



1. What is subquery and what are different types of subqueries?

Answer:

Subquery is query within a query. There are two or more types of queries used to fetch output of the query. One is query used inside the query, which is known as inner query, and the output of query is given to the query, which is used outside. The query used outside is called as outer query. Subquery is made up of combining two types of Queries the first type is outer query and other is inner query.

Following are different types of subqueries:

1. Single Row Subquery
2. Multi row Subquery
3. Co-Related Subquery
4. Scalar Subquery
5. Nested Subquery.

Example of Subquery:

Select e.Eno,e.Ename from Employee e where (This is outer query) Ename=

(Select E2.Ename from Employee E2 where E2.Name='Amit')(Inner Query Executes first);

2. What is correlated Subquery?

Answer:

Correlated query is the query, which is executed after the outer query is executed. The outer query is always dependent on inner query. The approach of the correlated subquery is bit different from normal subqueries. In normal subqueries, the inner queries are executed first and then the outer query is executed but in Correlated Subquery outer query is always dependent on inner query so first outer query is executed then inner query is executed. Correlated Subqueries always uses operator like **Exist, Not Exist, IN, Not IN**. (Source-[click here](#))

3. Explain example of correlated subquery?

Answer:

Select * from Employee E where Not exist

(Select Department_no from Department D where E.Employee_id=D.Employee_ID);



Step 1:

Select * from Employee E;

It will fetch the all employees.

Step 2:

The First Record of the Employee second query is executed and output is given to first query.

(Select Department_no from Department D where E.Employee_id=D.Employee_ID);

Step 3:

Step 2 is repeated until and unless all output is been fetched. (Source-[Click here](#))

4. What is faster? In Operator in SQL or Exist in SQL?

Answer:

In Operator is slower than Exist operator is. In Operator scans full table and checks the each value of the table. Exist operator only checks whether the value is present in the table or not.

5. What is query to calculate second highest salary of the employee using analytical function?

Answer:

We need to use Dense_Rank function to calculate second highest salary:

Select * from (Select Dense_Rank () over (order by salary desc) as Rnk, E.* from Employee E) where Rnk=2;

6. How can I create table with same structure of Employee table?

Answer:

Create table Employee_1 as Select * from Employee where 1=2;

7. What are Set operators in SQL?

Answer:

Set operators are nothing but the operators, which are used to connect two tables and fetch the records from the two tables. We need to follow one condition that the table set 1 columns and table set 2 columns are same and its datatype must be same. SQL Set Operators combines the result of 2 queries or components on to the single result.



8. Which are different set operators in SQL?

Answer:

Following are Set Operators in SQL:

1. Union
2. Union all
3. Intersect
4. Minus

9. What is Union Operator?

Answer:

Union Operator combines the result of 2 or more tables and fetches the results of two select statements. Union operator eliminates the duplicates from the table and fetches the result. For each duplicate row in table, only one row is displayed in the result. By considering the performance of SQL, using union is not preferable option but if there is situation where user wants to remove the duplicate data from two or more table the use of Union is preferable.

Syntax:

Select column1...column n from table1;

Union

Select column1...column n from table2;

10. Give us one example of union operator?

Answer:

Example/Real Life Scenario:

Kindly select employees from 2 different employee tables eliminating duplicates

Suppose following is first table:



Table name: Employee_OBIEE

Employee_num	Employee_name	Department	Salary
1	Amit	OBIEE	680000
2	Rohan	OBIEE	550000
3	Rohit	COGNOS	430000

Second table

Table Name: Employee_Cognos

Employee_num	Employee_name	Department	Salary
1	Pradnya	Cognos	522000
2	Mohit	Cognos	471100
3	Rohit	COGNOS	430000

Query:

Select Employee_Num, Employee_name, Department, Salary from Employee_OBIEE;

Union

Select Employee_Num, Employee_name, Department, Salary from Employee_COGNOS;

Here Duplicate record has been removed for Employee name Rohit and output is as follows:

Employee_num	Employee_name	Department	Salary
1	Amit	OBIEE	680000
2	Rohan	OBIEE	550000
3	Rohit	COGNOS	430000
1	Pradnya	COGNOS	522000
2	Mohit	COGNOS	471100

**11. What is union all operator in sql?****Answer:**

Union ALL Operator combines the result of 2 or more tables and fetches the results of two or more select statements. Union not all operator eliminates duplicate values. It shows duplicate records also. By considering the performance of SQL, using union all is preferable option because it does not check the duplicate values so no sorting required at the time of fetching the records. Union all operator is most widely used operator in reporting purpose where user needs to fetch the records from different tables.

