The EXPLAIN PLAN statement displays execution plans chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements. A statement's execution plan is the sequence of operations Oracle performs to run the statement.

**EXPLAIN PLAN** [parses](http://www.orafaq.com/wiki/Parsing) a [query](http://www.orafaq.com/wiki/Query) and records the "plan" that Oracle devises to execute it. By examining this plan, you can find out if Oracle is picking the right [indexes](http://www.orafaq.com/wiki/Index) and [joining](http://www.orafaq.com/wiki/Join) your [tables](http://www.orafaq.com/wiki/Table) in the most efficient manner

SQL> EXPLAIN PLAN FOR select \* from dept where deptno = 40;

Explained.

SQL> set linesize 132

SQL> SELECT \* FROM TABLE(dbms\_xplan.display);

PLAN\_TABLE\_OUTPUT

---------------------------------------------------------------------------------------

Plan hash value: 2852011669

---------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

---------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 1 | 20 | 1 (0)| 00:00:01 |

| 1 | TABLE ACCESS BY INDEX ROWID| DEPT | 1 | 20 | 1 (0)| 00:00:01 |

|\* 2 | INDEX UNIQUE SCAN | PK\_DEPT | 1 | | 0 (0)| 00:00:01 |

---------------------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

2 - access("DEPTNO"=40)

14 rows selected.

SQL\*Plus also offers an AUTOTRACE facility that will display the query plan and execution statistics as each query executes. Example:

SQL> SET AUTOTRACE ON

SQL> select \* from dept where deptno = 40;

DEPTNO DNAME LOC

---------- -------------- -------------

40 OPERATIONS BOSTON

Execution Plan

----------------------------------------------------------

Plan hash value: 2852011669

---------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

---------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 1 | 20 | 1 (0)| 00:00:01 |

| 1 | TABLE ACCESS BY INDEX ROWID| DEPT | 1 | 20 | 1 (0)| 00:00:01 |

|\* 2 | INDEX UNIQUE SCAN | PK\_DEPT | 1 | | 0 (0)| 00:00:01 |

select \* from table(dbms\_xplan.display) –to display the execution plan

SELECT

\*

FROM

table

(

DBMS\_XPLAN.DISPLAY

(

'plan\_table' -- default plan table name

, NULL -- NULL to show last statement inserted into plan table

, 'all' -- show all available metrics

)

);

What you end up with is a tabular representation of the execution plan; the table is a top-down, left-to-right traversal of the execution tree. Each operation is printed on a single line, together with more detailed information about that operation. The sequence and types of operations depend on what the query optimizer considers to be the best plan, that is the one with the lowest cost. The components that are factored into the cost of a query are

* Cardinality
* Access methods
* Join methods
* Join types
* Join orders
* Partition pruning
* Parallel execution

We shall now discuss each of these components separately.

## Cardinality

What cardinality refers to is the uniqueness (or lack thereof) of data in a particular column of a database table. It is a measure of the number of distinct elements in a column. A low cardinality implies that a column has very few distinct values, and is thus not very selective.

the cardinality shows the number of rows estimated to come out of each operation. The cardinality is computed from table and column statistics you can see runtime cardinality information by using the GATHER\_PLAN\_STATISTICS hint in your query, after

## Access Methods

* A **full table scan** reads all rows from a heap-organized table and filters out the ones that do not match the WHERE clause predicates. Each formatted block of data under the high water mark (HWM) is read only once and the blocks are read in sequence. A multi-block read can be performed to speed up the table scan:

A full table scan is typically employed if

* + no index exists;
  + the index cannot be used because the query predicate has a function applied to an indexed column (in a non-function-based index);
  + the optimizer decides against an index skip scan because the query predicate does not include the leading edge of a (B-tree) index;
  + the table has a high degree of parallelism, which makes the optimizer biased towards full table scans;
  + the query uses the FULL hint.
* In a **table access by ROWID**, Oracle looks up each selected row of a heap-organized table based on its ROWID, which specifies the data file, the data block within that file, and the location of the row within that block. The ROWID is obtained either from the WHERE clause predicate or through an index scan. If the execution plan shows a line TABLE ACCESS BY INDEX ROWID BATCHED it means that Oracle retrieves a bunch of ROWIDs from the index and then tries to access rows in block order to reduce the number of times each block needs to be accessed.

.

