SQL-35 QUESTIONS

Sample tables are listed to visualize the data and associate with query answers given.

1. ==================
2. Consider below tables
3. ==================
4. EMPLOYEE
5. empid   empname managerid   deptid  salary  DOB
6. 1       emp 1       0       1       6000    1982-08-06 00:00:00.000
7. 2       emp 2       0       5       6000    1982-07-11 00:00:00.000
8. 3       emp 3       1       1       2000    1983-11-21 00:00:00.000
9. 13      emp 13      2       5       2000    1984-03-09 00:00:00.000
10. 11      emp 11      2       1       2000    1989-07-23 00:00:00.000
11. 9       emp 9       1       5       3000    1990-09-11 00:00:00.000
12. 8       emp 8       3       1       3500    1990-05-15 00:00:00.000
13. 7       emp 7       2       5       NULL    NULL
14. 3       emp 3       1       1       2000    1983-11-21 00:00:00.000
16. --DEPARTMENT TABLE
17. deptid  deptname
18. 1       IT
19. 2       Admin

**Q1 Employee and Manager ID are in the same table; can you get manager names for employees?  
  
Answer:**  
With the help of Common table expressions, we can achieve this.

1. ;**with** empCTE **as**
2. (
3. **select** e.empid, e.empname, e.managerid,
4. CAST('' **as** **varchar**(50)) **as** Mname **from** employee e
5. **where** managerid = 0
7. **union** all
9. **select** e1.empid, e1.empname, e1.managerid,
10. CAST(c.empname **as** **varchar**(50)) **as** Mname **from** employee e1
11. **inner** join empCTE **as** C **on** e1.managerid=c.empid
12. **where** e1.managerid>0
13. ) **select** \* **from** empCTE

**Q2 Can you get employee details whose department id is not valid or department id not present in department table?  
  
Answer**Identifying Department IDs in employee table, which are not available in master.  
  
There are multiple ways to do this.

Using Left JOIN

1. **SELECT** E.EMPID,E.EMPNAME, E.DEPTID **FROM** EMPLOYEE E
2. left outer join DEPARTMENT d
3. **on** E.DEPTID = D.DEPTID
4. **WHERE** D.DEPTID **IS** NULL

Using NOT IN

1. **SELECT** E.EMPID,E.EMPNAME, E.DEPTID **FROM** EMPLOYEE E
2. **where** e.deptid not in (**select** deptid **from** department)

Using NOT Exists

1. **SELECT** E.EMPID,E.EMPNAME, E.DEPTID **FROM** EMPLOYEE E
2. **where** NOT EXISTS (**select** deptid **from** department **where** e.deptid=department.deptid)

**Note**  
"Not In"  is the least recommended, considering performance. Outer join and Not Exists are preferred.  
  
Using EXCEPT KEYWORD  
  
if you want to list Department IDs only. INTERSECT and EXCEPT keywords have rules

1. **SELECT** deptid **FROM** EMPLOYEE
2. **EXCEPT**
3. **SELECT** DEPTID **FROM** DEPARTMENT

**Q3. Can you get the list of employees with same salary?**

**Answer:**

With where clause

1. **Select** **distinct** e.empid,e.empname,e.salary
2. **from** employee e, employee e1
3. **where** e.salary =e1.salary
4. and e.empid != e1.empid

**Q4 How can you find duplicate records in Employee table?**

**Answer:**

1. **SELECT** EMPID,EMPNAME, SALARY, COUNT(\*) **AS** CNT
2. **FROM** EMPLOYEE
3. **GROUP** **BY** EMPID,EMPNAME, SALARY
4. **HAVING** COUNT(\*)>1

**Q5 How can you  DELETE DUPLICATE RECORDS?**

**Answer**

There are multiple options to perform this operation.   
  
Using row count to restrict delete only 1 record

1. **set** rowcount 1
2. **DELETE** **FROM** EMPLOYEE **WHERE** EMPID IN (
3. **SELECT** EMPID
4. **FROM** EMPLOYEE
5. **GROUP** **BY** EMPID,EMPNAME, SALARY
6. **HAVING** COUNT(\*)>1
7. )
8. **set** rowcount 0

Use auto increment primary key "add" if not available in the table, as in given example.

1. **alter** **table** employee
2. **add** empidpk **int** identity (1,1)

Now, perform query on min of auto pk id, group by duplicate check columns - this will give you latest duplicate records

1. **select** \* **from** employee **where**
2. empidpk not in ( **select** **min**(empidpk) **from** employee
3. **group** **by** EMPID,EMPNAME, SALARY )

Now, delete.

