1. **What is Oracle?**

Ans: - Oracle provides a comprehensive high performance for E-Business.

2. **What are features of Oracle 9*i*?**

Ans: - Scalability from department to enterprise e-business sites.
Robust, Reliable, secured architecture.
Single development model, easy deployment option.
Leverage an organizations current skillset throughout the oracle (including SQL, PL/SQL, and Java). One management interface for all application industry standard technologies, no proprietary locks in. In short Scalability, Reliability, Single Devlopment Model, Common skill set, one management interface.

3. **What is Database?**

Ans: - Database is an organized collection of information.

4. **What is Relational Database?**

Ans: - Relational Database is the collection of relations or two dimensional models.

5. **What is SQL?**

Ans: - SQL is structured Query language that provides interface to oracle database. SQL is the entrance to database. It is 4-GL.

6. **What is PL?**

Ans: - It is procedural language and extension to SQL provided by oracle. It is 3GL language. PL is having the features that SQL does not have like-loop through records, manipulating then one at a time. Keeps code secure and storing permanently on the server rather than client. Handle exceptions. Works with variable, parameters, collections, records, arrays, objects, cursors, exceptions etc.

7. **What are PL/SQL Data-types?**

Ans: -

- **SCALER** - `CHARACTER, NUMBER, BOOLEAN (True, False, Null), DATE`
- **CHARACTER/STRING** - `CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, LONG, RAW, LONG RAW`
- **NUMBER TYPES** - `NUMBER, BINARY_INTEGER, PLS_INTRGER`.
- **REFERENCES** - `REF CURSOR, REF`
- **COMPOSITE** - `RECORDS, NESTED TABLES, INDEX-BY-TABLES, VARRAYS`.
- **LOB** - Large Objects Data-types to work with Binary and Character data upto 4 GB. (10G 8-128 GB).

8. VARIABLE Are memory region used in a PL/SQL block to hold data, they are defined in `DECLARATION` Section of block.

9. %TYPE – The `%TYPE` attribute provides the datatype of a variable, constant, or column. This attribute is particularly useful when declaring a variable or procedure argument that refers to a column in a database table.

10. %ROWTYPE - The `%ROWTYPE` attribute is useful if you want to declare a variable to be a record that has the same structure as a row in a table or view, or a row that is returned by a fetch from a cursor.

11. BULK BINDING, FORALL & BULK COLLECT clause.

BULK BINDING - PL/SQL engine executes the procedural statement but sends SQL statements to the Database engine for execution. D.B. engine executes these SQL statements and in some cases returns data to the Pl/Sql engine. Each context switch between Pl/Sql engine and D.B. engine adds to the overhead. So if many switches are required performance suffers. That can be happens when SQL statement executes inside loop using collection elements as bind variables. In these cases bulk binding can improve performance.

Binding of entire collections at once is called bulk binding.
They improve the performance by minimizing the numbers of context switches between pl/sql engine and D.B. engines. With bulk bind entire collections are passed back and forth and not the individual element.

BULK COLLECT allows you to retrieve sets of records that you can store in nested table or index by tables.

FOR ALL enables you to send DML statements in batches. FORALL can insert, update, and delete data. These methods reduce the context switching between PL/SQL and SQL engines.

12. USE OF FORALL & BULK COLLECT

Program 1 >

```
DECLARE
TYPE NUM_TAB IS TABLE OF NUMBER;
TYPE NAME_TAB IS TABLE OF VARCHAR2(25);
V_EMPNOS      NUM_TAB;
V_ENAMES      NAME_TAB;
V_SALS        NUM_TAB;
BEGIN
        SELECT EMPNO, ENAME, SAL
        BULK COLLECT INTO V_EMPNOS,V_ENAMES,V_SALS
        FROM MYEMP;
FOR I IN 1..V_EMPNOS.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE(V_EMPNOS(I)||’  ‘||V_ENAMES(I)||’  ‘||V_SALS(I));
END LOOP;
END;
```

Program 2 >

```
DECLARE
BEGIN
TYPE NUM_TAB IS TABLE OF NUMBER;
FORALL  I IN 1..V_DEPTNOS.COUNT
TYPE NAME_TAB IS TABLE OF VARCHAR2(25);
        DELETE FROM MYEMP WHERE DEPTNO=V_DEPTNOS(I);
V_EMPNOS     NAME_TAB;
        SELECT EMPNO, ENAME, SAL BULK COLLECT INTO
        V_ENAMES, NAME_TAB, V_EMPNOS, V_ENAMES, V_SALS FROM MYEMP;
V_SALS            NUM_TAB;
END;
V_DEPTNOS NUM_TAB(40,20,30,10);
```

13. DATABASE STRUCTURE

Logical Structure	*Physical Structure*
Tablespaces	One or more Datafiles
Schemas objects	Two or more Redolog files-one for reading, one for writing.
Segments – collection of extents.	One or more control files.
Extents – Collection of Data blocks.	
Datablocks.	

PHYSICAL STRUCTURE

14] What is DATAFILES - Entire data of database is stored in the Datafiles. One or more datafiles forms a logical unit called as tablespace. One tablespace can have multiple datafiles but one datafiles can be mapped to only one tablespace. Size of datafile can be changed once they are created. Whenever new data is inserted or existing data is modified oracle first stores them into D.B. buffer cache and then it is transferred to Datafiles.

Whenever data that required to read is not available in D.B. buffer cache oracle loads it from datafiles.

15] REDOLOG FILES - Stores the log of changes made to the data of the database. Every D.B. has two or more Redo log files. Collection of Redolog files is called Redolog.

16] What is Multiplexing(mirroring) One or more copies of same redolog files can be maintained across different drives is called multiplexing. Information stored in redolog file is used to perform recovery operation.

Process of applying redolog files during recovery operation is called as **Rolling forward.**

17] CONTROL FILES

Stores name and timestamp of D.B. creation. Control file can be multiplexed.

Information stores in control file is read by oracle whenever oracle D.B. is getting mounted. Oracle maintains information stored in control file and update that information whenever their is change made in physical structure of oracle D.B.

LOGICAL STRUCTURE

18] DATA BLOCK

It is smallest unit of I/O operation used by oracle to read or write data on hard disk. Size of data block is determined by initialization parameter DB_BLOCK_SIZE.

There are five parts of datablock

Header – Type of segment to which datablock is mapped.

Table Directory – Information about table index having data in it.

Row – Information about row such as rowids.

Row data – Actual data.

Free space – pctfree- % of free space used for future updates.

19] EXTENTS – It is collection of contiguous data blocks. (contiguous-not in sequential order)

Oracle manages size of segment using extents.

20] SEGMENT – segment is the collection of multiple extents. There are four types of segments.

1) Data Segment – It is allocated for non-clustered table, cluster, materialized views, snapshots, snapshot logs, every partition of partitioned table.

2) Index segment – Allocated for every index that is created.

3) Rollback segment – It is used to maintain the read consistency. The data that is manipulated or deleted which is not committed is stored in the rollback segment. The session which are reading the data from the rollback segment.

4) TEMPERARY SEGMANT – They are used to store the intermediate results of SQL statement. Typically they are used for sorting or while processing GROUP BY, HAVING, UNION, UNION ALL, MINUS, INTERSECT, operators. Oracle allocates & de-allocates them automatically implicitly.

21] TABLESPACES – Tablespace is a collection of schema objects. One tablespace can contains objects of multiple schemas. Objects of one schema can span across multiple tablespaces. By default there is always one tablespace by the name system. System tablespace contains data dictionary objects.

22] ORACLE INSTANCE – Collection of SGA & background processes is called Instance. Oracle instance is mounted before database mounted. Mounting of oracle instance space for SGA is allocated & background processes starts functioning. There are around ten background processes out of that four are mandatory. Those are DBWR, LGWR, PMON and SMON.

23] MEMORY STRUCTURE –

1) SGA (Shared/System Global Area) – D.B. Cache, Redolog Buffer, Shared pool
2) PGA (Program Global Area)
3) Sort Area

Oracle allocates space for SGA when oracle instances is getting mounted. SGA stores information that is shared across the connected sessions. It stores the information and data such as shared SQL area, Data dictionary Cache, PL/SQL library cache. Log of changes made to the data of database as well as new or modified data.

SGA is divided into three parts. 1) DATABASE BUFFER CACHE 2) REDOLOG BUFFER 3) SHARED POOL

24] DATABASE BUFFER CACHE

It is collection of database buffers. Size of one Database buffer is equal to size of one datablock.

DB_BLOCK_BUFFERS parameter specifies the size of database buffer cache in terms of total nos of buffers in the cache.

Database Buffer in the cache are organized in two lists i,e Dirty list and LRU (least Recently Used)

Dirty list contains Dirty Buffers(Modified Buffer) and LRU contains free buffers, pinned buffer(currently being in access)

25] What is CACHEHIT – Accessing the data that is in the database buffer cache is called as cachehit.

26] What is CACHEMISS – When required data is not their in the database buffer cache, loading it from datafile to database buffer cache is called as cachemiss.

27] REDOLOG BUFFER – It contains a log of changes made to the data of database by each transaction. Size of Redolog buffer is determined by LOG_BUFFER parameter. This information is transferred to Redolog files as transaction commits.

28] SHARED POOL – Information stored in shared pool is divided into three parts data dictionary cache, PL/SQL library cache, and shared SQL Area.

1) DATA DICTIONARY CACHE – Data dictionary information that is required to varify the transaction, execution is loaded into the data-dictionary cache.

2) PL/SQL LIBRARY CACHE - Object code copy of the subprogram is loaded into the PL/SQL library cache before execution happens.

3)SHARED SQL AREA - Execution plan generated by oracle optimizer is stored into the shared SQL area. The same plan is shared across all the connected sessions. The size of shared pool is determined by initialization parameter SHARED_POOL_SIZE.

29] PGA (Private/program global Area)

It stores the information that is private to specific session such as subprogram or packages, variables, private SQL area. The space for PGA is allocated at server side, when session is established. If there is no space at server side to allocate the memory for PGA, oracle will give an error and session will not be established.

30] SORT AREA – It is typically used for sorting when ORDER BY, GROUP BY clause is used.

31] PROCESSES - 1) USER PROCESSES

2) ORACLE PROCESSES – I) SERVER PROCESSES II) BACKGROUNG PROCESSES.

32] USER PROCESSES – User processes started at client side, it is created at client side when connection request is made to the server. All the request made by the application to the database server are routed through the user process.

33] ORACLE PROCESSES – There are two types of oracle processes

1) SERVER PROCESS – these processes are started at the server side to carry out the request made by user processes. In dedicated server configuration for every user process their will be one server process started. Multithreaded server configuration allows to share some set of server processes across connected user processes.

2) BACKGROUND PROCESSES – Their are around 10 background processes that are used to carryout request. They provide increased parallelism and better performance. DBWR, LGWR, PMON, SMON.

DBWR – (Database Writer) - DBWR Background process writes dirty buffers to datafiles. The following are three situations when DBWR process writes dirty buffers to the datafiles.

When server process doesn't find enough clean reusable buffer it instructs DBWR to write.

When timeout occurs (every 3 seconds).

When checkpoint occurs it writes all dirty buffers to the datafiles.

LGWR – (Log writer) - LGWR writes redolog entries (that are stored into the redolog buffer) to the redolog files. In absence of CKPT it does the work of CKPT.

CKPT (Checkpoint) – Checkpoint is the time when oracle updates the datafiles and control files, when checkpoint occurs check point instruct DBWR to writes the dirty buffer to the datafiles.

PMON (Process Monitor) - It checks for the failed server process and restart them.

It free up the cache and resources that were used by the failed user process.

SMON (System Monitor) – Primary task of SMON is to perform the instance recovery. If you have oracle parallel server configuration then SMON of one instance can perform the recovery of another failed instance.

It deallocates the temporary segment when they are no longer in use.

It defragments the datablocks so that allocation of them for extent can be continuous.