* For an **index unique scan** only one row of a (B-tree) index or index-organized table will be returned because of an equality predicate on a unique index or a primary key constraint; the database stops looking for more rows once it has found its match because there cannot be any more matches thanks to the UNIQUE constraint. Hence, all columns of a unique index must be referenced with equality operators. Please observe the index itself must be unique: creating an index on a column and subsequently adding a UNIQUE or PRIMARY KEY constraint to the table is not enough, as the index is not aware of uniqueness; you are liable to end up with an index range scan.
* An **index range scan** scans values in order. By default, Oracle stores and scans indexes and index-organized tables in ascending order. Oracle accesses adjacent index entries and uses the ROWID values to retrieve the rows in ascending order. If multiple index entries happen to have the same keys, Oracle returns the entries in ascending order by ROWID.

The database chooses an index range scan if the leading columns of a non-unique index are specified in the WHERE clause or if the leading columns of a unique index have ranges rather than single values specified.

* If the entire index is read in order, then we are dealing with an **full index scan**. A full index scan does not read every block in the index though. Instead it reads the root block and goes down the left-hand side (ascending scan) or right-hand side (descending scan) of the branch blocks until it reaches a leaf block. From there it reads the index on a block-by-block basis.

The full index scan is used when one of the following situations arises:

* + a predicate references a (non-leading) column in an index;
  + no predicate is specified but all columns in the table as well as query are in the index, and at least one indexed column is not null;
  + the query has an ORDER BY clause on indexed non-null columns;
  + the query has a GROUP BY clause where all aggregation columns are present in the index.
* A **fast full index scan** reads index blocks as they exist on disk. The index (entries on the leaf blocks) rather than the table is read in multi-block I/O mode.

Oracle looks at a this access path whenever a query only asks for attributes in the index. It’s an alternative to a full table scan when the index contains all columns needed for the query, and at least one column in the index key has a NOT NULL constraint.

Because the index is not read in order (because of the multi-block I/O), a sort operation cannot be eliminated as with the full index scan.

* For queries with predicates on all but the leading column of a composite index, an **index skip scan** is a possibility. The index skip scan is also an option if the leading column has few distinct values.

The optimizer logically splits a composite index into smaller sub-indexes based on the number of distinct values in the leading column(s) of the index. The lower the number of distinct values, the fewer logical sub-indexes, the more efficient the scan is. Index blocks that do not meet the filter condition on the non-leading column are immediately skipped, hence the name. An index skip scan can of course only be efficient if the non-leading columns are highly selective.

* An **index join scan** is performed if all data can be retrieved from a combination of multiple indexes, which are hash-joined on the ROWIDs. Because all data is available in the indexes, no table access is needed.

An index join is often more expensive than simply scanning the most selective index and subsequently probing the table. It cannot be used to eliminate a sort operation.

* Whereas in traditional B-tree indexes each entry point refers to exactly one row, a bitmap index’s entry points refer to multiple rows. That is, each index key stores pointers to multiple rows. Each bit corresponds to a possible ROWID; if the bit is set, then the row with the corresponding ROWID contains the key’s value. The bit position is converted to an actual ROWID by a mapping function. Internally, Oracle stores the bitmap index in a B-tree structure for quick searching.

Bitmaps are frequently used in data warehousing (OLAP) environments, where ad hoc queries are commonplace. Typically, a bitmap index is recommended if the indexed columns have low cardinality and the indexed table is rarely modified or even read-only. Bitmap indexes are not appropriate for OLTP applications, though. If, for instance, the indexed column of a single row is updated, the database locks the index key entry, which points to many rows. Consequently, all these rows are locked.

Back to business. If the predicate on a bitmap-indexed column contains an equality operator, the query optimizer considers the **bitmap index single value** access path to look up a single key value. A single bitmap is scanned for all positions containing a value of 1. All matching values are converted into ROWIDs, which in turn are used to find the corresponding rows.

* A B-tree index can have an index range scan for ranges of values specified in the WHERE clause. Its counterpart for bitmap indexes is the **bitmap index range scan**.
* A **bitmap merge** is typically preferred by the optimizer when bitmaps generated from a bitmap index range scan are combined with an OR operation between two bitmaps.
* When a query accesses a table in an indexed cluster, the optimizer mulls over a **cluster scan**. Tables in [table clusters](http://docs.oracle.com/cd/E16655_01/server.121/e17633/tablecls.htm#CNCPT608) have common columns and related data stored in the same blocks. The proximity of the physical location of these shared columns reduces disk I/O when joining clustered tables and accessing related data. Table clusters are appropriate for tables that are rarely modified and do not require full table scans as in the case of retrieval of a lot of rows.

An index cluster is, as you might expect, a (B-tree) index on a cluster; the cluster index associates the cluster key with the database block address of the block with the relevant data. In a cluster scan, Oracle scans the index to obtain the database block addresses of all rows required. Because the index cluster is a separate entity, reading or storing rows in a table cluster requires at least two I/Os: one (or more) to retrieve/store the key value from/in the index, and one to read/write the row from/to the cluster.

