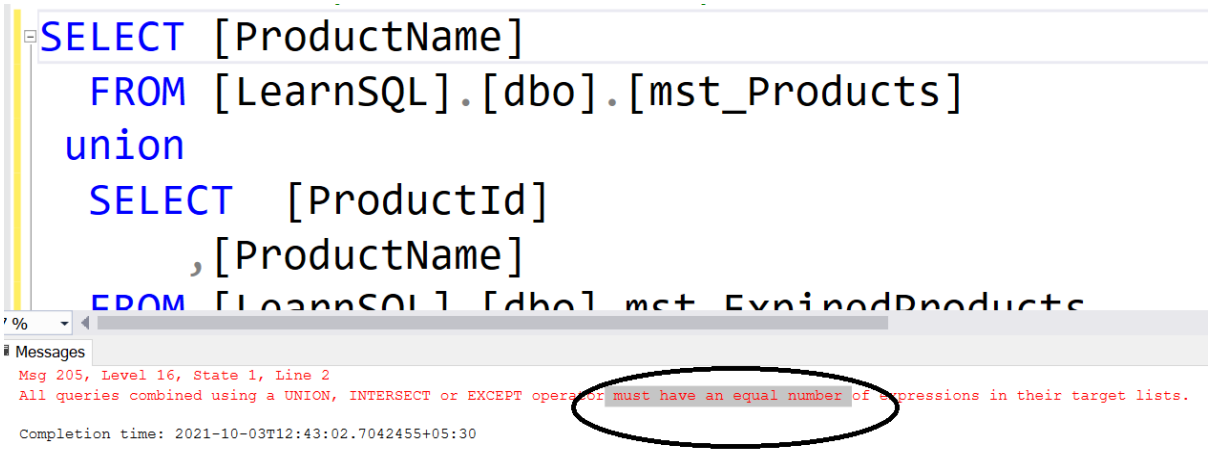
```
select [ProductId],
[ProductName]
        from [LearnSql].[dbo].[mst_Products]
        union all
select [ProductId],
        [ProductName]
        from [LearnSql].[dbo].[mst_ExpiredProducts]
```



Question 27:-can we have unequal columns in Union?

→ No.

```
select
[ProductName]
        from [LearnSql].[dbo].[mst_Products]
        union all
select [ProductId],
        [ProductName]
        from [LearnSql].[dbo].[mst_ExpiredProducts]
```



Question 28:-Can column have different data types in Union ?

→ No.

```
select
        [ProductName],
[ProductId]
```

from [LearnSql].[dbo].[mst_Products]

union all

select

[ProductId],

[ProductName]

from [LearnSql].[dbo].[mst_ExpiredProducts]

```
select
    [ProductName],[ProductId]
    from [LearnSql].[dbo].[mst_Products]
    union all
select [ProductId],
    [ProductName]
    from [LearnSql].[dbo].[mst_ExpiredProducts]
```

```
esults  Messages
Msg 245, Level 16, State 1, Line 1
Conversion failed when converting the varchar value 'Shoes' to data type int.
```

Question 29:- Which Aggregate function have you used ?

→

Sum,Avg,Max, Min and Count.

select sum([CustomerAmount]),

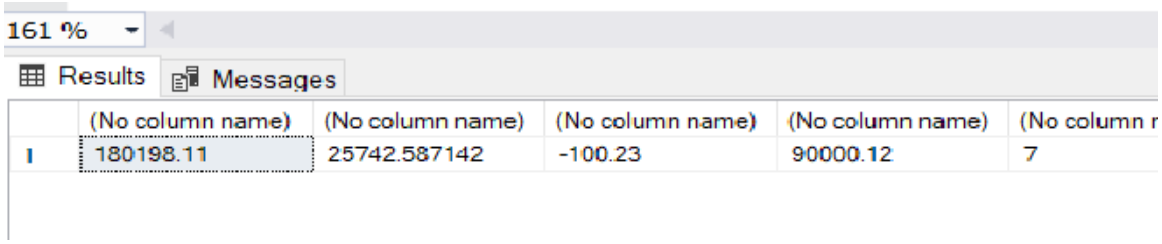
Avg([CustomerAmount]) ,

min([CustomerAmount]),

max([CustomerAmount]),

Count(*)

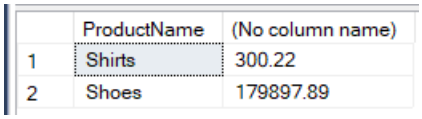
from [dbo].[txn_Customer]



Question 30:- When to use Group by ?