ARCH (Archiever) - It copies Redolog files to the offline storage device. It is available only when automatic archieving is enable and redolog is used in archilog mode.

RECO – (Recover) - It is available only in distributed database system. It is used to resolve in-doubt distributed transaction that is pending due to network failure. It is available only when distributer_transaction parameter value is set to greater than zero.

SNPn (snapshot refresh process)

Their can be around 36 snapshots refresh processes. They are used to refresh the snapshots the is scheduled to be refreshed.

34] Locks: Locks are mechanisms intended to prevent destructive interaction between users accessing ORACLE data. ORACLE uses locks to control concurrent access to data.

35] What are uses of locks

Locks are used to achieve two important database goals :

Consistency: Ensures that the data a user is viewing or changing is not changed by other users until the user is finished with the data.

Integrity : Ensures that the database's data and structures reflect all changes made to them in the correct sequence.

<u>36] Types of Locks</u> :

1.Data Locks (DML)
2. Dictionary Locks (DDL)
3.Internal Locks and Latches
4.Distributed Locks
5.Parallel Cache Management Locks

37] Exclusive Locks:

An exclusive lock prohibits the sharing of the associated resource. The first transaction to exclusively lock a resource is the only transaction that can alter the resource until the exclusive lock is released.

38] Share Locks:

A share lock allows the associated resource to be shared, depending on the operations involved (e.g., several users can read the same data at the same time). Several transactions can acquire share locks on the same resource. Share locks allow a higher degree of data concurrency than exclusive locks.

39] Deadlocks:

A deadlock is a situation that can occur in multi-user systems that causes some number of transactions to be unable to continue work. A deadlock can occur when two or more users are waiting for data locked by each other. It typically happens when each of two or more users are waiting to access a resource that another user has already locked. This creates a deadlock situation because each user is waiting for resources held by the other user.

40] Lock Escalation: (ORACLE never escalates locks)

In some database systems, lock escalation occurs when numerous locks are held at one level and the database automatically changes the locks to different locks at a higher level. For example, if a single user locks many rows in a table, the database might automatically escalate the user's row locks to a single table lock. With this plan, the number of locks has been reduced, but the restrictiveness of what is being locked has increased.

Lock escalation greatly increases the likelihood of deadlocks.

41] What is schema?

A schema is collection of database objects of a user.

42] What are Schema Objects?

Schema objects are the logical structures that directly refer to the database's data. Schema objects include tables, views, sequences, synonyms, indexes, clusters, database triggers, procedures, functions packages and database links.

43] Can objects of the same schema reside in different tablespaces? Yes.

44] Can a tablespace hold objects from different schemes? Yes.

45] What is Oracle table?

A table is the basic unit of data storage in an Oracle database. The tables of a database hold all of the user accessible data. Table data is stored in rows and columns.

46] What is an Oracle view?

A view is a virtual table. Every view has a query attached to it. (The query is a SELECT statement that identifies the columns and rows of the table(s) the view uses.)

WITH CHECK OPTION specifies that only rows accessible to the view can be inserted or updated

WITH READ ONLY ensures that no DML operations performed through view.

47] Do a view contain data? Views do not contain or store data.

48] Can a view based on another view? Yes.

49] What are the advantages of views?

- Provide an additional level of table security, by restricting access to a predetermined set of rows and columns of a table.
- Simplify commands for the user.
- Store complex queries.
- To protect some of the columns of a table from other users.
- To hide complexity of calculations.

50] What is a synonym?

A synonym is an alias for a table, view, sequence or program unit.

51] What are the types of synonyms?

There are two types of synonyms private and public.

52] What is a private synonym? Only its owner can access a private synonym.

53] What is a public synonym? Any database user can access a public synonym.

54] What are synonyms used for?

- Mask the real name and owner of an object.
- Provide public access to an object
- Provide location transparency for tables, views or program units of a remote database.

- Simplify the SQL statements for database users.

55] What is an Oracle index? An index is an optional structure associated with a table to have direct access to rows, which can be created to increase the performance of data retrieval. Index can be created on one or more columns of a table. If you do not have index then full table scan occurs. When you drop the table corrosponding indexes also dropped.

56] What are types of indexes?

1) B-Tree OR Normal Index 2) Bitmap Index

57] How indexes are created?

Automatically – A Unique index is created automatically when you define primary or unique key on table.

Mannually – User can sreate nonunique index on column to speed up the access.

Oracle recommands to apply unique constraint instead of creating unique index because when you apply constraint it is stored within the data structure into the data dictionary and becomes the part of definition.

Syntax – CREATE INDEX index_name ON table_name (column_name); FOR B-tree Index

Syntax – CREATE BITMAP INDEX index_name ON table_name (column_name); FOR Bitmap Index

58] When to create Index?

A column contains wide range of values.

A column contains large numbers of Null vaues

One or more columns are frequently used together in WHERE clause or JOIN condition.

A table is large and most of queries are expected to retrieve less than 4 % of total rows.

59] When not to create Index?

Table is small

Columns are not often used as a condition in query.

Most of queries are expected to retrieve more than 4 % of total rows.

The table is updated frequently.

60] How can i know on which columns indexes are created?

USER_INDEXES data dictionary view contains name od indexes and its uniqueness.

USER_IND_COLUMNS view contains index_name, table_name and column_name.

61] Can you modify Index?

You cannot modify index, To change an index you must drop and recreate it.

NOTE – If you drop the table Index and Constraints automatically dropped but view and sequences remains.

62] How are the index updates?

Indexes are automatically maintained and used by Oracle. Changes to table data are automatically incorporated into all relevant indexes.

63] Can you create index on view –Ans No

64] what is bitmap index?

When you create bitmap index oracle stores 1 or 0 for every record. Index key values are stored only once. The bit that is ON is mapped to corrosponding rowid. While retrieving records oracle converts that bit to corrosponding rowid then fetches the record.

65] When to go for bitmap index?

column that has large numbers of duplicate values.

Column that are infrequently updated. Bitmap index is best for read only system.

66] What is difference between B-tree Index & Bitmap Index?

1 update on key is relatively inexpensive.

1 update on key is relatively expensive.

2 inefficient for query using AND / OR predicates.

2 efficient for query using AND / OR predicates.

3 More storage space required.

3 Less storage space required.
4 Useful for OLTP.
4 Useful for DSS.

67] What are compressed Indexes?

CREATE INDEX ind_ename ON emp(ename,empno) COMPRESS;
ALTER INDEX ind_ename REBUILD COMPRESS;
Specify compress to enable key compression which eliminates repeated occurence of key column values and may substancially reduce storage space.

68] What is Reverse Key Index?

Creating Reverse key index reverses bytes for each column key value, keeping column order in case of composite key.

When to create – for ever increasing key such as sequencial number for employee.

Disadvantage – When statement specifies ranges it takes full table scan because index is not usable for range scan.

69] What are clusters?

Clusters are groups of one or more tables physically stores together to share common columns and are often used together. Cluster is a schema objects that allows to group the tables inside the cluster. The table which are mapped to cluster will share the same segment that was allocates during creation of cluster. While creating cluster you need to mention the cluster key. Cluster key is the column that is common among the tables mapped to the cluster.

70] Why clusters are created?

It helps to save diskspace.

Retrieve the data using joins will be faster.

71] What is cluster key?

The related columns of the tables in a cluster are called the cluster key.

72] What is index cluster? A cluster with an index on the cluster key.

73] What is hash cluster?

A row is stored in a hash cluster based on the result of applying a hash function to the row's cluster key value. All rows with the same hash key value are stores together on disk.

74] When can hash cluster used?

Hash clusters are better choice when a table is often queried with equality queries. For such queries the specified cluster key value is hashed. The resulting hash key value points directly to the area on disk that stores the specified rows.

75] What is database link?

A database link is a named object that describes a "path" from one database to another.

76] What are the types of database links?

Private database link, public database link & network database link.

77] What is private database link?

Private database link is created on behalf of a specific user. A private database link can be used only when the owner of the link specifies a global object name in a SQL statement or in the definition of the owner's views or procedures.

78] What is public database link?

Public database link is created for the special user group PUBLIC. A public database link can be used when any user in the associated database specifies a global object name in a SQL statement or object definition.

79] What is network database link?

Network database link is created and managed by a network domain service. A network database link can be used when any user of any database in the network specifies a global object name in a SQL statement or object definition.

80] What is difference between TRUNCATE & DELETE?
TRUNCATE commits after deleting entire table i.e., cannot be rolled back.
Database triggers do not fire on TRUNCATE
DELETE allows the filtered deletion. Deleted records can be rolled back or committed.
Database triggers fire on DELETE.

81] What is a Cartesian product?
Expected answer: A Cartesian product is the result of an unrestricted join of two or more tables. The result set of a three table Cartesian product will have x * y * z number of rows where x, y, z correspond to the number of rows in each table involved in the join.

82] What is a join? Explain the different types of joins?
Join is a query, which retrieves related columns or rows from multiple tables.
Self Join - Joining the table with itself.
Equi Join - Joining two tables by equating two common columns.
Non-Equi Join - Joining two tables by equating two common columns.
Outer Join - Joining two tables in such a way that query can also retrieve rows that do not have corresponding join value in the other table.

83] What is Outer Joins : An outer join returns all rows that satisfy the join condition as well as those rows from one table for which no rows from the other satisfy the join condition. To write a query that performs an outer join of tables A and B and returns all rows from A, apply the outer join operator (+) to all columns of B in the join condition

84] What is left outer and right outer joins? Study the query given below
SELECT empno,ename,e.deptno,d.deptno,dname,loc from emp e, dept d WHERE e.deptno(+)=d.deptno;
SELECT empno,ename,e.deptno,d.deptno,dname,loc from emp e, dept d WHERE e.deptno=d.deptno(+);
What is the difference between these two queries and what will be the result?
First query is left outer join. And second query is right join.
Left outer join- query return allrows in table1 which is left table even if there is no match in table2.
Right outer join- query return allrows in table2 which is Right table even if there is no match in table1.

85] How can i Retrieve the defficiencies in both the tables? What is full Outer join?
SELECT e.empno, e.ename, d.deptno FROM emp e
FULL OUTER JOIN dept d
ON (e.deptno=d.deptno); (ON clause for substitution of WHERE clause. Join condition seperates from search condition)
The query returns all rows in the table 1, even if there is no match in the table2. It also retrieves all rows in table2 even if there is no match in table1.

86] What is the sub-query?
Sub-query is a SELECT statement that is embeded in clause of another sql statement called parent statement.

87] What is correlated sub-query?
Correlated sub-query is a sub-query, which has reference to the main query.

88] What is difference between sub-query & corelated subquery?
1 Inner query in nested subquery statement executes first & return result results are used in filtering condition of main query.
1 Corelated subquery executed parallel with the parant statement.
2 Subquery executes once for entire parent statement.
2 Corelated subquery executes once for every record that is getting processed in parent statement.
3 Normal subquery also known as single row subquery.
3 Corelated sy=ubquery also known as multiple row subquery.
4 e.g Retrieve employee who can earn more than martin
SELECT * FROM emp WHERE sal < (SELECT sal FROM emp WHERE ename = ‘martin,);

4 Retrieve empno, ename, sal with their dept. Avg sal & total nos. Of employee in their dept.

SELECT empno, ename, sal, avg_sal, total_emp FRPM emp A,
(SELECT deptno, AVG(sal) avg_sal, COUNT(*) total_emp GROUP BY deptno) B
WHERE A.deptno=B.deptno

89] Explain UNION, MINUS, UNION ALL and INTERSECT? I m having two tables tab1 having records 1,2,3 & tab2 having records 3,4,5 What is results for all set operators for query
SELECT * FROM tab1;
<set_operator>
SELECT * FROM tab2;

UNION - returns all distinct rows selected by either query **Results 1,2,3,4,5**

UNION ALL- returns all rows selected by either query, including all duplicates. **Result** 1,2,3,3,4,5

INTERSECT - returns all distinct rows selected by both queries. **Result** 3

MINUS - returns all distinct rows selected by the first query but not by the second. **Result** 1,2

90] What is the usage of SAVEPOINTS?