Syntax:

Select column1...column n from table1;**Union all****Select column1...column n from table2;****12. Explain example of union all operator?****Answer:**

Kindly select employees from 2 different employee tables without eliminating duplicates

Suppose following is first table:

Table name:

Employee_OBIEE

Employee_num	Employee_name	Department	Salary
1	Amit	OBIEE	680000
2	Rohan	OBIEE	550000
3	Rohit	COGNOS	430000

Second table

Table Name: Employee_Cognos



Employee_num	Employee_name	Department	Salary
1	Pradnya	Cognos	522000
2	Mohit	Cognos	471100
3	Rohit	COGNOS	430000

Query:

Select Employee_Num, Employee_name, Department, Salary from Employee_OBIEE;

Union All

Select Employee_Num, Employee_name, Department, Salary from Employee_COGNOS;

Result:

Employee_num	Employee_name	Department	Salary
1	Amit	OBIEE	680000
2	Rohan	OBIEE	550000
3	Rohit	COGNOS	430000
1	Pradnya	COGNOS	522000
2	Mohit	COGNOS	471100
3	Rohit	COGNOS	430000

13. What is difference between union and union all?

Answer:

Union	Union ALL
1.Union Set operator is used to fetch the records from 2 different tables which eliminates the duplicate records	1.Union all Set operator is used to fetch the records from 2 different tables which does not eliminates the duplicate records
2.Syntax:	2.Syntax:



Select col1,col2...from table1; Union Select col1,col2...from table2;	Select col1,col2...from table1; Union all Select col1,col2...from table2;
3.For Performance tuning Union operator is not preferable as it takes time to eliminate duplicate records	3. Union all is preferable operator in Performance tuning.

14. Which operator is faster operator?

Answer:

Union all is faster than union because it does not have sorting program running inside. The union operator is slower than union all because it removes duplicates from the data. To Remove Duplicate compiler needs to run sorting program and table is scanned full.

Therefore, Union all is faster than union operator is.

15. Why union all is faster than union operator is?

Answer:

Union all is faster than union because it does not have sorting program running inside. The union operator is slower than union all because it removes duplicates from the data. To Remove Duplicate compiler needs to run sorting program and table is scanned full.

Therefore, Union all is faster than union operator is.

16. What is difference between varchar and varchar2 datatype?

Answer:

Varchar can store up to 2000 bytes and varchar2 can store up to 4000 bytes of memory space. Varchar will occupy the space for null values whereas varchar2 cannot occupy the space for null values. So varchar2 is good to use not to face performance related problems. varchar2 is faster than varchar datatype.



17. What is Rowid in sql?

Answer:

1. ROWID is nothing but the physical memory location on which that data/row is stored. ROWID returns address of row.
2. ROWID uniquely identifies row in database.
3. ROWID is combination of data object number, data block in data file, position of row and data file in which row resides.
4. ROWID is 16 digit hexadecimal number whose datatype is also ROWID Or UROWID
5. The fastest way to access a single row is ROWID
6. ROWID is unique identifier of the ROW.

18. What is main use of rowid in sql?

Answer:

When there is no primary key or unique key constraints on oracle then rowid is used as unique identifier to identify unique records in the table.

19. Explain example of rowid?

Answer:

Example: - select rowid from dual;

AAAAECAABAAAAgiAAA

20. What is rownum in sql?

Answer:

1. ROWNUM is magical column in Oracle, which assigns the sequence number to the rows retrieves in the table.
2. To limit the values in the table you can use rownum pseudo column
3. ROWNUM is nothing but logical sequence number given to the rows fetched from the table.
4. ROWNUM is logical number assigned temporarily to the physical location of the row.
5. You can limit the values in the table using rownum
6. ROWNUM is also unique temporary sequence number assigned to that row.

21. What will be the output of following statement?

Select rownum from dual;

Select * from dual;



Answer:

As dual is dummy table of one row and one column the rownum is one.

In addition, Select * from dual fetches 'X' as record.

22. What is difference between rowid and rownum?

Answer:

<u>ROWID</u>	<u>ROWNUM</u>
<ol style="list-style-type: none"> 1. ROWID is nothing but Physical memory allocation 2. ROWID is permanent to that row which identifies the address of that row. 3. ROWID is 16 digit Hexadecimal number that is uniquely identifies the rows. 4. ROWID returns PHYSICAL ADDRESS of that row. 5. ROWID is automatically generated unique id of a row and it is generated at the time of insertion of row. 6. ROWID is the fastest means of accessing data. 	<ol style="list-style-type: none"> 1. ROWNUM is nothing but the sequence, which is allocated to that data retrieval bunch. 2. ROWNUM is temporarily allocated sequence to the rows. 3. ROWNUM is numeric sequence number allocated to that row temporarily. 4. ROWNUM returns the sequence number to that row. 5. ROWNUM is a dynamic value automatically retrieved along with select statement output. 6. ROWNUM is not related to access of data.