* If you create a table cluster with the HASHKEYS clause, you end up with a hash cluster. A hash cluster is similar to an indexed cluster, only the index key is replaced with a hash function. All rows with the same hash are stored in the same data block. Because no separate cluster index exists as in the case with an indexed cluster, I/O can be usually halved.

A **hash scan** is used to locate rows based on a hash value of the key in a hash cluster. A disadvantage of hash clusters is the absence of range scans on cluster keys that are not in the index, in which case a full table scan must be performed.

## Join Methods

Join methods refer to the way in which two row sources are joined with each other. The query optimizer designates one table the outer table and the other the inner table. If more than two tables need to be joined, the optimizer analyses all permutations to determine the optimal join sequence.

The join method dictates to a large degree the cost of joins:

* A **hash join** is usually preferred for (equi)joining large data sets. The query optimizer takes the smaller of two data sources to build a (deterministic) hash table in memory. It then scans the larger table and performs the same hashing on the join columns. For the in-memory hash table, the database probes each value and if it matches the predicate’s conditions returns the corresponding row.

If the smaller data set fits in memory, there is only the cost of a single read pass over both data sets. If the hash table does not fit in memory, then Oracle partitions it. The join is then performed partition by partition, which uses a lot of sort area memory and causes a lot of I/O to the temporary tablespace.

* A **nested loop join** is typically used when small subsets of tables are joined or if there is an efficient way of accessing the inner table, for example with an index lookup. For every row selected from the outer table, the database scans all rows of the inner table. If there is an index on the inner table, then it can be used to access the inner data by ROWID.

The database can read several rows from the outer row source in a batch, which is typically part of an [adaptive plan](http://docs.oracle.com/cd/E16655_01/server.121/e15858/tgsql_optcncpt.htm#TGSQL221).

It is not uncommon to see two nested loops in the execution plan (as of 11g) because Oracle batches multiple I/O requests and process these with a vector I/O, which means that a set of ROWIDs is sent to the requesting operating system in an array. What Oracle does with two nested loops is basically the following:

* 1. Iterate through the inner nested loop to obtain the requested outer source rows.
  2. Cache the data in the [PGA](http://docs.oracle.com/cd/E16655_01/server.121/e15857/tune_pga.htm#TGDBA346).
  3. Find the matching ROWID from the inner loop’s inner row source.
  4. Cache the data in the PGA.
  5. Organize the ROWIDs for more efficient access in the cache.
  6. Iterate through the outer loop to retrieve the data based on the cached ROWIDs; the result set of the inner loop becomes the outer row source of the outer nested loop.
* A **sort-merge join** is done when join conditions are inequalities (i.e. not an equijoin). It is commonly chosen if there is an index on one of the tables that eliminates a sort operation. The sort operation on the first data set can be avoided if there is such an index. The second data set will always be sorted, regardless of any indexes. It generally performs better for large data sets than a nested loop join.

A sort-merge join proceeds in two steps:

1. Sort join: both tables are sorted on the join key. 1. Merge join: sorted tables are merged.

Oracle accesses rows in the PGA instead of the SGA with a sort-merge join, which reduces logical I/O because it avoids repeated latching blocks in the database buffer cache; a [latch](http://oracle.readthedocs.io/en/latest/glossary.html#term-latch)protects shared data structures in the SGA from simultaneous access.

* Whenever a table has no join conditions to any of the other tables in the statement, Oracle picks a **Cartesian join**, which is nothing but a Cartesian product of the tables. Because the number of rows of a Cartesian join of two tables is the product of the number of rows of the tables involved, it is generally used only if the tables are small. This join method is, however, very rare in production environments.

Partial join evaluation is available from Oracle Database 12c onwards. It allows joined rows that would otherwise have to be eliminated by a SORT UNIQUE operation to be removed during the execution of the join an inner join or semi-join instead. This optimization affects DISTINCT, MIN(), MAX(), SUM(DISTINCT), AVG(DISTINCT), COUNT(DISTINCT) , branches of UNION, MINUS, and INTERSECToperators, [ NOT ] EXISTS subqueries, and so on. For instance, a HASH JOIN > SORT UNIQUE is replaced by a HASH JOIN SEMI > HASH UNIQUE combo.

## Join Types

Join types are easier to explain because they are determined by what you have typed. There are four categories of join types: i) inner joins, ii) outer joins (left, right, and full), iii) semi-joins, and iv) anti-joins. Even though semi-joins and anti-joins are syntactically subqueries, the optimizer beats them until they accept their fate as joins.