SAVEPOINTS are used to subdivide a transaction into smaller parts. It enables rolling back part of a transaction.

91] What are the data types allowed in a table?

CHAR, VARCHAR2, NUMBER, DATE, RAW, LONG and LONG RAW.

92] What is difference between CHAR and VARCHAR2? What is the maximum SIZE allowed for each type?

CHAR pads blank spaces to the maximum length.

VARCHAR2 does not pad blank spaces.

For CHAR the maximum length is 255 and 2000 for VARCHAR2.

93] How many LONG columns are allowed in a table? Is it possible to use LONG columns in WHERE clause or ORDER BY?

Only one LONG column is allowed. It is not possible to use LONG column in WHERE or ORDER BY clause.

94] What are the pre-requisites to modify datatype of a column and to add a column with NOT NULL constraint?

-To modify the datatype of a column the column must be empty or null.
-To add a column with NOT NULL constrain, the table must be empty.
-We can add column without NOT NULL constraint.
-For character column We can increase size and decrease up to max size in column.
-For numeric column only we can increase size cannot decrease.

95] What is a database link? Database link is a named path through which a remote database can be accessed.

96] What are different types of SQL functions?

1) CHAR FUNCTIONS – CONCATINATION, UPPER, LOWER, INITCAP, SUBSTR(<string>, <M>, [<N>]), LPAD, RPAD, LTRIM, RTRIM, LENGTH, REPLACE, TRANSLATE, INSTR, ASCII, CHR.

SELECT SUBSTR(‘sunil’, 1,3) FROM dual; Result **sun**

e.g If i want how many characters ‘A’ in string ‘MAHAJAN’ Then,

SELECT LENGTH(‘MAHAJAN’)-LENGTH(REPLACE(‘MAHAJAN’,’A’)) FROM dual;

First it will calculate LENGTH of ‘MAHAJAN’ that is equal to 7 and LENGTH(REPLACE) will give result as 4 hence
7-4=3 i.e ‘A’ will appears 3 times.

2) NUMERIC FUNCTIONS ROUND, TRUNC, CEIL, FLOOR, SQRT, POWER, ABS

ceil rturns nearest integer value >= n
floor returns nearest integer value <=n

3) DATE FUNCTIONS ADD_MONTHS, MONTHS_BETWEEN, SYSDATE, LAST_DAY, NEXT_DAY, NEW_TIME.

4) CONVERSION FUNCTIONS TO_CHAR, TO_DATE.

5) LIST FUNCTIONS

GREATEST, LEAST, NVL - It returns value if expression is null else it returns value of expression.

SIGN - It returns 1 if expression greater than 0 , it returns –1 if expression is negative, and returns 0 for 0.

DECODE - Function is used to manipulate or retrieve data conditionally, though you are manipulating the data conditionally you cannot use comparison or logical operator but you can use Arithmetic operator.

6) GROUP FUNCTIONS COUNT, SUM, MIN, MAX AVG.

97] Difference between SUBSTR and INSTR?

INSTR (String1, String2 (n, (m)),

INSTR returns the position of the m-th occurrence of the string 2 in string1. The search begins from nth position of string1.

SELECT INSTR(‘SUNILMAHAJAN’, ‘H’) FROM dual; Result **8**

SUBSTR (String1 n, m)

SUBSTR returns a character string of size m in string1, starting from n-th position of string1.

98] What is integrity constraints?+

Data integrity means validating the data that is stored in database. This validation of data is done as per the business rules.There are two mechanism used in oracle to maintain the data integrity.

Integrity constraint and Database triggers

Integrity constraints are NOT NULL, UNIQUE, PRIMARY KEY, CHECK, FOREIGN KEY(References), Self Referential Integrity.

99] What is referencial Integrity constraint? What is diff. Between foreign key & reference key

REFERENTIAL Integrity constraint is used to maintain the relationship between the tables of relational database. Referential integrity constraints guarantee that these relations are preserved. Table that contain parent key is known as parent table and table that contains foreign key is known as child table. Parent key must have primary key or unique constraints. You can insert null in foreign key and can have other constraint.

SELF REFERENTIAL integrity-When foreign key of table refers to parent key of same table.

Foreign key is the key i.e. attribute which refers to another table primary key.

Reference key is the primary key of table referred by another table.

100] What is ON DELETE CASCADE?

When ON DELETE CASCADE is specified Oracle maintains referential integrity by automatically removing dependent foreign key values if a referenced primary or unique key value is removed.

101] Adding constraints to existing table.

Whenever you are adding constraint to existing table, oracle checks that constraint against existing data of that column, if any of existing data is voilating the constraint that constraints will not be applied. It will give error.

ALTER TABLE statement is used to add constraint to existing table.

MODIFY clause is used to add column level constraint and ADD clause is used to add table level constraint.

ALTER TABLE emp MODIFY(sal NUMBER(5) CONSTRAINT nn_sal_emp NOT NULL,
ename VARCHAR2(25) CONSTRAINT NN_ename_emp NOT NULL); --column level.

ALTER TABLE emp ADD(CONSTRAINT pk_empno_emp PRIMARY KEY(empno),
CONSTRAINT chk_sal_emp CHECK(sal>1000)); -- table level.

102] Where the integrity constraints are stored in data dictionary?

The integrity constraints are stored in USER_CONSTRAINTS.

103] How will you activate/deactivate integrity constraints?

The integrity constraints can be enabled or disabled by ALTER TABLE ENABLE CONSTRAINT / DISABLE CONSTRAINT.

104] If unique key constraint on DATE column is created, will it validate the rows that are inserted with SYSDATE?
It won't, Because SYSDATE format contains time attached with it.

105] SQL COMMANDS

DDL(Data Definition Language Statement)
They are used to create and maintain the database objects.
CREATE, ALTER, RENAME, TRUNCATE, and DROP.

DML (Data Manipulation Language Statement)
They are used to manipulate data of database.
INSERT, UPDATE, DELETE

DQL(Data Query Language Statement) It is used to Retrieve data of database. - SELECT

DCL(Data Control Language Statement) - GRANT, REVOKE

TCL(Transaction Control Language Statement) COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION

SCL(Session Control) - ALTER SESSION

SYSTEM CONTROL STATEMENT

EMBEDED SQL STATEMENT

106] Privillege: A privilege is a right to execute a particular type of SQL statement or to access another user's object.
Types of privileges :
1. System privileges 2. Object privileges

107] System Privileges: System privileges allow users to perform a particular systemwide action, or to perform a particular action on a particular type of object. E.g. Create Tablespace, Delete the row of any table, etc.

108] Object Privileges: Object privileges allow users to perform a particular action on a specific object. E.g. delete row of specific table, etc.

109] Roles: Roles are named groups of related privileges that are granted to users or other roles.

110] Advantages of Roles :

1. reduced granting of privileges	
2. dynamic privilege management	(Changing of privileges)
3. selective availability of privileges	(Enalbling/Disabling roles)
4. application awareness	(Enalbling/Disabling of roles by application)

111] Can you increase the size of a tablespace? How? Yes, by adding datafiles to it.

112] Can you increase the size of datafiles? How? No.

113] What are the various types of RollBack Segments?
Public – Available to all instances
Private – Available to specific instance

114] Is Sysdate a system variable or a system function –Ans System function

115] Result of Greatest(1,NULL) or Least(1,NULL) -Ans NULL

116] What is the difference between a view and a synonym?
Synonym is just a second name of table used for multiple link of database. View can be created with many tables, and with virtual coulmns and with condiions. But synonym can be on view.

117] What is the difference between alias and synonym?
Alias is temporary and used with one query.
Synonym is permanent and not used as alias.

118] Assume that there are multiple databases running on one machine. How can you switch from one to another? Changing the ORACLE_SID

119] PCTFREE : PCTFREE is a percentage of the space in a data block which is reserved for future updates of the existing rows in a data block. (Default is 10%)

PCTUSED : PCTUSED is a sort of threshold percentage of the space in a data block. Once the free space in a data block reaches PCTFREE, no new rows are inserted in that block until the percentage of space used falls below PCTUSED. ORACLE tries to keep a data block at least PCTUSED full. (Default is 40%)

120] What is "snapshot too old" error? How can this be prevented or mitigated?

This is caused by large or long running transactions that have either wrapped onto their own rollback space or have had another transaction write on part of their rollback space. This can be prevented or mitigated by breaking the transaction into a set of smaller transactions or increasing the size of the rollback segments and their extents.

121] What is tkprof?

The tkprof tool is a tuning tool used to determine CPU and execution times for SQL statements.

122] What is Normalisation?

Normalisation is the process of organising the tables to remove the redundancy.There are mainly 5 Normalisation rules.

123] Can you store pictures in database? How?

Yes , in long Raw datatype.

124] What are types of triggers?

1. DML Triggers
(Before, After, Insert, update, delete, Row Level, Statement Level).
2. INSTEAD OF Triggers
Instead of triggers are written only on views.
3. SYSTEM Triggers
System Triggers are written on D.B. and DDL Statements i. e. D.B. Level & schema Level .

125] What is Mutating Table error?

This error comes when trigger or subprogram called from body of trigger tries to read or process the data that is being currently processed by oracle.

126] Trigger cascading

When statement given in body of one trigger causes another trigger to fire is called trigger cascading, oracle allows max 32 triggers to cascade at a time.

127] What are mutating triggers? A trigger giving a SELECT on the table on which the trigger is written.

128] What are constraining triggers?

A trigger giving an Insert/Update on a table having referential integrity constraint on the triggering table.

129] Which trigger is more efficient: AFTER or BEFORE?

AFTER row triggers are slightly more efficient than BEFORE row triggers. With BEFORE row triggers, affected data blocks must be read (logical read, not physical read) once for the trigger and then again for the triggering statement. Alternatively, with AFTER row triggers, the data blocks need only be read once for both the triggering statement and the trigger.

130] What are the parts of a database trigger?

The parts of a trigger are:

- A triggering event or statement
- A trigger restriction
- A trigger action

131] What are the values of :new and :old in Insert/Delete/Update Triggers?

INSERT :new = new value, :old = NULL
DELETE :new = NULL, :old = old value
UPDATE :new = new value, :old = old value

132] Can a trigger written for a view?- Ans No

133] What are uses of triggers?

- Maintaining complex integrity constraints not possible through declarative constraints.
- Auditing information in a table by recording the changes made and who made them.

- Automatically signaling other programs that action needs to take place when changes are made to table.
- Publishing information about various events in a publish-subscribe environment.

134] WHEN clause in trigger

WHEN clause is valid only for row level triggers. If present trigger body will be executed for those rows that meets WHEN clause condition.

```
e.g CREATE OR REPLACE  TRIGGER checkprice BEFORE INSERT OR UPDATE OF price ON table_name
FOR EACH ROW
WHEN(new.price>49.99) –OR IF :new.price>49.99 THEN
BEGIN
<Trigger_Body>
END;
```

135] What are Trigger predicates?

INSERTING, UPDATING, DELETING --e.g WHEN INSERTING THEN <condition>

136] INSTEAD OF TRIGGERS

Instead of triggers can be defined only on views. Instead of triggers are used in two cases.
To allow view that would otherwise not be modified to be modified.
To modify the columns of nested tables column in a view.
Modificable view is one against which you can issue DML statement
In general view is modificable if it doesnt contains any of following.
SET operator (UNION,UNION ALL,INTERSECT, MINUS)
AGGREGATE functions (SUM, AVARAGE etc.)
GROUP BY, CONNECT BY, START WITH
DISTINCT opereator
JOINS.

137] Can i commit or rollback in Triggers?

Trigger will be created but when event occurs it will give error –

```
ORA-04092: cannot COMMIT in a trigger
ORA-06512: at "SCOTT.TCL_TRIG", line 3
ORA-04088: error during execution of trigger 'SCOTT.TCL_TRIG'
```

138] Why TCL is not possible in Triggers?