23. What is output of following sql statement where Employee table contains 100 records?

Select * from Employee where rownum > 1;

Answer:

No rows found.



24. What will be the count of records of following statement?

Select * from employee where rownum > 1 and rownum <9;

Answer:

The seven records displayed if the Employee table contains more than seven records.

25. How to display 1 to 100 Numbers with query?

Answer:

Select level from dual connect by level <=100;

26. What is the query to fetch first record of employee table?

Answer:

Select * from Employee where [Rownum](#) =1;

27. What is query to fetch nth record from the table?

Answer:

The following query is used to fetch Nth record from the table.

Select * from tablename where rownum= &N; where &N is Nth number of record.

28. What is difference between group by and having clause?

Answer:

Both Where Clause and having clause are filters but these two filters are used for two different purposes. Where clause is used to filter the non-aggregated values and having clause is used for aggregated values. Simply Where clause is used in Static non aggregated values and having clause is used in aggregated values where you are using aggregated functions like Count,Sum,Avg.etc.Having Clause always come with group by clause, you cannot use having clause directly.

Example:

Select * from Employees where name='Amit';

Select count(*) from Employees group by name having salary > 10000;

**29. What is DDL in SQL?**

Answer:

DDL stands for Data Definition Language. Following are the different DDL statements:

Statement	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

30. What are different parameters to consider the database performance of Application?

Answer:

There following parameters to consider the performance of application:

1. What size of images we are using in application. The images we are using on application should not be maximum size.
2. What is the data volume used to fetch the data
3. Data cardinality: The most important factor is data cardinality of the data in application. Data should be divided in proper manner and the database should be in well-normalized form
4. Indexing done: Indexing should be done properly on database table columns.

31. How many long columns allowed in table?

Answer:

Only one long datatype column is allowed in one table in oracle.

32. How to find table name and its owner?

Answer:

Make sure that the database user have logged in with SYS user.

Select table_name, Owner from **All_tables** order by table_name, owner;

33. How to find Selected Tables from a User?

Answer:

SELECT Table_Name FROM **User_Tables** WHERE Table_Name LIKE 'STU%';



34. How to Select Users from Database?

Answer:

```
SELECT Username FROM All_Users ORDER BY Username;
```

35. How to check all oracle views created on the database?

Answer:

```
Select VIEW_NAME, OWNER from ALL_VIEWS order by OWNER, VIEW_NAME;
```

36. How to find all details about Constraints?

Answer:

```
SELECT * From User_Constraints;
```

```
SELECT * FROM User_Cons_Columns;
```

37. How to find Constraint Name?

Answer:

```
SELECT Table_Name, Constraint_Name FROM User_Constraints;
```

38. How to find Constraint Name with Column_Name?

Answer:

```
SELECT Column_Name, Table_Name, Constraint_Name FROM User_Cons_Columns;
```

39. How to find Selected Tables, which have Constraint?

Answer:

```
SELECT Table_Name FROM User_Cons_Columns WHERE Table_Name LIKE 'STU%';
```

40. How to find Constraint_Name, Constraint_Type, Table_Name?

Answer:

```
SELECT Table_Name, Constraint_Type, Constraint_Name FROM User_Constraints;
```

```
SELECT Table_Name, Constraint_Type, Constraint_Name, Generated FROM User_Constraints;
```



41. How to check Sequences?

Answer:

```
SELECT * FROM USER_SEQUENCES;
```

42. How to check Procedures?

Answer:

```
SELECT * FROM User_Source  
  
WHERE Type='PROCEDURE'  
  
AND NAME IN ('SP_CONNECTED_AGG','SP_UNCONNECTED_AGG');
```

43. How to find procedure columns information?

Answer:

```
Select OWNER, OBJECT_NAME, ARGUMENT_NAME, DATA_TYPE, IN_OUT from ALL_ARGUMENTS order  
by OWNER, OBJECT_NAME, SEQUENCE;
```

44. How to find functions created in database?

Answer:

```
Select OBJECT_NAME, OWNER from ALL_OBJECTS where upper (OBJECT_TYPE) = upper ('FUNCTION')  
order by OWNER, OBJECT_NAME ;
```

45. How to find indexes used in database?

Answer:

```
Select * from USER_INDEXES;  
  
Select * from ALL_IND_COLS where Index_name='Name of Index';
```