→ It helps to convert rows in to summary rows using common values.

```
select [ProductName],sum([CustomerAmount])
        from [dbo].[txn_Customer]
        group by [ProductName]
```



	ProductName	(No column name)
1	Shirts	300.22
2	Shoes	179897.89

Question 31:- Can we select column which is not part of group by ?

→No. In a group by you can only select columns which are present in groupby.

select[CustomerName],[ProductName],sum([CustomerAmount]) from [dbo].[txn_Customer]

group by [ProductName]

```
select[CustomerName],[ProductName],
    sum([CustomerAmount]) from [dbo].[txn_Customer]
group by [ProductName]
```

```
Msg 8120, Level 16, State 1, Line 2
Column 'dbo.txn_Customer.CustomerName' is invalid in the select list because it is not
 contained in either an aggregate function or the GROUP BY clause.
```

Question 32:- What is having clause ?

→ Having clause helps to filter group by data.

select [ProductName],sum([CustomerAmount]) from [dbo].[txn_Customer]

group by [ProductName]

having [ProductName]='shoes'

	ProductName	(No column name)
1	Shoes	179897.89

Question 33:- Having clause vs Where clause

	Having	**Where**
Sequence	Filter is applied After Group by	Filter is applied Before Group by
Aggregate	Having can have Aggregate	Where cannot have Aggregate
Filter level	Aggregate group level	Row level

Question 34:- How can we sort records ?

Sorting is done by using order by clause.

select * from [dbo].[txn_Customer]

order by [CustomerAmount] desc

Results Messages

	Id	CustomerName	CustomerAmount	ProductName
1	3	Guru	90000.12	Shoes
2	5	Rohit	89898.00	Shoes
3	2	Raju	100.22	Shirts
4	4	Simran	100.00	Shirts
5	6	Ganesh	100.00	Shirts
6	8	Aditya	100.00	Shoes
7	1	Shiv	-100.23	Shoes

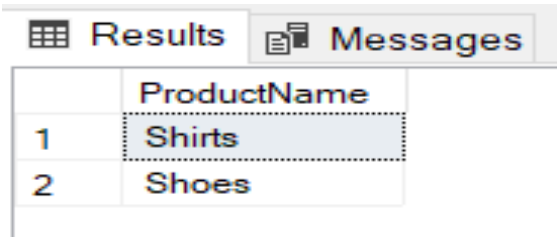
Question 35:- What's the default sort ?

→ Ascending.

Question 36:- How can we remove duplicates ?

→ By using Distinct keyword.

select distinct [ProductName] from [dbo].[txn_Customer]



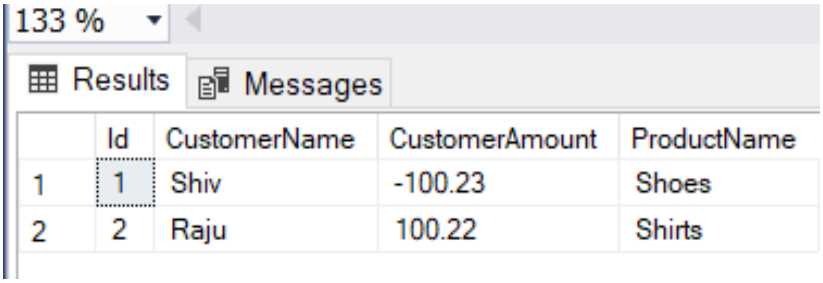
Results Messages

	ProductName
1	Shirts
2	Shoes

Question 37:- Select the first top X records ?

→ By using the top keyword.

select top 2 * from [dbo].[txn_Customer]



133 %

Results Messages

	Id	CustomerName	CustomerAmount	ProductName
1	1	Shiv	-100.23	Shoes
2	2	Raju	100.22	Shirts

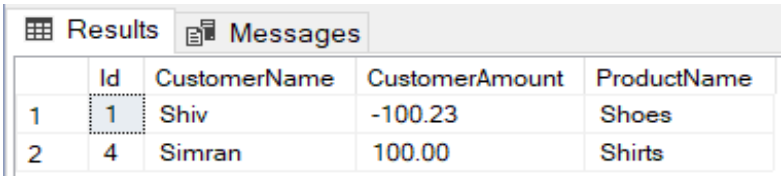
Question 38:- How to handle NULLS ?

→ By using ISNULL function.

Question 39:- What is use of wild cards ?

→ Wild card helps in pattern matching.

select * from [dbo].[txn_Customer] where [CustomerName] like 's%';



Results Messages

	Id	CustomerName	CustomerAmount	ProductName
1	1	Shiv	-100.23	Shoes
2	4	Simran	100.00	Shirts

Question 40:- What is the use of Alias ?