Because triggers are defined for each tables, if you use commit or Rollback in triggers it will affects logical transaction processing.

139] What is Trigger Execution sequence?

1 executes Before statement Triggers
2 Loop for each records manipulated by triggering statement
  a) Executes before statement triggers
  b) Lock the records & then manipulate it.
  c) Executes after row triggers.
3 Executes after statement triggers.

140] How will you enable or disable trigger?

To Disable Enable one perticular trigger

```
ALTER TRIGGER trigger_name [DISABLE / ENABLE ];
```

To Disable / Enable all trigger on one perticular table

```
ALTER TABLE table_name ENABLE ALL TRIGGERS;
ALTER TABLE table_name DISABLE ALL TRIGGERS;
```

141] What is collection? Collections are lists, which my be ordered or unordered. Ordered lists are indexed by unique subscripts unordered lists are indexed by unique identifiers which may be numbers or string names.

142] What is difference between nested tables and Varrays?

Nested table-Items of type table are called nested table.

Varying array-Items of type varray are called varying array.

Nested table-its size is dynamic. Varray-its size is fixed.

Nested table-Elements stored in NT may not retain ordering if they are deleted in between.

Varray-it is ordered group of element of same datatype.

143] What are index-by-table or pl/sql-tables.

Index-by-tables are previously known as pl/sql tables and known as associative arrays 10 g onwards. They are known as sparsely populated arrays which means numbers does not have sequential only unique.

```
e.g TYPE sal_tab IS TABLE OF NUMBER INDEX BY BINARY_INTEGER
        v_sal  sal_tab (--v_sal=index by table sal_tab=type)
```

144] How will you enforce security using stored procedures?

Don't grant user access directly to tables within the application. Instead grant the ability to access the procedures that access the tables. When procedure executed it will execute the privilege of procedures owner. Users cannot access tables except via the procedure.

145] What is P-CODE?

When subprogram is created with CREATE OR REPLACE it is stored in data dictionary. In addition to source text subprogram is stored in compiled form known as p-code.

146]What are actual and formal parameters in procedures?

Actual parameter - Subprograms pass information using parameters. The variables or expressions referenced in the parameter list of a subprogram call are actual parameters. For example, the following procedure call lists two actual parameters named emp_num and amount:

Eg. raise_salary(emp_num, amount);

Formal parameter - The variables declared in a subprogram specification and referenced in the subprogram body are formal parameters. For example, the following procedure declares two formal parameters named emp_id and increase:

Eg. CREATE PROCEDURE raise_salary (emp_id INTEGER, increase REAL) IS current_salary REAL;

147] Advantages of procedure, function-

Security- While creating subprogram one can provide security to schema objects. Since the business logic is centralized at one place all the data manipulation methods are also stored at one place. Application need to call subprogram to manipulate data of D.B. This way objects are not directly exposed to application.

Owner can create stored procedures and functions and instead of granting privilege directly on table to other user he can grant execute privilege on subprogram, this way better control on manipulation performed on table.

Improved performance-since subprograms are stored at D.B. server side, from application only call will be made and entire execution of subprogram is done at D.B. end. Which reduces network round trips and network traffic.

Ease of maintenance if change in business logic we need to change corresponding subprogram.

148] Advantages of packages

Encapsulation-it allows grouping logically related declaration together in single unit.

Improved performance-when any of member of package is called entire package is loaded into memory so it reduces need of procedure function loaded individually.

Function Overloading: Packages allow you to overload procedures or functions. Overloading a procedure means creating multiple procedures with the same name in the same package, each taking arguments of different number or datatype.

149] Name inbuilt packages with their uses.

DBMS_OUTPUT – it is used for debugging. Limitation is 255 character per line.

DBMS_REFRESH – This package creates group of materiallised views for more constant refresh.

DBMS_SQL – This package is used for Dynamic SQL.

DBMS_PIPE – Communication between different sesions.

DBMS_JOB – Package sechedule and run job at predefined interval of time. One common use is to schedule the job to automatically execute. INIT.ORA parameter JOB_QUE_PROCESS controls nos.Of jobs that can be qued.

DBMS_JAVA – PLSQL can access java feature using DBMS_JAVA package. Use this package for loadong java stored procedures into database and granting permission on java objects to user.

DBMS_BACKUP_RESTORE – The databasecinsists of datafiles,redolog and control files. The ARCH background process writes redolog to archieve redolog for recovery, DBMS_BACKUP_RESTORE Package uses information contain in control file to create backup of redolog/archieve log and datafile ang restore them.

150] Difference between Procedure and Function

Procedure can accept more than one argument function cannot

Procedure do not have return clause. Function contains return clause.

Functions return a single variable by value whereas procedures do not return any variable by value. Rather they return multiple variables by passing variables by reference through their OUT parameter.

151] Difference between procedure and triggers

Data dictionary contains source code of triggers in USER_TRIGGERS.

Source code of procedure in USER_SOURCE view of data dictionary.

COMMIT, SAVEPOINT and ROLLBACK are not allowed for TRIGGERS.

COMMIT, SAVEPOINT and ROLLBACK are not allowed for PROCEDURES.

PROCEDURE require to call by calling program whereas TRIGGERS automatically fired when event occurs.

152] What are passing parameter modes in procedures?

IN (Default) – **The** value of actual parameter is passed into procedure when procedure is called.

Inside the procedure formal parameter acts like PL/SQL constant. It is considered Read only and cannot be changed.

When procedure finishes and control returns to calling environment the actual parameter is not changed.

OUT - Any value actual parameter has when the procedure is called, is ignored.

Inside procedure formal parameter acts like an uninitialised PL/SQL variables and thus has value of NULL. It can be Read from and written to.

When procedure finishes and control returns to calling environment the content of formal parameter are assigned to actual parameter. (This behavior can be changed by using NOCOPY)

IN OUT - This mode is combination of IN and OUT.

The value of actual parameter is passed into procedure when procedure is called.

Inside the procedure the formal paraqmeter acts like an intialised variables and can be read from and written to.

When procedure finishes and control returns to calling environment the content of formal parameter are assigned to actual parameter. (subjects to NOCOPY as for OUT).

153] Passing parameter by reference or by value

A subprogram parameter can be passed in one of two ways By Reference or By Value.

When a parameter is passed by Reference a pointer to actual parameter is passed to corrosponding formal parameter. AND

When parameter is passed by Value it is copied from actual parameter to formal parameter.

Passing by reference is generally faster because it avoids copy. This specially true for collections parameter, due to their larger sizes. By default plsql pass IN parameter by Reference and OUT, IN OUT parameter by value. To pass parameter by reference NOCOPY compiler hint is used. Syntax is

Parameter_name [mode] NOCOPY datatype
If NOCOPY is present plsql compiler will try to pass the parameter by reference rather than by value. NOCOPY canot be used in IN parameter. It will generates the compilation error because IN parameter are always passed by Reference thus NOCOPY doesnt apply.

154] What is overloading?

Overloading of subprogram means thatthere is more than one procedure or function with the same name but different parameters this is very useful feature because it allows same operation to be applied to object of different types.

155] What are restrictions for overloading?

-you canot overload two subprogram if there parameter differs only in name or mode.

```
   e.g  P1(p_parameter  IN NUMBER);
        P1(p_parameter  OUT NUMBER);
```

-You canot overload two functions that differs only in Return tpes.

```
   e.g   CREATE OR REPLACE FUNCTION ---- RETURN DATE
         CREATE OR REPLACE FUNCTION ---- RETURN NUMBER
```

-Parameter of overloaded subprogram must differ by type family.

```
   e.g    CHAR and VARCHAR2 are in same family.
           P1(p_parameter IN CHAR);
           P1(p_parameter IN VARCHAR2);
```

156] What is cursor?

Cursor is pointer or handle towards work area. This work area stores the records that are processed. It points to a memory region in the PGA called context area that holds the following
1. Rows return by query 2. Numbers of rows processed by the query 3. A pointer to parsed query in the shared pool.

157] Types of cursors

1) Implicit cursor 2)Explicit cursor

158] What is explicit cursor?

Explicit cursor provides control over cursor processing. They are meant to work on SELECT statements that return more than one records at a time. This cursors are declared in the declaration section of any block.

159] What is implicit cursor?

Implicit cursors are opened and closed automatically by oracle. Every DML SQL statement executed is provided a context area in the PGA and in turn has a cursor.
Implicit cursors are automatically and implicitly controlled.Implicit cursors are declared by PL/SQL implicitly for all DML and PL/SQL SELECT statements,including queries that return only one row.

160] Advantages of explicit cursor.

Explicit cursors are useful when we want to read or process more than one records.
Their is control over explicit cursors.

161] Cursor Attribute

%FOUND – The %FOUND attribute tests whether a fetch returns a record. The returnn value is of boolean type. if true, a row was returned by fetch, if false row was returned.
%NOTFOUND – It is reverse of %FOUND
%ROWCOUNT – This test for number of rows fetch from cursor at any given time and return number.
%ISOPEN – This attribute tests to see if cursor already open. If true cursor is open, if false cursor is not open.

162] What is parametric cursors?

You can pass parameters to the cursor in a cursor FOR loop. This means that you can open and close an explicit cursor several times in a block, returning a different active set on each occasion. For each execution, the previous cursor is closed and re-opened with a new set of parameters.Each formal parameter in the cursor declaration must have a corresponding actual parameter in the OPEN statement. Parameter data types are the same as those for scalar variables, but you do not give them sizes.The parameter names are for references in the query expression of the cursor.

```
DECLARE
CURSOR emp_cursor
(p_deptno NUMBER, p_job VARCHAR2) IS
SELECT employee_id, last_name
FROM employees
WHERE department_id = p_deptno
AND job_id = p_job;
BEGIN
OPEN emp_cursor (80, ’SA_REP’);
. . .
CLOSE emp_cursor;
OPEN emp_cursor (60, ’IT_PROG’);
. . .
END;
```

163] FOR UPDATE, WHERE CURRENT OF, WAIT NOWAIT clause.

FOR UPDATE – It is used with SELECT statement of CURSOR to lock record retrieved by SELECT statement.

CURRENT OF – can be used in WHERE clause of DML statement to process the current record of cursor. When CURRENT OF clause is used records retrieved for that cursor must be locked explicitly using FOR UPDATE clause.

NO WAIT – It will not wait to lock the record if any of the records are already locked for other session.

WAIT( ) - It will wait for specified time and if locks are not released by that session control will come out by generating error.

SELECT <column_name> FROM emp FOR UPDATE OF <column_name to lock) [NO WAIT/WAIT(60)];

164] Materialized views

It is schema objects that contain data. Oracle allocates data segment to store data of those views.

They are used in datawarehousing, mobile computing, and distributed D.B. environment.

165] What are Refresh Modes of Materiallised View?

Mannual Refresh are performed using DBMS_MVIEW package. This package provides numbers of procedures & functions to manage materiallised views including REFRESH, REFRESH_INDEPENDANT & REFRESH_All_MVIEW procedure.

Automatic Refreshing can performed.

ONCOMMIT – When ONCOMMIT option is specified for MV that view is updated whenever changes to one of base table are commited.

At specified time – Refresh of MV can be scheduled to occurs at specified time by using START WITH & NEXT clause.

166] What is blank-padded comparison semantics?

If two values have different lengths, ORACLE adds blanks to the shorter value, until the two values are the same length. Two values that differ only in the number of trailing blanks are considered equal. (ORACLE compares CHAR values using blank-padded comparison semantics)

167] What is anonymous block?

Anonymous block of code are not stored and not named. They are executed in session.

To execute same code again you must save it to os file.

168] What is Role? Roles are named groups of related privileges that are granted to users or other roles.

169] What are advantages of roles?

1. Reduced granting of privileges, dynamic privilege management.

2. selective availability of privileges.