46. How to find trigger information in SQL?

Answer:

```
Select TRIGGER_NAME, OWNER from ALL_TRIGGERS order by OWNER, TRIGGER_NAME;
```



47. How to find DB name?

Answer:

```
SELECT Ora_Database_Name FROM DUAL;
```

```
SELECT * FROM GLOBAL_NAME;
```

```
SELECT Name from V$DATABASE;
```

48. What is query to find duplicate records from the table?

Answer:

```
Select a.* from Employee a where rowid!=  
      (select max (rowid) from Employee b where a.Employee_num =b.Employee_num;
```

49. What is query to delete duplicate records from the table?

Answer:

```
Delete from Employee a where rowid !=  
      (Select max (rowid) from Employee b where a.Employee_num =b.Employee_num;
```

50. How to find count of duplicate rows?

Answer:

```
Select rollno, count (rollno) from Student
```

```
Group by rollno
```

```
Having count (rollno)>1
```

```
Order by count (rollno) desc;
```

51. How to Show the Max marks and min marks together from student table?

Answer:

```
Select max (marks) from Student
```

```
Union
```

```
Select min (marks) from Student;
```



52. How to create table with primary key?

Answer:

```
CREATE TABLE TABLE_NAME  
  
(Column_name1 Datatype (Size),  
  
Column_name2 Datatype (Size),  
  
.  
  
.  
  
.  
  
Column_name3 Datatype (size);  
  
Constraint Constraint_name Primary Key (Column_name));
```

Example of create table with primary key (Oracle / SQL Server):

```
CREATE TABLE STUDENT_Primary  
  
(Rollno Number (3, 0),  
  
NAME VARCHAR2 (30),  
  
Standard VARCHAR2 (10),  
  
Constraint pk_roll Primary key (Rollno));
```

The table named STUDENT_primary is created with Rollno as primary key constraint. If user want to check the constraint use, the table named all_constraints and put the filter of the table. There are following type constraints available in all_constraints table.

53. How to identify constraints in constraints table?

Answer:

Query:

```
Select Constraint_name, Constraint_type from all_constraints where table_name=upper  
(‘STUDENT_Primary’);
```



List Of Constraints identification from All Constraints table :

C – Check constraint on a table

P – Primary key

U – Unique key

R – Referential integrity

V – With check option, on a view

O – With read only, on a view

H – Hash expression

F – Constraint that involves a REF column

S – Supplemental logging

54. How to create replica of other table?

Answer:

Create table table_name as select * from table_name1;

The above statement will create the replica of table named 'table_name1'.

55. How to display following using query?

*

**

Answer:

There are two ways to do this.

With using dual table:



Select '*' from dual

Union

Select '**' from dual

Union

Select '***' from dual;

Using LPAD function:

```
SELECT lpad ('*', ROWNUM, '**') FROM Student WHERE ROWNUM <4;
```

56. How to add constraints on already created table?

Answer:

To add the constraint to already created table we need to use following syntax:

```
ALTER TABLE TABLENAME
```

```
ADD Constraint Constraint_name PRIMARY KEY (Column_name);
```

57. What is the use of NVL function in Oracle?

Answer:

NVL function is most important function to replace null value with another value.

Example: select NVL (null, 'Amit') from dual;

Which will give you output as Amit.

58. What is raw datatype?

Answer:

Raw datatype is used to store values in binary data format.

There are 2 types of RAW datatype:

1. Raw
2. Long Raw.

Long raw datatype is used to store graphics, sound documents.



Raw datatype is variable length datatype like varchar2 but basically it only stores data in 1's and 0's means binary data format.

59. What is Query to drop all user tables from Oracle?

Answer:

To Drop all tables user needs to write simple PLSQL block

Begin

For l In

(Select * from Tabs) —Tabs is system table in which user get the different user defined table names.

Loop

Execute immediate ('Drop Table ' || i.table_name || 'cascade constraints');

End loop;

End;

60. What is query to fetch last day of previous month in oracle?

Answer:

Select LAST_DAY (ADD_MONTHS (SYSDATE,-1)) from dual;

61. How to display the String vertically in Oracle?

Answer:

SELECT SUBSTR ('AMIET', LEVEL, 1) FROM dual

Connect by level <= length ('AMIET');

62. Write query to find the repeated characters from string?

Answer:

Select regexp_count ('AmitA','A') as Repeated_character from dual;



63. How to convert seconds in to time format?

Answer:

```
SELECT  
  
TO_CHAR (TRUNC (2700/3600),'FM9900') || ':' ||  
  
TO_CHAR (TRUNC (MOD (2700, 3600)/60),'FM00') || ':' ||  
  
TO_CHAR (MOD (2700, 60),'FM00')  
  
FROM DUAL;
```

Where 2700 is seconds.