→ Alias helps to give different display names to original column names.

select [CustomerName] as name ,[CustomerAmount] as Amount,

[ProductName] as Product from [dbo].[txn_Customer]

Results Messages

	name	Amount	Product
1	Shiv	-100.23	Shoes
2	Raju	100.22	Shirts
3	Guru	90000.12	Shoes
4	Simran	100.00	Shirts
5	Rohit	89898.00	Shoes
6	Ganesh	100.00	Shirts
7	Aditya	100.00	Shoes

Question 41:- How to write a case statement ?

```
select [CustomerName],
case
when [CustomerAmount]<200 then 'less than 200'
when [CustomerAmount]>200 then 'more than 200'
else 'NA'
END as CustomerAmount,[CustomerAmount] from [dbo].[txn_Customer]
```

161 %

Results Messages

	CustomerName	CustomerAmount	CustomerAmount
1	Shiv	less then 200	-100.23
2	Raju	less then 200	100.22
3	Guru	more then 200	90000.12
4	Simran	less then 200	100.00
5	Rohit	more then 200	89898.00
6	Ganesh	less then 200	100.00
7	Aditya	less then 200	100.00

Question 42:- What is self reference tables ?

→ Self reference tables are those tables who have primary key and foreign key in the same table.

Question 43:- What is self join ?

→ When you make joins (inner,left,right) with same table it's called as Self join.

```
select t1.Id,t1.Referenceid_fk,
t2.CustomerName as name,t1.CustomerName as Reference
from txn_Customer as t1
inner join
txn_Customer t2 on t1.id=t2.Referenceid_fk
```

214 %

Results | Messages

	Id	Referenceid_fk	name	Reference
1	1	NULL	Raju	Shiv
2	1	NULL	Guru	Shiv
3	2	1	Simran	Raju
4	1	NULL	Rohit	Shiv
5	4	2	Ganesh	Simran

Question 44:- Explain the between clause ?

→ Between clause helps to find values in between the range.

```
select * from txn_Customer
where CustomerAmount between -200 and 200
```

214 %

Results | Messages

	Id	CustomerName	CustomerAmount	ProductName	Referenceid_fk
1	1	Shiv	-100.23	Shoes	NULL
2	2	Raju	100.22	Shirts	1
3	4	Simran	100.00	Shirts	2
4	6	Ganesh	100.00	Shirts	4
5	8	Aditya	100.00	Shoes	NULL

Question 45:-Explain SubQuery ?

→ Subquery is a query inside a query (nested query).In Subquery first the inner query gets evaluated and then outer query.

Question 46:-Can inner Subquery return multiple results ?

→ Yes , Inner query can return multiple results but then in where clause you will need to us the "IN" keyword.

Question 47:-What is Co-related Query ?

→ In co-related query first the outer query sends records to the inner query , inner query then evaluates and sends its back to the outer query.

Question 48:-Differentiate between Joins and SubQuery ?

	Subquery	**Join**
Intention	Series of processing where one processing sends output to other.	Join two tables and get matching or not matching records.
Select fields	Can not select from inner Query	Can select multiple fields from table.

Question 49: -Performance Joins vs Subquery?

→ Most of the times Joins should perform better. But not necessarily, its possible subquery can be faster many times. So SQL plan needs to be looked in to determine whose performance can be better.

Question 50:- Select the top nth highest salary using top and order by?

→ Example of a Query to find the second highest salary i.e. n=**2** :

Select Top 1 EmployeeSalary from (select distinct Top **2** EmployeeSalary from tblEmployee order by EmployeeSalary desc) as innerquery order by EmployeeSalary asc

Question 51:- Select the top nth highest salary using correlated Queries?

→ Example of a Query to find the 2nd highest salary i.e. n=**2** :

Select E1.EmployeeSalary from tblEmployee as E1 where **2**=(Select count(*) from tblEmployee as E2 where E2.EmployeeSalary>=E1.EmployeeSalary)

Question 52:- Select top nth using TSQL

→ Example of a Query to find the 2nd highest salary i.e. n= 2-1= **1**

Select Distinct(EmployeeSalary) from tblEmployee order by EmployeeSalary DESC offset **1** rows Fetch next 1 rows only

Note: Offset **1** rows means skip first row. So if we have to select the 2nd highest then offset skip 1st row and then fetch 1st row from the remaining rows which is the second highest.

Question 53:- Performance comparison of all the methods.

	Order by/Top	**Co-related**	**OFFSET fetch**
Performance	Good	Slow	Good
Parametric	No	Yes	Yes
Cross DB	Yes	Yes	No

Happy Learning, Happy Job Hunting!