170] Default Roles Created by Oracle :

1. Connect 2. Resource 3. DBA 4. Exp_Full_Database
5. Imp_Full_Database

171] If i give INSERT then UPATE, then INSERT, then INSERT and then COMMIT and EXIT == Will all the four statements committed? == YES All the SQL Statement will be COMMITed.

172] If i give INSERT then UPATE, then INSERT, then INSERT and then CTEATE TABLE and EXIT == What will be the action, Will all the four statements committed?if yes/no WHY?

YES The data will be saved because CREATE TABLE is DDL which is implicitly COMMITed.

173] INSERT, INSERT, UPDATE, INSERT then ROLLBACK == What will be the action?

All statements before ROLLBACK up to last COMMIT or up to last DDL statement will be ROLLBACKed.

174] INSERT, INSERT, UPDATE, INSERT then EXIT or QUIT == What will be the action? Give reason

All statements ill be COMMITed because EXIT and QUIT are implicitly COMMITed.

175] Suppose table having columns A,B,C,D and E. And column B having 10 rows out of which 7 having some values and remaining 3 are null, what will be the output of COUNT(B)? Why?

Output will be 7 because COUNT(column_name) counts only the total nos of valued rows whereas COUNT(*) counts total nos of rows in table.

176] What is snapshot?

Snapshot is the object created in local database having the data of remote database. While creating snapshot you need to mention the frequency to refresh them.

177] What are advantages of snapshots?

using snapshots you can eliminates the needs to retrieve the data from remote database. This helps to reduce network

178] Why clusters are created?

It helps to save diskspace.

Retrieve the data using joins will be faster.

179] What is Wrapper?

a standalone utility that converts PL/SQL source code into portable object code. You can use the Wrapper to deliver PL/SQL applications without exposing your source code.(Wrapped file extention is .plb)

syntax:- WRAP INAME=input_file [ONAME=output_file]

WRAP INAME=c:\temp\my_sql.sql

180] What are the advantages of Wrappers?

1. To deliver PL/SQL application without exposing PL/SQL source code.
2. Platform independence,you need not deliver multiple versions of the same compilation unit.
3. Dynamic loading-users need not shut down and relink to add a new feature.
4. Dynamic binding-external references are resolved at load time.
5. Strict dependency checking-invalidated program units are recompiled automatically.
6. Normal importing and exporting-the Import/Export utility accepts wrapped files.

181] Can we use select statement in FORALL?

No, We can use only Insert, Update or Delete statement.

The keyword FORALL instructs the PL/SQL engine to bulk-bind input collections before sending them to the SQL engine. Although the FORALL statement contains an iteration scheme, it is not a FOR loop.

Its syntax follows:

```
FORALL index IN lower_bound..upper_bound
SQL_statement;
```

The SQL statement must be an INSERT, UPDATE, or DELETE statement that references collection elements.

182] What are uses of variables? Why we use variables?

Temporary storage of data.   Manipulation of storage values.

Reusability.   Ease of maintenance.

183] How many rows will the following SQL return :

SELECT * FROM emp WHERE rownum < 10;   ----   9 rows

184] How many rows will the following SQL return :

SELECT * FROM emp WHERE rownum = 10;   --- No rows

185] Defination of relation Data. by Dr. Codd (IBM)?

A Relational Database is a database where all data visible to the user is organized strictly as tables of data values and where all database operations work on these tables.

186] Can I update through a view?

You can do this if

1.Your view is a simple subset of a single table.

2.All "NOT NULL" columns for the table must be in the view.

3.The PRIMARY key is in the view.

NOTE. You cannot modify through view if it contains GROUP functions, GROUP BY clause, DINSTINCT keyword, ROWNUM keyword,and columns definrd by Expression.

187] What are INLINE Views?

Inline views are subselect in the FROM clause that acts as a view at execution time. e.g To select top 10 records.

SELECT * FROM **(SELECT * FROM emp ORDER BY empno)** WHERE rownum<=5 ORDER BY empno;

Hilighted is the inline view.

188] To know installed version of oracle? SELECT banner FROM v$version;

189] Name a tablespace automatically created when you create a database. The **SYSTEM** Tablespace.

190] Can we add column to table?

Yes ALTER TABLE emp ADD(mgr NUMBER (4));

Column added is NULL column

You canot add NOT NULL column it will give errro doing so -

ORA-1758 table must be empty to add mandatory (NOT NULL) column.

191] Can we modify datatype, size of column?

We can change datatype only if table is empty e.g we can change from number to varchar2 etc.

You canot decrease size of column if table contains records in it. Doing so will give error.

Column to modified must be empty to decrese precision or scale of number column but we can increase size anyway you can increase size of character column and can decrease up to max value in column.

For numerical column only we can increase size, canot decrease.

192] What is difference between WHERE & HAVING clause?

1.   You canot use WHERE clause to restrict the group. WHERE clause is used to restrict rows

1.   To restrict group use HAVING clause.

2.   You canot use GROUP functions in whrere clause.

2.   Use HAVING clause for this.

e.g 1) SELECT deptno, AVG(sal)

FROM emp

WHERE AVG(sal) > 4000; -----Incorrect

WHERE AVG(sal) > 4000

*****

ERROR at line 3:

ORA – 00934 group functions are not allowed here.

SELECT deptno, AVG(sal)

```
     FROM emp
     HAVING AVG(sal) > 4000;  -----Correct  - It will give desired result.
```

```
2)          SELECT deptno, sal FROM emp
WHERE sal > 4000;  -----Correct – it will give desired result.
```

```
SELECT deptno, sal FROM emp
HAVING sal > 4000;  -----Correct – it will not give desired result.
HAVING sal > 4000
          *
ERROR at line 2:
ORA – 00979 : not a group function.
```

193] If table is creates without giveng sizes as below, what will be the resrictions while inserting records.

- **CREATE TABLE data (col1 NUMBER, col2 CHAR);**
- **CREATE TABLE data (col1 NUMBER, col2 VARCHAR2);**

ANS – for NUMBER column having size not mentioned you can insert any floating (e.g 20.25) and any size --number but for char if size not given bydefault it takes CHAR(1) .

It is not possible to have VARCHAR2 without size, Table not created giving error like missing right paranthesis.

194] What is NVL, NVL2 and NULLIF?

NVL(comm,0) it will return comm if comm is not null other wise it return 0.

NVL2(expr1,expr2,expr3) it will return expr2 if expr1 is not null otherwise it will return expr3.

e.g NVL2(comm,sal+comm,sal)

NULLIF(expr1,expr2) expr1 is compared with expr2, if expr1!= expr2 then expr1 is returned. It is equivalant to CASE WHEN function. I.e CASE WHEN expr1 THEN NULL ELSE expr1 END.

195] What is COALCASE, CASE & DECODE?

COALCASE(expr1,expr2, ----- exprn)

It returns expressions if preceeding expressions are null. If expr1 is null it will return expr2, If expr2 is null it will return expr3 ---etc.

CASE expression - It fascilate conditional enquiries by doing work of IF-THEN-ELSE statement.

```
e.g  SELECT last_name, job_id,sal
CASE  job_id  WHEN ‘IT_PROG’ THEN 1.10 * sal
              WHEN ‘CLERK’ THEN  1.15 * sal
               ELSE  sal  END ‘REVISED SALARY’  FROM emp;
```

DECODE - It is LIST function.

```
SELECT last_name, job_id,sal
DECODE (job_id,  ‘IT_PROG’,  1.10 * sal.
           ST_CLERK’,  1.15 * sal, sal)  REVISED SALARY FROM emp;
```

196] Suppose i have a column having values like A0010,B0001,C0050 Then how can i order the data in following manner.

B0001
A0010
C0050

SELECT srno FROM swap ORDER BY substr(SRNO,-2,2);

197] If my column has null values in it and column data type is number how can i write a query giving me 'NA' which is char in palce of those null values. SELECT NVL(TO_CHAR(comm),'NA') from emp;

198] What is difference between NUMBER, BINARY_INTEGER & PLS_INTEGER?

NUMBER having precision *p* and scale *s*. The precision *p* can range from 1 to 38. The scale *s* can range from

84 to 127.

BINARY_INTEGER Base type for integers between -2,147,483,647 and 2,147,483,647.

PLS_INTEGER Base type for signed integers between -2,147,483,647 and 2,147,483,647.

PLS_INTEGER values require less storage and are faster than NUMBER and BINARY_INTEGER values.

199] What is exception & Different typed of exceptions

Exceptions are essentially interrupt that can halt processing when an error/exception condition is raised.

Predefined exception

Userdefined exceptions

PREDEFINED EXCEPTIONS –

DUP_VAL_ON_ONDEX – ORA-0001 - unique constraint violated.

NO_DATA_FOUND ORA-1403 Single row SELECT returned no data.

TOO_MANY_ROWS ORA-01422 Single-row SELECT returned more than one row.

INVALID_NUMBER ORA-01722 Conversion of character string to number fails e.g 1A.

INVALID_CURSOR ORA – 01001 Illegal cursor operation occurred.

CURSOR_ALREADY_OPEN ORA-06511 Attempted to open an already open cursor.

ZERO_DEVIDE ORA-1476 Attempted to divide by zero.

200] What are SQLCODE and SQLERRM and why are they important for PL/SQL developers?

SQLCODE returns the value of the error number for the last error encountered. The SQLERRM returns the actual error message for the last error encountered. They can be used in exception handling to report, or, store in an error log table, the error that occurred in the code. These are especially useful for the WHEN OTHERS exception.

201] What is Propagating an Exception

If the exception is raised in the executable section of the block and there is no corresponding exception handler, the PL/SQL block terminates with failure and the exception is propagated to the calling environment.

202] What is PRAGMA_EXCEPTION_INIT?

Associate the declared exception with the standard Oracle server error number using the PRAGMA EXCEPTION_INIT statement.

Syntax

PRAGMA EXCEPTION_INIT(exception, error_number);

where: *exception* is the previously declared exception.

error_number is a standard Oracle Server error number.

203] What is OTHERS handler?

pl/sql defines a special exception handler, known as OTHERS. This handler will execute for all raised exceptoins that are not handled by any other when clauses defined in current exception section. It should be always the last handler in the block, so that all previous handlers will be scanned first. WHEN OTHERS will trap all exceptions, be they user defined or predefined.

204] What is RAISE_APPLICATION_ERROR.

Use the RAISE_APPLICATION_ERROR function to communicate a predefined exception interactively by returning a nonstandard error code and error message. With RAISE_APPLICATION_ERROR, you can report errors to your application and avoid returning unhandled exceptions.

In the syntax:

error_number is a user-specified number for the exception between –20000 and –20999.

message is the user-specified message for the exception. It is a character
string up to 2,048 bytes long.

TRUE | FALSE is an optional Boolean parameter (If TRUE, the error is placed on
the stack of previous errors. If FALSE, the default, the error replaces all previous errors.)

RAISE_APPLICATION_ERROR can be used in either (or both) the executable section and the exception section of a PL/SQL program. The returned error is consistent with how the Oracle server produces predefined, nonpredefined, or user-defined error. The error number and message is displayed to the user.

RAISE_APPLICATION_ERROR IN Executable section:

```
BEGIN
...
DELETE FROM employees
WHERE manager_id = v_mgr;
IF SQL%NOTFOUND THEN
RAISE_APPLICATION_ERROR(-20202, 'This is not a valid manager');
END IF;
...
```

EXCEPTION handling section

```
...
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR (-20201,
'Manager is not a valid employee.');
END;
```

205] What is difference between RAISE & RAISE_APPLICATION_ERROR?

1 RAISE is the statement & when error associated with exception occurs, the exception is raised. Userdefined exceptions are raised explicitly via raise statement and predefined exceptions are raised implicitly.

1 RAISE_APPLICATION_ERROR is the built in function of package standard which is used to create your own error message text.

2 Raise – does not allow for message text.

2 RAE – allow to supply your own error message text which can contains application specific data.

3 In general use RAE for errors that are designed to seen by end users. Specific error number and descriptive text are useful here.

3 Raise on the other is useful for error that are designed to be handled programatically.

206] What are packages?