Output:

00:45:00

64. What is materialized view in SQL?

Answer:

Materialized view is also logical structure of one or more table in which data is stored physically in the view. Data has been stored physically in materialized view so data retrieval is faster as compare to simple view.

65. What are different options of materialized view?

Answer:

There are following different options we used to create materialized view or snapshot.

1. Build Immediate:

Means materialized views (mv) created immediately.

2. Build Deferred:

Means materialized views (mv) created after one refresh.

3. Refresh on commit:

This option committed the data in materialized view in SQL immediately after data inserted and committed in table. This option is known as incremental refresh option. View is not fully refreshed with this option



4. Refresh on Demand:

Using this option, you can add the condition for refreshing data in materialized views.

66. What is difference between view and materialized view?

View	Materialized Views(Snapshots)
1. View is nothing but the logical structure of the table, which will retrieve data from 1 or more table.	1. Materialized views (Snapshots) are also logical structure but data is physically stored in database.
2.You need to have Create view privileges to create simple or complex view	2.You need to have create materialized view 's privileges to create Materialized views
3.Data access is not as fast as materialized views	3.Data retrieval is fast as compare to simple view because data is accessed from directly physical location
4.There are 2 types of views: 1.Simple View 2.Complex view	4.There are following types of Materialized views: 1.Refresh on Auto 2.Refresh on demand
5.In Application level views are used to restrict data from database	5. Materialized Views are used in Data Warehousing.

67. What is Incremental or Fast Refresh?

Answer:

When in database level some DML changes are done then Oracle Database stores rows describing those changes in the materialized view log and then uses the materialized view log to refresh materialized views based on the master table. This process is called **incremental or fast refresh**.

In this situation, materialized view is not full refresh and it only refresh the data on incremental basic and only incremental data is added in to materialized view.

68. What is Complete Refresh?

Answer:

Without a materialized view log, Oracle Database must re-execute the materialized view query to refresh the materialized view. This process is called **a complete refresh**. Usually, a fast refresh takes less time than a complete refresh. A materialized view log is located in the master database in the same schema as the master table. A master table can have only one materialized view log defined on it. Oracle Database can use this materialized view log to perform fast refreshes for all fast-refreshable materialized views based on the master table. To fast refresh a materialized join view, you must create a materialized view log for each of the tables referenced by the materialized view.



69. How materialized views are used in different scenarios?

Answer:

Materialized Views are basically used in the scenarios where actual performance tuning for query is needed. Materialized views are used mostly in reports where user wants to fetch the records very fast. I will try to explain the real life scenario where exactly materialized view is useful. When user creates materialized view, then one table structure is created and user directly fetches that data from that table structure.

Suppose there are 2 tables named Employee and Department. The Employee table contains 1 million records and department table contains 20 records. We need to fetch the Employees associated with that department.

Step 1:

To perform above scenario we basically create view:

Create View V_Employee

as

```
Select E.Employee_num,E.Employee_name,D.Department_Name
```

```
from Employee E, Department D where E.Dept_no=D.Dept_no;
```

Step 2:

Fetch the records from the View

```
Select * from V_Employee;
```

It will fetch 10 million records with associated department. But to fetch that records check the time. Let us consider it will take 2 Mins means 120 secs to fetch records

Step 3 :

Let us Create materialized view which will refresh automatically.

Create or Replace Materialized view MV_Employee

as

```
Select E.Employee_num,E.Employee_name,D.Department_Name
```

```
from Employee E , Department D where E.Dept_no=D.Dept_no
```



Refresh auto on commit select * from Department;

We have created materialized view in sql for that.and lets check performance.

Select* from MV_Employee;

It will fetch 1 million records in 60 secs.So performance is improved double when you use materialized view.

70. What is snapshot in SQL?

Answer:

Snapshot is nothing but the materialized view in sql. MVs are used in data-warehouse like aggregate materialized views, materialized view with joins etc. MVs are used in data-warehouse like aggregate materialized views, materialized view with joins etc.

71. How to get 3 Highest salaries records from Employee table?

Answer:

select distinct salary from employee a where 3 >= (select count(distinct salary) from employee b where a.salary <= b.salary) order by a.salary desc;

72.How to display odd records in table?

Answer:

Select * from (Select rownum as rno,E.* from Employee E) where Mod(rno,2)=1;

73. What is pivot in sql?

Answer:

One of the function named Pivot is new Oracle 11 G feature which is used specifically to transpose or convert rows in to columns or columns in to rows (Unpivot) to display the result in crosstab format. The simple meaning of Pivot in English is 'Center point on which mechanism turns or oscillates'. Just like that, Pivot in SQL is used to convert the column values in to attributes (transpose rows in to columns).