1. **Delete** **from** employee **where**
2. empidpk not in ( **select** **min**(empidpk) **from** employee
3. **group** **by** EMPID,EMPNAME, SALARY )

**Q6 Find the second highest salary.**

**Answer**

1. **Select** **max**(Salary) **from** employee
2. **where** Salary not in (**Select** **max**(Salary) **from** employee)

**Q7 Now, can you find 3rd, 5th or 6th i.e. N'th highest Salary?**

**Answer**

Query for 3rd highest salary

1. **SELECT** \* **FROM** EMPLOYEE E
2. **WHERE** 2 = (**SELECT** COUNT(**DISTINCT** E1.SALARY)
3. **FROM** EMPLOYEE E1
4. **WHERE** E1.SALARY>E.SALARY)

Here, 2= 3-1 i.e. N-1 ; can be applied for any number.

**Q8. Can you write a query to find employees with age greater than 30?**

**Answer**

1. **select** \* **from** employee
2. **where** datediff(year,dob, getdate()) >30

**Q9 Write an SQL Query to print the name of the distinct employees whose DOB is between 01/01/1960 to 31/12/1987**

**Answer**

1. **SELECT** **DISTINCT** EmpName **FROM** Employee
2. **WHERE** DOB BETWEEN '01/01/1960' AND '12/31/1987'

**Q10  Please write a query to get the maximum salary from each department.**

**Answer**

1. **select** DeptId, **max**(salary) **as** Salary **from** employee **group** **by** deptid

**Q11 What is wrong with the following query?**

1. **SELECT** empName **FROM** employee **WHERE** salary <> 6000

**Answer**

The following query will not fetch record with the salary of 6000 but also will skip the record with NULL.  
  
As per SQL Server logic, it works on 3 values in matching conditions. TRUE or FALSE and UNKNOWN. Here,  NULL implies UNKNOWN.

to fix this:

1. **SELECT** empName  **FROM**
2. employee **WHERE** salary **is** NULL or salary <> 6000

**Q12. Can you show one row twice in results from a table?**

**Answer**Yes. We can use union all or cross join to obtain this.

1. **select** deptname **from** department d **where** d.deptname='it'
2. **union** all
3. **select** deptname **from** department d1 **where** d1.deptname='it'

 -- also cross join alias same table

1. **select** d.deptname **from** department d, department d1
2. **where** d.deptname='it'

**Q13 Could you tell the output or result of the following SQL statements?**

**Answer**

1. **select** '7'
2. -- output = 7
3. **select** 7
4. -- output = 7
5. **select** count (7)
6. -- output = 1
7. **SELECT** COUNT('7')
8. -- output = 1
9. **SELECT** COUNT(\*)
10. -- output = 1

**Q14 What is an alternative for TOP clause in SQL?**

**Answer**

- There can be two alternatives for the top clause in SQL.  
  
#1

-- Alternative - ROWCOUNT function

1. **Set** rowcount 3
2. **Select** \* **from** employee **order** **by** empid **desc**
3. **Set** rowcount 0

#2

-- Alternative and  WITH and ROWNUMBER function

-- between 1 and 2

1. **With** EMPC **AS**
2. ( **SELECT** empid, empname,salary,
3. ROW\_NUMBER() OVER (**order** **by** empid **desc**) **as** RowNumber
4. **FROM** employee )
5. **select** \*
6. **from** EMPC
7. **Where** RowNumber Between 1 and 7

**Q15** **Will the following statements  run or give error?**

**Answer**NO error.

1. **SELECT** COUNT(\*) + COUNT(\*)
2. **Output**  = 2
3. **SELECT** (**SELECT** 'c#')
4. **Output** = c#

**Q16 Can you write a query to get employee names starting with a vowel?**

**Answer**

Using like operator and expression,

1. **Select** empid, empname **from** employee **where** empname like '[aeiou]%'

**Q17 Can you write a query to get employee names ending with a vowel?**

**Answer**

1. **Select** empid, empname **from** employee **where** empname like '%[aeiou]'

**Q18 Can you write a query to get employee names starting and ending with a vowel?**

**Answer**

Here you will get only one record of "empone".

1. **select** empid, empname **from** employee **where** empname like '[aeiou]%[aeiou]'

**Q19  Write a query to get employees whos ID is even.**

**Answer**

1. **select** \* **from** employee
2. **where** empid %2 =0

**Q20 Write a query to get employees whos ID is an odd number.**

**Answer**

1. **select** \* **from** employee
2. **where** empid %2 !=0

**Q21 How can you get random employee record from the table?**

**Answer**

1. **select** **top** 1 \* **from** employee **order** **by** newid()

**Q22(Tricky) Below is the table data which has 1 columns and 7 rows**

1. **Table** -TESTONE
2. DATACOL
3. 10/12
4. 1a/09
5. 20/14
6. 20/1c
7. 3112
8. 11/16
9. mm/pp

Give data in a table is of format 'NN/NN', verify that the first and last two characters are numbers and that the middle character is '/'.

**Answer**

Print the expression 'NUMBER' if valid, 'NOT NUM' if not valid.  
  
This can be done using like operator and expression. Checking numbers and not characters.