A package usually has a specification and a body, stored separately in the database.

The specification is the interface to your applications. It declares the types, variables, constants, exceptions, cursors, and subprograms available for use. The package specification may also include PRAGRMAs, which are directives to the compiler.

The package itself cannot be called, parameterized, or nested. Still, the format of a package is similar to

that of a subprogram. Once written and compiled, the contents can be shared by many applications.

Cannot be invoked, parameterized, or nested Allow the Oracle server to read multiple objectsinto memory at once.

A package specification can exist without a package body, but a package body cannot exist
without a package specification. Define the package specification before the body.

207] What is forward declaration?

PL/SQL does not allow forward references. You must declare an identifier before using it. Therefore, a subprogram must be declared before calling it.

208] What are steps to Process SQL Statements?

All SQL statements have to go through various stages. Some stages may be skipped.

Parse

Every SQL statement must be parsed. Parsing the statement includes checking the statement’s syntax and validating the statement, ensuring that all references to objects are correct, and ensuring that the relevant privileges to those objects exist.

Bind

After parsing, the Oracle server knows the meaning of the Oracle statement but still may not have enough information to execute the statement. The Oracle server may need values for any bind variable in the statement. The process of obtaining these values is called binding variables.

Execute

At this point, the Oracle server has all necessary information and resources, and the statement is executed.

Fetch

In the fetch stage, rows are selected and ordered (if requested by the query), and each successive fetch retrieves another row of the result, until the last row has been fetched. You can fetch queries, but not the DML statements.

209] What is SEQUENCE?

Sequence is user created database object that can be shared by multiple user to generate unique integer. Sequence generates serial list of unique number for column of table.

210] Syntax for sequence/

```
  CREATE SEQUENCE  S1
  [START WITH n]
  [INCREMENT BY n]
  [MAXVALUE n | NOMAXVALUE]
  [MINVALUE n | NOMINVALUE]
  [CYCLE | NOCYCLE]
  [CACHE n | NOCACHE];
```

MAXVALUE – Specifies maximum value sequence can generates.

NOMAXVALUE – Specifies MAX value of 10^{27} for ascending sequence and –1 for descending sequence(default).

MINVALUE – n specifies minimum sequence value.

MOMINVALUE – Specifies minimum value of 1 for ascending sequence and –(10^{27}) for descending. (Default).

CACHE n / NOCACHE –

Specifies how many values oracle preallocates and keeps in memory (Bydefault oracle server caches 20 values).

CYCLE specifies that the sequence continues to generate values after reaching either maximum or minimum value. After pan-ascending sequence reaches its maximum value, it generates its minimum value. After a descending sequence reaches its minimum, it generates its maximum.

NO CYCLE specifies that the sequence cannot generate more values after reaching its maximum or minimum value.

211] How to access the current value and next value from a sequence? Is it possible to access the current value in a session before accessing next value?

Sequence name CURRVAL, sequence name NEXTVAL. It is not possible. Only if you access next value in the session, current value can be accessed.

212] What are the restrictions for using NEXTVAL & CURRVAL.

You can use NEXTVAL & CURRVAL in the following...

SELECT list of select statement which not the part of subquery. SELECT list of subquery in INSERT.

The SET clause of UPDATE.

You cannot use NEXTVAL & CURRVAL in following.

SELECT list of view. SELECT statement with DISTINCT keyword.

SELECT statement with GROUP BY, HAVING, ORDER BY.

Subquery in SELECT, DELETE, or UPDATE statement.

213] Can i modify sequence?

Yes, by using ALTER SEQUENCE. but START WITH option cannot be changed using alter sequence. The sequence must be dropped and recreated in order to restart sequence number at different number.

Some validation is performed e.g new maxvalue cannot be less than current sequence number.

214] I have created sequence i.e
CREATE SEQUENCE S1
START WITH 10
INCREMENT BY 1 now my curent sequence number is 100 can i restart sequence at 51.

Ans – no, start with option cannot be changed. See above ans.

215] What is NULL? NULL is a value that is unavailable, unassigned, unknown, & inapplicable.

216] What following query will return?

```
1) SELECT empno, ename FROM emp
   WHERE sal = (SELECT MIN(sal) FROM emp GROUP BY deptno);
   ERROR ORA-01427 single row subquery returns more than one row.
```

217] If i have two tables

tab1		tab2	
A (Number)	B(Number)	A (Number)	B(Number)
P.K..	F.K.	F.K.	P.K.

Now i want to insert row in both the column of tab1 or tab2, is it possible? why?

ANS – You cannot insert row in both the columns of tab1 or tab2 because it will give error of constraint voilation. Foreign key value must match existing value in the parent table or be null.

218] What are notations used in subprograms?

Positional notation – the actual argument are associated with the formal argument by position.

```
Procedure --                                   calling program--
e.g CREATE OR REPLACE PROCEDURE  P1            DECLARE
   ( p_parameterA   VARCHAR2,                  v_variable1     VARCHAR2(10);
     p_parameterB   NUMBER,                    v_variable2     NUMBER(7,2);
     p_parameterC   BOOLEAN,                   v_variable3     BOOLEAN;
     p_parameterD   DATE)  AS                  v_variable4     DATE;
   BEGIN                                       BEGIN
   ........                                    P1(v_variable1, v_variable2, v_variable3, v_variable4)
   END                                         END;
```

Named notation – The formal parameter and actual parameter are both included for each argument. This allows us to rearrange the order of arguments if desired.

e.g calling program for above procedure by named notation.

```
DECLARE
......
BEGIN
P1( p_parameterB => v-variable2,
    p_parameterC => v_variable3,
    p_parameterD => v_variable4,
    p_parameterA => v_variable1);
END;
```

Mixed Notation – Positional & mixed notaions can be mixed in the same call as well if desired. The first argument must be specified by position and the remaining arguments can be specified by name.

```
DECLARE
......
......
BEGIN
P1( v_variable1,v-variable2,
  p_parameterC => v_variable3,
 p_parameterD => v_variable4);
 END;
```

219] What is difference between Positional and Named notation?

1) The actual arguments are associated with formal parameters by position.

1) The formal & actual parameters are both included for each arguments.

2) As arguments are associate with position it does not illustrates the association between actual and formal parameters.

2) Clearly illustrates association between actual and formal parameters.

3) all calls to procedure using positional notation must be changed if order of formal parameter changed.

3) Order used for formal and actual parameters is indepedant, one can changed without modifying other.

4) Names used for actual and formal are indepedant one can changed without modifying other.

4) All calls to procedure using named notations must be changed if name of formal parameter is changed.

220] What is Rowid & Rownum?

Rowid is a system generated 18 character long unique identifier that is created for every recoreds in the database. This binary value is the address or location of data in the system.

Rowid can be physical as in case with records in standard database table.

Rowid can be logical as well, as is the case with rows in indexed organised table.

AAAH14AABAAAO+HAAA

First 6 characters AAAH14 is Database segment.

Next 3 characters AAB is Datafile no.

Next 6 characters AAAO+H is Datablock no.

Last 3 characters AAA is Block Row no.

Rownum It is the pseudocolumn which returns row number of record. This is logical number determined at time query is run. As such delete or insert can cause differrent rownum assignment. Rownum do not stick to perticular record so never rely on them.

One common use of Rownum is the Numbers of records returned.

221] What are Optimizer Modes?

Rule based optimization

In this mode server process chooses its access path to the data by examining query. This optimizer has complete sets of rules for ranking access path. It is syntax driven in that it uses statement syntax in combination with data dictionary information about data structure to determine which execution to use.

ALTER SESSION Set optimiser_goal = rule;

Cost based Optimization

In this mode optimizer examine each statement and identifies all possible access path to data it then calculate resource cost of each access path and chooses least expensive one.

It is statistic driven in that it uses statistic generated for object involved in sql statement to determine most effective execution plan. Cost based optimizaiton is used if any object in sql statement has had statistic generated for it.

222] What are hints? What are types of hints?

Hints are suggestions that you give the optimizer for optimizing a SQL statement. Hints allow you to make decisions usually made by the optimizer.

ALL_ROWS : The ALL_ROWS hint explicitly chooses the cost-based approach to optimize a statement block with a goal of best throughput.

FIRST_ROWS : The FIRST_ROWS hint explicitly chooses the cost-based approach to optimize a statement block with a goal of best response time.

FULL : The FULL hint explicitly chooses a full table scan for the specified table.

ROWID : The ROWID hint explicitly chooses a table scan by ROWID for the specified table.

CLUSTER : The CLUSTER hint explicitly chooses a cluster scan to access the specified table.
HASH : The HASH hint explicitly chooses a hash scan to access the specified table.
INDEX : The INDEX hint explicitly chooses an index scan for the specified table.
INDEX_ASC : The INDEX_ASC hint explicitly chooses an index scan for the specified table. If the statement uses an index range scan, ORACLE scans the index entries in ascending order of their indexed values.
INDEX_DESC : The INDEX_DESC hint explicitly chooses an index scan for the specified table. If the statement uses an index range scan, ORACLE scans the index entries in descending order of their indexed values.
ORDERED : The ORDERED hint causes ORACLE to join tables in the order in which they appear in the FROM clause.
USE_NL : The USE_NL hint causes ORACLE to join each specified table to another row source with a nested loops join using the specified table as the inner table.
USE_MERGE : The USE_MERGE hint causes ORACLE to join each specified table with another row source with a sort-merge join.

223] How will you enforce using Index?

In order to force optimizer to use index it is put in the hint.
SELECT / * + INDEX (ind_ename) * / empno, sal, job FROM emp WHERE ename = 'scott';

224] How can you avoid using Index?

To make index access path unavailable -
Use FULL hint to optimizer for full table scan.

225] When we executes subprogram, how it executes internally?

First time When the stored subprogram is called, the p-code is loaded from disk to shared pool. This object code copy of subprogram is loaded into the pl/sql library cache before execution happens.

226] What is pinning of shared pool?

The shared pool is the protion of SGA that contains among other things, the p-code of compiled subprogram as they run. The first time the stored subprogram is called the p-code is loaded from disk to shared pool. Once the object is no longer referenced it is free to be aged out. Objects are aged out of shared ppol using LRU algorithms.
DBMS_SHARED_POOL package allows you to pin object in the shared pool. When object is pinned, it will never be ages out until you request it. No matter how full the pool get and how often the object accessed.
Advantages -
This can improve perormance as it takes time to reload package from disk.
Pinning of object also helps to minimise fragmentation of shared pool.
DBMS_SHARED_POOL package has four procedures i.e **KEEP, UNKEEP, SIZES, ABORTED_REQUEST_THRESHOLD**
1 **KEEP** – The DBMS_SHARED_POOL.KEEP procedure is used to pin object in shared pool. Packages, procedures, triggers, sequences, object types, java objects & SQL statements can be pinned.
Syntax –
PROCEDURE KEEP (name VARCHAR2, flag CHAR DEFAULT 'P');
Where name varchar2 - name of object. This can be object name or ideb=ntifier associated with SQL atatement.
Flag char determines type of object, the value of flag has following meaning.
P – Package, function, procedure. Q – SequencesR – Trigger T – Object types JS – Java source
JC – Java class JR – Java Resource JD – Java stored data C – sql cursor.
2 **UNKEEP** – UNKEEP is only way to remove kept objects from shared pool without restarting database. Kept objects never aged out automatically.
If specified object does not already exist in shared pool an error is raised.

3 **SIZES** – sizes will echo the contents of shared pool to screen. PROCEDURE SIZES (minimise NUMBER);

4 **ABORTED_REQUEST_THRESHOLD** – when the database determo=ine sthat there is no snough memory in the shared pool to satisfy a given request, it willbegin aging objects out until there is enough memory.

PROCEDURE ABORTED_REQUEST_THRESHOLD(threshold_size NUMBER);

227] What is autonomous transaction?

Autonomous transaction is independent transaction started by another transaction. It lets you suspend main transaction, do the sequel operation, commit or rollback those operations and then resume the main operation. Autonomous transactions are started by main transaction but operate independantly of parent for transaction control.