"Pivot in SQL helps to convert column values into attributes or transpose rows into columns."



74.What are different parameters of pivot table?

Answer:

Following are different parameters and arguments used in Pivot/Unpivot:

1. Pivot Clause with Aggregate Function:

There must be the aggregate function to Pivot the table. The aggregate functions like Sum, Avg,Min,Max and Count needs to be used for pivoting table.

2.Pivot For Clause:

The Column name which needs to be converted from rows to column.

3.Pivot IN Clause:

These are nothing but the list of values to column 2 to pivot it in to headings in to cross table result.

4.Subquery:

We need to use the Subquery for fetching the records instead of list of values.In this case result of subquery would be used to determine the values from column to pivot in to headings.

75. Explain example of pivot?

Answer:

Consider Following table.We need count of Employees department wise where Department ID is column.

Name of Table: Department

Department ID	Employee Name
100	Amit
100	Rohan
101	Rohit
102	Pradnya

We need to convert the Department ID column in to Rows and then We need to display the count of employees .

Select * from

(Select DepartmentId from Department)



PIVOT

(Count(Employee_name)

For DepartmentId IN (100,101,102)

);

Using above Pivot statement the DepartmentId is pivoted and the table is been transposed and we are using Count() as aggregate function

Output:

100	101	103
2	1	1

76. What is database normalization? What is use of normalization?

Answer:

Database Normalization is nothing but technique of designing the database in structured way to reduce redundancy and improve data integrity.

Database Normalization is used for following Purpose:

1. To Eliminate the redundant or useless data
2. To Reduce the complexity of the data
3. To Ensure the relationship between tables as well as data in the tables
4. To Ensure data dependencies and data is logically stored.

77. Which are different database normalization forms?

Answer:

There are following Four Normal Forms used in Database Normalization:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. Boyce-code Normal Form(BCNF)

**78. How rank function is used as aggregate function?****Answer:**

Rank function is used as aggregate function to return the rank of rows in the table within group of rows. If someone needs to find out the rank of specific row in the table then we will use the rank function.

Rank Function Syntax:

RANK(expr1 [, expr2, ... expr_n]) WITHIN GROUP (ORDER BY expr1 [, expr_2, ... expr_n]);

Real Life Example:

Consider following table:

Employee_num	Employee_name	Department	Salary
1	Amit	OBIEE	680000
2	Rohan	OBIEE	550000
3	Rohit	COGNOS	430000

Find out Rank of Employee whose Department is OBIEE and Name is rohan?

select RANK(Rohan, OBIEE) WITHIN GROUP (ORDER BY Name, Department) from employees;

79. What is rank as analytical function?**Answer:**

Rank function is used as aggregate function to return the rank of rows in the table within group of rows. If someone needs to find out the rank of specific row in the table then we will use the rank function.

Rank Function Syntax:

RANK(expr1 [, expr2, ... expr_n]) WITHIN GROUP (ORDER BY expr1 [, expr_2, ... expr_n]);



80. What is difference between unique and distinct?

Answer:

There is no difference between unique and distinct keywords apart from one difference. Unique is applied before insertion and retrieval. It consists of non-duplicate values. If unique constraint is given, it does not take duplicate values. Distinct is used in retrieval it gives the suppressed row (ex if two rows are same it will show single row and non-duplicate row) therefore distinct is the combination of suppressed duplicate and non-duplicate rows. Specify DISTINCT or UNIQUE if you want Oracle to return only one copy of each set of duplicate rows selected (these two keywords are synonymous). Duplicate rows are those with matching values for each expression in the select list.

Therefore, there is no functional difference between Unique and distinct both have same functionalities.

81. What is self-join?

Answer:

Self-join is nothing but the table joins with itself. There are lot of tables which contains more than one functionality at that time the concept of self-join comes to the picture.

Syntax:

```
SELECT a.column_name, b.column_name...
FROM table1 a, table1 b
WHERE a.common_field = b.common_field;
```

Real Example:

If one table contains Employee and its position. If we need to fetch the Employees and its managers then self-join is used.

82. What are indexes in SQL?

Answer:

“Index is optional structure associated with the table which may or may not improve the performance of Query”

In simple words, suppose we want to search the topic in to book we go to index page of that book and search the topic, which we want. Just like that to search the values from the table when indexing is there you need not use the full table scan.



83. What are advantages of Indexes?

Answer:

Indexes are memory objects, which are used to improve the performance of queries, which allows faster retrieval of records.