1. **SELECT** DataCol, 'CHECK' =
2. CASE
3. **WHEN** datacol like '%[0-9]%[^A-Z]%/%[^A-Z]%[0-9]%' **then** 'NUMBER'
4. **else** 'NOT NUM'
5. **end**
6. **from** TestOne

**Q23 Consider following 3 tables with one column**

1. Tbl1
2. col1
3. 1
4. 1
5. 1
6. Tbl2
7. col1
8. 2
9. 2
10. 2
11. Tbl3
12. col1
13. 3
14. 3
15. 3

**How many rows will following query return? (0, 3 or 9)**

1. **Select** \* **from** Tbl1 **inner** join tbl2 **on** tbl1.col1=tbl2.col1
2. Left outer join Tbl3 **on** Tbl3.Col1=Tbl2.Col1

**Answer- 0 .**

**Q24 If all values from tbl2 are deleted. What will be the output of the following query?**

**Answer**

select Tbl1.\* from tbl1,tbl2

Ans - 0 Rows.

**Q25 Can you write a query to print prime numbers from 1 to 100?**

**Answer**  
For this, we have to use a loop as in other programming languages.

1. **DECLARE**
3. @i **INT**,
4. @a **INT**,
5. @count **INT**,
6. @result **varchar**(**Max**)
8. **SET** @i = 1
9. **set** @result=''
11. WHILE (@i <= 100)
12. **BEGIN**
13. **SET** @count = 0
14. **SET** @a = 1
15. -- logic to check prime number
16. WHILE (@a <= @i)
17. **BEGIN**
18. IF (@i % @a = 0)
19. **SET** @count = @count + 1
21. **SET** @a = @a + 1
22. **END**
23. IF (@count = 2)
24. **set** @result = @result+cast(@i **as** **varchar**(10))+' , '
26. **SET** @i = @i + 1
27. **END**
29. **set** @result = (**select** substring(@result, 1, (len(@result) - 1)))
30. print(@result)

**Q26 Write query to print numbers from 1 to 100 without using loops**

**Answer**

This can be done using Common Table Expression without using a loop.

1. ;**with** numcte
2. **AS**
3. (
4. **SELECT** 1 [**SEQUENCE**]
6. **UNION** ALL
8. **SELECT** [**SEQUENCE**] + 1 **FROM** numcte **WHERE** [**SEQUENCE**] <100
9. )
11. **SELECT** \* **FROM** numcte

**Q.27 What will be the output of following SQL?(tricky)**

1. **Select** $
3. Options  -
4. a. 0.00,
5. b. $,
6. c. 0,
7. d. Syntax Error
9. Answer  = 0.00

**Q.28 What will be the output of following SQL queries?**

1. **Select** **select** 'TD'
2. Options -
3. 1. TD,
4. 2. Syntax Error,
5. **3. select** TD
7. Answer - Syntax Error. (Incorrect syntax near the keyword 'select'. )
9. **select** \* **from** 'Employee'
11. Answer -  Incorrect syntax near 'Employee' .

**Q29 What will be the outputs in the following SQL queries with aggregate functions?**

1. **SELECT** SUM (1+4\*5)
3. Options - a.21,      b.25,         c.Error        d.10
5. Answer -: 21

8. **SELECT** **MAX**  (1,3,8)
10. Options - a.8,        b. 12,        c.Error        d.1
12. Answer -: Error. **Max** **function** takes **only** 1 argument.

15. **SELECT** **Max** ('TD')
17. Options -  a.TD         b. Error      c. 1       d.0
19. Answer-: TD

22. **SELECT** **Max** ('TD'+'AD')
24. Options -  a.TDAD         b. Error      c. T2D       d.0
25. Answer-: TDAD

**Q.30 What will be the output of following queries? [Tricky involving 0]**

1. **SELECT** 0/0
3. A. Divide **by** 0 error,   B. 0
4. C. NULL,                   D. Incorrect syntax error
6. Answer -:  Divide **by** 0 error
8. **SELECT**  0/6
10. A. Divide **by** 0 error,   B. 0
11. C. 6,                         D. Incorrect syntax error
13. Answer -:  0

**Q31 What will be the output of given statement?**SELECT SUM (NULL)   
  
**Answer** = Error. Cannot pass null type in SUM function.  
  
Operand data type NULL is invalid for avg operator. 

**Q32 What will be the output of given statement?**SELECT

MAX (NULL)  
  
**Answer** = Error. Operand data type NULL is invalid for MAX operator.  
  
**Q. 33 Will following statement give error or 0 as output?**   
  
SELECT

AVG (NULL)  
  
Answer = Error. Operand data type NULL is invalid for Avg operator.   
  
**Note**MIN, MAX,SUM,AVG none of these function takes NULL parameter/argument. Also, these functions accept only one argument.

**Q.34 Will the following statements execute? if yes what will be output?**SELECT NULL+1

SELECT NULL+'1'  
  
**Answer** - Yes, no error. The output will be NULL. Perform any operation on NULL will get the NULL result.  
  
**Q35 Will following statement give Divide by Zero error?**SELECT NULL/0  
  
**Answer**- No. It will execute and result will be NULL.