This feature is used for logging application events. To define autonomous transaction use PRAGMA_AUTONOMOUS _TRANSACTION. The pragme (compiler hint) is placed in the declare section of block.

228] How can i mark all subprograms in package as autonomous?

To mark all subprogram in package as autonomous you can't use pragma at package level, but individual subprogram must be marked as autonomous.

229] What are isolation levels in autonomous transaction?

READ COMMITED – (Default)

Changes made in autonomous transaction are visible to main transaction when it resumes.

SERIALLIAZABLE – Changes made in autonomous transactions are not visible to main transaction. Also changes made in main transaction are not visible to autonomous transaction.

```
  CREATE OR REPLACE PROCEDURE  log
  ( p_username IN VARCHAR2(10),
    p_datetime   IN  TIMESTAMP)  IS
  PRAGMA AUTONOMOUS_TRANSACTION;
           BEGIN
           INSERT INTO log (user_name, date_time)
           VALUES(p_username, p_datetime);
  COMMIT;
END;
```

230] What will happen if i do not commit or rollback in autonomous transaction?

There will be no error while compiling the code but while running the code it will give error i.e

ERROR at line 1 :

ORA – 06519 : active autonomous transaction detected and rolled back.

231] What is table?

Table is the collection of ralational data in the form of rows and columns.

232] What is Index Organised Tables?

Index Organised Tables is like regular table with index on one or more of its columns bur instead of maintaining two separate segments for table and B-Tree index database system maintains one single B-Tree structure that contains both the primary key value and other column value for corrosponding rows. For Index organised tables Primary Key is mandatory.

233] What is External table?

An extrenal table is read only table whose metadata is stored in the database but whose data is stored outside database. This way You can use external data as a virtual table. This data can be queried and joined directly and in parallel without requiring the external data to be first loaded in the database. You can use SQL, PL/SQL, and java to query the data in external table.

The main defference between external table and regular table that externally organized tables are read only. No DML operations are possible, and no indexes can be created on them. The oracle server provides two major access drivers for external tables.

One, the Loader acess driver, or ORACLE_LOADER is used for reading of data from external files using the oracle loader technology. This access driver allows the oracle server to access the data from any data source whose format can be interpreted by the SQL*Loader utility.

The other oracle provided access driver, Import/Export access driver, or oracle_internal can be used for both the importing and exporting of data using a platform independance format.

234] What is Temporary Table? Structure of table is permanant but data is temporary. Orale allocates temperary segment to store the data of the temperary table. Temperary table holds session specific private data means no two session have access to each others data. Alter table statement can be issued only when no sessions are bound to it. Session gets bound when insert is issued and get unbound when eighther truncate table or commit, rollback is used on transaction specific tables or session is closed. Indexes can be created on these tables which also be temperary.

Locks are not required since data stored in temperary table is private to specific session.

```
CREATE GLOBAL TEMPORARY TABLE flight_schedule
( flight_no             NUMBER(4),
  destination      VARCHAR2(25),
  time             DATE)
ON COMMIT DELETE ROWS;          ----- For transaction specific tables.
ON COMMIT PRESERVE ROWS;        ----- For session specific tables.
```

Use ON COMMIT DELETE ROWS option to create transaction specific temporary table. If you create transaction specific temporary table, that table will hold data for that perticular transaction only. The data form the table will be automatically lost if the transaction is completed by commit, rollback or any DDL statement.

Use ON COMMIT PRESERVE ROWS option to create session specific temporary table. If you create session specific temporary table, the data will be there in the table as ,Long as session is not closed.

235] Multiple Insert statement

Que – if i Want to select data from one table and that data is to be inserted in two or more tables is it possible? – Yes

1 Unconditional INSERT ALL

The INSERT statement refers to unconditional as no further restriction is applied to the row retrieved by SELECT. All the rows retrieved by select statement are inserted in two tables. E.g

Select empno, hiredate, sal and mgr from emp table whose empno is greater than 200. Insert those records in sal_history and mgr_history table.

```
INSERT ALL
INTO sal_history VALUES ( emp_id, hire_date,salary)
INTO mgr_history VALUES ( emp_id, manager_id, salary)
SELECT empno  emp_id,  hiredate  hire_date,  sal  salary, mgr  manager_id  FROM emp
WHERE empno > 200;
```

2 Conditional INSERT ALL

The INSERT statement refers to conditional all insert as a further restriction is applied on rows retrieved by select statement.

```
INSERT ALL
WHEN salary > 1000 THEN
INTO sal_history VALUES (emp_id,  hire_date, salary)
WHEN manager_id > 300 THEN
INTO mgr_history VALUES (emp_id, manager_id, salary)
SELECT empno  emp_id,  hiredate  hire_date,  sal  salary, mgr  manager_id  FROM emp
WHERE empno > 200;
```

3 Conditional FIRST INSERT

This INSERT statement refers to as conditional first insert. If the first when clause evaluated to true the oracle server executes the corrosponding INTO clause and skip subsequent WHEN clause for this rows. For the rows that do not satisfy the FIRST WHEN clause condition the rest of conditions are evaluated just an conditional insert and records retrieved by select statements are inserted in that tables.

4 Pivoting INSERT

Pivoting insert is used to convert the data from nonrelational database table to more relational table. E.g

Suppose you have data like emp_id, week_id, sales_mon, sales_tue, sales_wed, sales_thu, sales_fri etc from table sales_source_data. Now you want to store records in in sales_info table in following format emp_id, week, sales.

```
INSERT ALL
INTO sales_info VALUES(emp_id, week_id,sales_mon)
INTO sales_info VALUES(emp_id, week_id,sales_tue)
INTO sales_info VALUES(emp_id, week_id,sales_wed)
INTO sales_info VALUES(emp_id, week_id,sales_thu)
INTO sales_info VALUES(emp_id, week_id,sales_fri)
SELECT emp_id, week_id, sales_mon, sales_tue, sales_wed, sales_thu, sales_fri
FROM sales_source_data;
```

236] Suppose we have function like below, examine & give the result

```
CREATE OR REPLACE FUNCTION f1
(a      IN      NUMBER(3),
 b      OUT     NUMBER(3))
RETURN NUMBER IS
BEGIN
a := a + 2;
b := a + 2;
RETURN b + 2;
END;
/
```

Function will be created with compilation error.

If you type sho error on sql+ ------

6/1 PLS-00363: expression 'a' cannot be used as an assignment target – bcoz a is IN parameter & read only.

6/1 PL/SQL: Statement ignored

237] What is SDLC?

System Devlopment Life Cycle

You can devlop database by using SDLC which contains multiple stages of devlopment. This is the systematic approach to transformation of business information requirement into an operational database.

Stratagy & analysis

Study and analyse the business requirements. Build the models of the system. Transfer the business narrative in to the graphical representation of business information and rules. Confirm and refine the model with analyst and expert.

Design

Design the database based on the model devloped in the strategy and analysis phase.

Build & Document.

Build prototype system. Write and execute commands to create the table & supporting objects for database.

Devlop user documentation, help text and operation mannual to support the use and operating of system.

Transition

Refine prototype. Move an application into production with user acceptance testing conversion of existing data. Make any modification required. Roll out system to users. Operate the production system. Monitor its perfermance and enhance & refine system.

238] What are methods used in collections?

Collection API is provided to give simplified access to collections. These methods are functions and procedures. Three, EXTEND, TRIM, AND DELETE are procedures. The rests are functions. These are

COUNT, DELETE(n), DELETE(n,m), EXISTS(n), EXTEND, EXTEND(n), EXTEND(n,i), FIRST, LAST, LIMIT, NEXT(n), PRIOR(n) TRIM, TRIM(n),

COUNT

For varray & nested table - COUNT method returns the number of elements with space allocated in varray and nested table.

For index by table - COUNT method returns the number of elements in index by table.

COUNT can be smaller than LIMIT for varray. e.g

```
DECLARE
TYPE number_table IS TABLE OF INTEGER;
Number_list    NUMBER_TABLE := number_table(1,2,3,4,5);
BEGIN
DOPL('how many elements);
DOPL('-----------------');
DOPL('count  ['|| number_list.COUNT || ' ] ');
END;
/
```

DELETE(n)   DELETE(n,m)

DELETE method is overloaded procedure . it has one version that takes a single formal parameter. The parameter must be a valid subscript value in the collection. This version will remove the element in that subscript. DELETE(n)

The other version takes two formal parameter, both parameters must be valid subscript value in the collection. This version deletes a continuous inclusive range of elements from the collection.DELETE(n,m)

e.g number_list.DELETE(2,4);

EXISTS(n)

EXISTS method really a function.it has only one formal parameter. It takes a subscript value. Exists is only the method that will not raise COLLECTION_IS_NULL exception.for null element collection.

Null elements collections have two varieties. First, varrays and nested tables constructed with a null constructor. Second, index by tables that have zero element initiallized.

EXTEND   EXTENT(n)   EXTEND(n,i)

The EXTEND is a procedure. It is an overloaded procedure.

It has one version that takes no formal parameters. When used with no formal parameter EXTEND allocates space for a new elementin collection however if you attempt to EXTEND space beyond limit os varray, it will raise an exception.

A second version takes a single formal parameter. The parameter must be a valid integer value.EXTEND with a single actual parameter will allocates space for that number of elements specified by actual parameter. if you attempt to EXTEND space beyond limit os varray, it will raise an exception.

The last version takes two fromal parameters. Both parameters must be valid integers. The second must also be a valid subscript valuein collection. This version allocates element space equal to the first actual parameter. Then it copies the contents of the referenced subscript found in second actual parameter. E.g

Number_last.EXTEND(2)   Allocate two null elements.
Number_last.EXTEND(3,2) Allocates three elements and copy two.

FIRST

The FIRST method is a function.it return the lowest subscript value used in the collection. If it is numeric index it returns PLS_INTEGER. If it is index by table it returns VARCHAR2 or long datatype. You cannot use first methos in a range for – loop when index is non-numeric.

LAST

LAST method is function. It returns the highest subscript value used ina collection. If it is a numeric index, it returns a PLS_INTEGER. If it is index by table, it returns a VARCHAR2 OR LONG datatype. You cannot use the LAST method ina range for-loop when index is non-numeric.

LIMIT

The LIMIT is a function. It returns the highest possible subscript value used in a varray. it has no values for the other two collection types. It returns aPLS_INTEGER.

NEXT(n)

NEXT method is a function.it returns a next subscript value used ina collection. If there is no higher subscript value, it returns a null. If it is anumeric index, it returns a PLS_INTEGER. If it is a index by table, it returns a VARCHAR2 or LONG datatype.

PRIOR(n)

The PRIOR method is a function. It returns the prior subscript value used in a collection. If there is no lower subscript value, it returns a null. If it is a numeric index, it returns a PLS_INTEGER. Is it is index by table, it returns a VARCHAR@ or LONG datatype.

TRIM   TRIM(n)

The TRIM method is a procedure. Ikt is an overloaded procedure.

It has one version that takes no formal parameters. When used without formal parameter TRIM deallocates space for an element in a collection. However if you attempt to TRIM space below zero, it will raise an exception.

The other version takes a single formal parameter. The parameter must be a valid integer value. TRIM witha single actual parameter will deallocate space for the number of elements specified by the actual parameter. Like the version without parameter, attempting to trim space below zero elements will raise an exception.