Following are advantages of Indexes:

1. It allows faster retrieval of data
2. It avoids the Full table scan so that the performance of retrieving data from the table is faster.
3. It avoids the table access altogether
4. Indexes always speeds up the select statement.
5. Indexes used to improve the Execution plan of the database

84. What is parser?

Answer:

When SQL Statement has been written and generated then first step is parsing of that SQL Statement. Parsing is nothing but checking the syntaxes of SQL query. All the syntax of Query is correct or not is checked by SQL Parser.

There are 2 functions of parser:

1. Syntax analysis
2. Semantic analysis

85. What is composite index?

Answer:

When 2 or more columns are related to each other in the table and the same columns are used in where condition of the query then user can create index on both columns. These indexes are known as composite indexes.

Example:

Create index CI_Employee on Employee(Eno,Deptno);



86. What are clustered indexes?

Answer:

1. The clustered indexes are indexes, which are physically stored in order means it stores in ascending or descending order in Database.
2. Clustered indexes are created once for each table. When primary key is created then clustered index has been automatically created in the table.
3. If table is under heavy data modifications the clustered indexes are preferable to use.

87. What is Unique key constraint?

Answer:

The UNIQUE Constraint uniquely identifies each record in a database table.

The UNIQUE and PRIMARY KEY Constraints both provide a guarantee for Uniqueness for a column or set of columns.

A PRIMARY KEY Constraint automatically has a UNIQUE Constraint defined on it.

Note: You can have many UNIQUE Constraints per table, but only one PRIMARY KEY Constraint per table is allowed.

88. What is mean by Sequence in database?

Answer:

Use the CREATE SEQUENCE statement to create a **sequence**, which is a database object from which multiple users may generate unique integers. You can use sequences to automatically generate primary key values.

When a sequence number is generated, the sequence is incremented, independent of the transaction committing or rolling back.

Once a sequence is created, you can access its values in SQL statements with the CURRVAL Pseudo Column, which returns the current value of the sequence, or the NEXTVAL Pseudo Column, which increments the sequence and returns the new value.



89. What is Bit-map index?

Answer:

1. If Table contains the distinct values, which are not more than 20 distinct values, then user should go for Bit map indexes.
2. User should avoid the indexing on each row and do the indexing only on distinct records of the table column. You should be able to check drastic change in query cost after changing the normal index to Bit map index.
3. The bit map indexes are very much useful in data ware housing where there are low level of concurrent transactions. Bit map index stores row_id as associated key value with bitmap and does the indexing only distinct values.
4. Means If in 1 million records only 20 distinct values are there so Bitmap index only stores 20 values as bitmap and fetches the records from that 20 values only.

Syntax:

Create bitmap index Index_name on Table_name(Columns which have distinct values);

Example:

CREATE BITMAP index BM_DEPT_NAME on DEPT(Department_name);

90. What is difference between NVL, NVL2 and Nullif?

Answer:

1.NVL :

NVL function substitutes a value when a null value is encountered.

2.NVL2 :

NVL2 substitutes a value when a null value is encountered as well as when a non-null value is encountered.

3.NULLIF:

NULLIF function compares expr1 and expr2. If expr1 and expr2 are equal, the NULLIF function returns NULL. Otherwise, it returns expr1.



91. What is mean by all and any operator in SQL?

Answer:

1.ALL Operator:

ALL Operator is used to compare all the values from another dataset and fetch the specific records from the table.

Example:

```
Select * from Student where 10000 >ALL (Select Fees from Student_Fees);
```

The above statement will display all the records from Student table where Student fees is > 10000 which we are fetching from different table named 'Student_Fees'

2.Any Operator:

Any Operator is used to compare the values from Any value from another table with that condition.

Example:

```
Select * from Student where 10000 >ANY (Select Fees from Student_Fees);
```

The above statement will display ANY of the records from Student table where Student fees is > 10000 which we are fetching from different table named 'Student_Fees'.

92. What is mean by SQL Scalar Functions?

Answer:

SQL scalar functions are the functions whose input range is one-dimensional and which returns the single output for each row. SQL Scalar functions returns value of every row, which we are used in query to process.

SQL Scalar Functions are also known as Single Row Functions..

Following is Single line Explanation of some of the useful Scalar Functions:

- **UCASE()** – Converts a field to upper case
- **LCASE()** – Converts a field to lower case
- **SUBSTR()** – Extract characters from a text field
- **LEN()/LENGTH()** – Returns the length of a text field
- **ROUND()** – Rounds a numeric field to the number of decimals specified



93. What is join in SQL?

Answer :

The most used concept in real life scenarios are nothing but SQL Joins. Although in reporting, stand alone applications development, Web application development the concept of join is really important.

Joins are nothing but Combining the records from two or more tables.

There are following 2 types of joins:

1. Joins using Operators -> Equi Join, Non Equi Join
2. Joins using Concept -> Inner Join, Outer Join, Cross Join, Self Join.

94. Explain about sql joins?

Answer:

Join is nothing but connecting 2 tables to fetch the records from 2 or more different tables. There are following types of joins in SQL:

1. Inner join:

Inner join retrieves the records which are common between 2 or more tables.

2. Outer join:

Outer join retrieves the common records from the table as well as uncommon records from Left or right table.

2.1. Left outer join:

When user needs to fetch all data from left table and common records from left and right table then the join is called as left outer join.

2.2. Right outer join:

When user needs to fetch all data from right table and common records from left and right table then the join is called as right outer join.

2.3. Full Outer Join:

When user needs to fetch the data from both the tables and common records from both of the tables.



3. Cross join/Cartesian join:

When each and every record is connected to each and every record from other table then it is called as cross join or Cartesian join.

95. What is index and what are types of indexes?

Answer:

Indexing is nothing but the performance tuning mechanism, which allows the fast retrieval of the records from table.

Following are types of indexes:

1. Normal Indexes
2. Bit Map indexes
3. Unique indexes
4. Clustered Indexes
5. Non Clustered Indexes

96. How do I fetch only common records between 2 tables.

Answer:

```
Select * from Employee;
```

```
Intersect
```

```
Select * from Employee1;
```

97. How to get distinct records from the table without using distinct keyword.

Answer:

```
select * from Employee a where rowid = (select max(rowid) from Employee b  
where a.Employee_no=b.Employee_no);
```


**98.What is mean by table partition?****Answer:**

partition allows table,index or index organized tables to be subdivided in to smaller pieces and each piece of table,index or index organized table is called as Partition.

Following are Advantages of Partition:

- 1.Increase Performance
- 2.Increases availability
- 3.Enable storage cost optimization
- 4.Enables Simpler management

99.What are different types of partitions?**Answers:**

There are following types of Table partition:

- 1.Range Partition
- 2.List Partition
- 3.Hash Partition

100.What is difference between Oracle 11G and Oracle 12c?**Answer:**

Oracle 11G	Oracle 12C
Indexing: User can create only one index on one column. The Invisible index feature has been implemented in Oracle 11G.	Indexing: User can create more than one index on same column. Only one type of index on column is usable at a same time.
Newly added Functions: The newly added function are not considering the performance tuning of the data. The newly added functions in oracle 11G are LISTAGG and Nth_value functions.	Newly added Functions: Oracle 12C has added the new feature approx_count_distinct () which provides approximate count distinct aggregation.
Caching in Oracle 11G: The Oracle 11G caching In memory parallel query did not working well with multiple scans contended for cache memory.	Caching in Oracle 12C: The new cache mechanism called as big table cache is used to improve the performance for full table scan. This new concept is known as Automatic big table caching in Oracle 12C.Big table cache provides significant performance improvement for full table scan.



Full Database caching is not implemented in Oracle 11G.	Oracle 12C provides the feature of full database caching to improve the significant performance benefits especially for workloads that will be previously limited by I/O throughput or response time.
In Memory table : Oracle 11G supports the concept of In-memory tables, which will, used to improve the full table scans. The user needs to put the table in the memory so that user can access the fast data. Oracle 11G does not support the In memory aggregation concept.	In Memory aggregation : Oracle 12C supports the In memory aggregation concept which is very useful in star queries.
Table partition and subpartition movement : To move the partition and subpartition from one tablespace to other tablespace user needs to write complex procedural logic.	Table partition and subpartition movement : To migrate the table partition and subpartition oracle 12c uses 2 methods one is with online keyword and second is offline method.
Invisible columns: In Oracle 11g,the couple of good enhancements introduced in form of invisible indexes and virtual columns. The invisible column has not been introduced in oracle 11g.	Invisible columns : In Oracle 12c R1,User can define the invisible columns in the table. When column is defined as invisible column, it will not come in generic query. It is explicitly referred to SQL statement or condition in SQL statement.

Instructions:

1. All Rights of this document is reserved to www.complexsql.com
2. Nobody have copy the content of this document.
3. Anyone found copying this content will be the policy violation
4. This File is secured with password and will be shared with the owner of this document.

All Rights [Reserved@ www.complexsql.com](http://www.complexsql.com)



All Rights [Reserved@ www.complexsql.com](http://www.complexsql.com)



All Rights [Reserved@ www.complexsql.com](http://www.complexsql.com)