239] syntax

1] CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW *view*
[(*alias*[, *alias*]...)]
AS *subquery*
[WITH CHECK OPTION [CONSTRAINT *constraint*]]
[WITH READ ONLY [CONSTRAINT *constraint*]];

NOFORCE creates the view only if the base tables exist (This is the default.)
VIEW is the name of the view
alias specifies names for the expressions selected by the view's query
(The number of aliases must match the number of expressions
selected by the view.)
subquery is a complete SELECT statement (You can use aliases for the
columns in the SELECT list.)
WITH CHECK OPTION specifies that only rows accessible to the view can be inserted or updated
constraint is the name assigned to the CHECK OPTION constraint
WITH READ ONLY ensures that no DML operations can be performed on this view

Varrays

2] CREATE OR REPLACE TYPE marks_va AS VARRAY(4) OF NUMBER(3);

```
/
DECLARE
v_marks marks_va:=marks_va(75,80,90,85);
BEGIN
FOR i IN 1..v_marks.COUNT
LOOP
DBMS_OUTPUT.PUT_LINE(v_marks(i));
END LOOP;
END;

CREATE TABLE student_va
(rollno NUMBER(4),
marks marks_va);

INSERT INTO student_va
VALUES(1001,marks_va(85,90,95,70));
```

Nested Tables

```
3] CREATE OR REPLACE TYPE type-name AS TABLE OF  element_type [ NOTNULL ];
CREATE OR REPLACE TYPE emp_nt AS TABLE OF  VARCHAR2(20);
```

Index by table

```
4] CREATE OR REPLACE TYPE type-name AS TABLE OF  element_type [NOTNULL]
    INDEX BY [ PLS_INTEGER | BINARY_ INTEGER | VARCHAR2(size)];

5] CREATE GLOBAL TEMPORARY TABLE flight_schedule
( flight_no                 NUMBER(4),
  destination        VARCHAR2(25),
  time               DATE)
ON COMMIT DELETE ROWS;                    ----- For transaction specific tables.

CREATE GLOBAL TEMPORARY TABLE flight_schedule
( flight_no                 NUMBER(4),
  destination               VARCHAR2(25),
  time                      DATE)
ON COMMIT PRESERVE ROWS;     ----- For session specific tables.
```

240] Standard Queries

1) To delete duplicate rows

```
DELETE from emp WHERE rowid NOT IN (SELECT MAX(rowid) FROM emp
GROUP BY empno); To delete duplicate rows.
```

2) ODD RECORDS

```
SELECT * FROM emp WHERE (rowid,1) IN(SELECT rowid, mod(rownum,2) FROM emp);
```

3) EVEN RECORDS

```
SELECT * FROM emp WHERE (rowid,0) IN(SELECT rowid, mod(rownum,2) FROM emp);
```

4) Nth ROW

```
SELECT * FROM emp WHERE rownum=1 AND rowid NOT IN(SELECT rowid FROM emp
WHERE rownum<10);
```

5) Top n records

```
SELECT * FROM(SELECT * FROM emp ORDER BY empno) WHERE rownum<=5 ORDER BY
empno;
```

6) Last n records

```
SELECT * FROM (SELECT * FROM emp ORDER BY empno desc) WHERE rownum<=5 ORDER
BY empno;
```

7) To retrieve hiredate day order by day

```
SELECT hiredate,TO_CHAR(hiredate,'day') FROM emp ORDER BY TO_CHAR(hiredate,'d');
```

8) To retrieve salary in words SELECT sal,TO_CHAR(TO_DATE(sal,'j'),'jsp') FROM emp;

9) To display first day of current month SELECT TO_CHAR(TRUNC(sysdate,'month'),'day') FROM dual;

10) To display Last day of current month

```
SELECT TO_CHAR(last_day(sysdate),'day') FROM dual;
```

```
SELECT TO_CHAR(TRUNC(sysdate,'year'),'day') FROM dual;
```

12) To display first day of Next month

```
SELECT TO_CHAR(sysdate,'month'),'day') FROM dual;--GIVES FIRST DAY OF CURRENT
MONTH
-- AS ROUND(SYSDATE) GIVES 01-JAN-07 ON 01-JAN-07
```

13) To select duplicate rows groupwise.

```
SELECT deptno FROM emp WHERE deptno IN(SELECT MAX(deptno) FROM emp GROUP BY
deptno
HAVING COUNT(deptno)>1);
Result   10
   10       10
   20
   20
```

14) To select nth highest paid employee

```
 SELECT * FROM emp e
 WHERE 4=(SELECT COUNT(DISTINCT(sal)) FROM emp x
 WHERE e.sal<=x.sal);
```

15) To select nth lowest paid employee

```
SELECT * FROM emp e
WHERE 4=(SELECT COUNT(DISTINCT(sal)) FROM emp x
 WHERE  e.sal>=x.sal);
```

16) To select second max salary

```
SELECT MAX(sal) FROM emp WHERE sal=(SELECT MAX(sal) FROM emp WHERE sal <
(SELECT MAX(sal) FROM emp));
```

241] Data Dictionary Tables

OBJECTS

```
SQL> desc user_objects
OBJECT_NAME                             VARCHAR2(128
SUBOBJECT_NAME                          VARCHAR2(30)
OBJECT_ID                                NUMBER
DATA_OBJECT_ID                          NUMBER
OBJECT_TYPE                             VARCHAR2(18)
CREATED                                 DATE
LAST_DDL_TIME                           DATE
TIMESTAMP                               VARCHAR2(19)
STATUS                                  VARCHAR2(7)
TEMPORARY                                VARCHAR2(1)
GENERATED                               VARCHAR2(1)
SECONDARY                               VARCHAR2(1)
```

TEXT FOR OBJECTS

```
SQL> desc user_source
-----------------------------------
 NAME                                   VARCHAR2(30)
 TYPE                                   VARCHAR2(12)
 LINE                                   NUMBER
 TEXT                                   VARCHAR2(4000)
```

TRIGGERS

```
SQL> desc user_triggers
---------------------------------------- -------- ----------------
TRIGGER_NAME                            VARCHAR2(30)
TRIGGER_TYPE                            VARCHAR2(16)
TRIGGERING_EVENT                                VARCHAR2(227)
TABLE_OWNER                             VARCHAR2(30)
BASE_OBJECT_TYPE                        VARCHAR2(16)
TABLE_NAME                              VARCHAR2(30)
COLUMN_NAME                             VARCHAR2(4000)
REFERENCING_NAMES                       VARCHAR2(128)
WHEN_CLAUSE                             VARCHAR2(4000)
STATUS                                  VARCHAR2(8)
DESCRIPTION                             VARCHAR2(4000)
ACTION_TYPE                                     VARCHAR2(11)
TRIGGER_BODY                             LONG
```

```
SELECT TRIGGER_BODY,TABLE_NAME,STATUS FROM USER_TRIGGERS WHERE
TRIGGER_NAME='TRIG_MYEMP';
```

VIEWS

```
SQL> desc user_views;
 Name                           Null?    Type
 ---------------------------------------- -------- --------------------
 VIEW_NAME                              NOT NULL VARCHAR2(30)
 TEXT_LENGTH                            NUMBER
 TEXT                                   LONG
 TYPE_TEXT_LENGTH                               NUMBER
 TYPE_TEXT                      VARCHAR2(4000)
 OID_TEXT_LENGTH                NUMBER
 OID_TEXT                       VARCHAR2(4000)
 VIEW_TYPE_OWNER                     VARCHAR2(30)
 VIEW_TYPE                      VARCHAR2(30)
 SUPERVIEW_NAME                 VARCHAR2(30)
```

SEQUENCES

```
SQL> desc user_sequences;
Name                    Null?    Type
--------------------------------------- -------- ------------
SEQUENCE_NAME                 NOT NULL VARCHAR2(30)
MIN_VALUE                     NUMBER
MAX_VALUE                     NUMBER
INCREMENT_BY                  NOT NULL NUMBER
CYCLE_FLAG                    VARCHAR2(1)
ORDER_FLAG                    VARCHAR2(1)
CACHE_SIZE                    NOT NULL NUMBER
LAST_NUMBER                   NOT NULL NUMBER
```

USER_USERS

```
SQL> desc user_users;
 Name                   Null?    Type
 --------------------------------------- -------- ---------------
 USERNAME                     NOT NULL VARCHAR2(30)
 USER_ID                      NOT NULL NUMBER
 ACCOUNT_STATUS               NOT NULL VARCHAR2(32)
 LOCK_DATE                    DATE
 EXPIRY_DATE                       DATE
 DEFAULT_TABLESPACE           NOT NULL VARCHAR2(30)
 TEMPORARY_TABLESPACE              NOT NULL VARCHAR2(30)
 CREATED                      NOT NULL DATE
 INITIAL_RSRC_CONSUMER_GROUP       VARCHAR2(30)
 EXTERNAL_NAME                VARCHAR2(4000)
```

ALL_USERS

```
SQL> desc all_users;
 Name                   Null?    Type
 --------------------------------------- -------- -------------
 USERNAME                     NOT NULL VARCHAR2(30)
 USER_ID                      NOT NULL NUMBER
 CREATED                      NOT NULL DATE
```

USER_SYS_PRIVS

```
SQL> desc user_sys_privs;
 Name                   Null?    Type
 --------------------------------------- -------- ------------
 USERNAME                     VARCHAR2(30)
 PRIVILEGE                                 NOT NULL VARCHAR2(40)
 ADMIN_OPTION                              VARCHAR2(3)
```

USER_ROLE_PRIVS

```
SQL> desc user_role_privs;
 Name                          Null?   Type
 --------------------------------------- -------- -------------
 USERNAME                                  VARCHAR2(30)
 GRANTED_ROLE                              VARCHAR2(30)
 ADMIN_OPTION                              VARCHAR2(3)
 DEFAULT_ROLE                              VARCHAR2(3)
 OS_GRANTED                                        VARCHAR2(3)
```

USER_CATALOG

```
SQL> desc user_catalog;
 Name                          Null?   Type
 --------------------------------------- -------- --------------
 TABLE_NAME                                        NOT NULL VARCHAR2(30)
 TABLE_TYPE                                  VARCHAR2(11)
```

user_tables

```
SQL> desc user_tables;
 Name                          Null?   Type
 --------------------------------------- -------- -----------------
 TABLE_NAME                                                NOT NULL VARCHAR2(30)
 TABLESPACE_NAME                                           VARCHAR2(30)
 CLUSTER_NAME                                      VARCHAR2(30)
 IOT_NAME                                          VARCHAR2(30)
 PCT_FREE                                          NUMBER
 PCT_USED                                          NUMBER
 INI_TRANS                                         NUMBER
 MAX_TRANS                                         NUMBER
 INITIAL_EXTENT                                    NUMBER
 NEXT_EXTENT                                       NUMBER
 MIN_EXTENTS                                       NUMBER
 MAX_EXTENTS                                       NUMBER
 PCT_INCREASE                                      NUMBER
 FREELISTS                                         NUMBER
 FREELIST_GROUPS                                   NUMBER
 LOGGING                                           VARCHAR2(3)
 BACKED_UP                                         VARCHAR2(1)
 NUM_ROWS                                          NUMBER
 BLOCKS                                            NUMBER
 EMPTY_BLOCKS                                      NUMBER
 AVG_SPACE                                         NUMBER
 CHAIN_CNT                                         NUMBER
 AVG_ROW_LEN                                       NUMBER
```

AVG_SPACE_FREELIST_BLOCKS	NUMBER
NUM_FREELIST_BLOCKS	NUMBER
DEGREE	VARCHAR2(10)
INSTANCES	VARCHAR2(10)
CACHE	VARCHAR2(5)
TABLE_LOCK	VARCHAR2(8)
SAMPLE_SIZE	NUMBER
LAST_ANALYZED	DATE
PARTITIONED	VARCHAR2(3)
IOT_TYPE	VARCHAR2(12)
TEMPORARY	VARCHAR2(1)
SECONDARY	VARCHAR2(1)
NESTED	VARCHAR2(3)
BUFFER_POOL	VARCHAR2(7)
ROW_MOVEMENT	VARCHAR2(8)
GLOBAL_STATS	VARCHAR2(3)
USER_STATS	VARCHAR2(3)
DURATION	VARCHAR2(15)
SKIP_CORRUPT	VARCHAR2(8)
MONITORING	VARCHAR2(3)
CLUSTER_OWNER	VARCHAR2(30)
DEPENDENCIES	VARCHAR2(8)