Semi-joins can occur when the statement contains an IN or EXISTS clause that is not contained in an OR branch. Analogously, anti-joins are considered if the statement contains a NOT IN or NOT EXISTS clause that is not contained in an OR branch. Moreover, an anti-join can be used the statement includes an outer join and has IS NULL applied to a join column.

## Join Orders

So, how does Oracle determine the order of joins? The short answer is cost. Cardinality estimates and access paths largely determine the overall cost of joins:

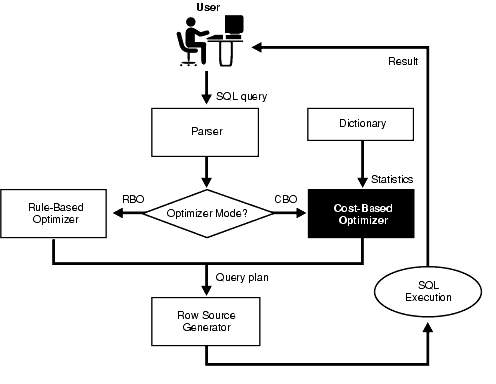
* Whenever a particular join results in at most one row (e.g. because of UNIQUE or PRIMARY KEYconstraints) it goes first.
* For outer joins, the row-preserving (outer) table comes after the other tables to ensure that all rows that do not satisfy the join condition can be added correctly.
* When Oracle converts a subquery into an anti- or semi-join, the subquery’s table(s) come after tables in the outer (connected/correlated) query block. Hash anti-joins and semi-joins can sometimes override the ordering though.
* If [view merging](http://blogs.oracle.com/optimizer/entry/optimizer_transformations_view_merging_part_1) is impossible, then all tables in the view are joined before joining the view to the tables outside the view.

## Partition Pruning

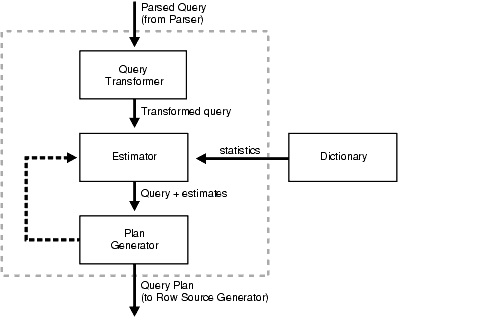
Partition pruning, which is also known as partition elimination, is used for partitioned tables when not all partitions need to be accessed.Pruning is visible in the execution plan as integers in the columns PSTART, the number of the first partition, and PSTOP, the number of the last partition to be accessed. [[2]](http://oracle.readthedocs.io/en/latest/sql/plans/explain-plan.html#partition) If you do not see a number (e.g. KEY), don’t worry! It means that Oracle was not able to determine the partition numbers at parse time but thinks that dynamic pruning (i.e. during execution) is likely to occur. This typically happens when there is an equality predicate on a partitioning key that contains a function, or if there is a join condition on a partitioning key column that, once joined with a partitioned table, is not expected to join with all partitions because of a filter predicate.For hash-partitioned tables, there can only be pruning if the predicate on the partition key column is an equality or IN-list predicate. If the hash partition key involves more than one column, then all these columns must appear in the predicate for partition pruning to be able to occur.

## Parallel Execution

Sometimes Oracle executes statements in parallel, which can significantly improve the runtime performance of a query. The query coordinator (QC) initiates the parallel execution of a SQL statement. The parallel server [processes](http://oracle.readthedocs.io/en/latest/glossary.html#term-processes) (PSPs) are responsible for the actual work that is done in parallel. The coordinator distributes the work to the PSPs, and may have to do some of the work that cannot be done in parallel. A classic example is the SUM(...) operation: the PSPs calculate the subtotals but the coordinator has to add the subtotals from each PSP to obtain the final tally.



**Cost based optimizer components:**



Bitmap Index vs. B-tree Index: Which and When?

Comparing the Indexes

There are several disadvantages to using a bitmap index on a unique column--one being the need for sufficient space (and Oracle does not recommend it). However, the size of the bitmap index depends on the cardinality of the column on which it is created as well as the data distribution. Consequently, a bitmap index on the GENDER column will be smaller than a B-tree index on the same column. In contrast, a bitmap index on EMPNO (a candidate for primary key) will be much larger than a B-tree index on this column. But because fewer users access decision-support systems (DSS) systems than would access transaction-processing (OLTP) ones, resources are not a problem for these applications.

in summary, bitmap indexes are best suited for DSS regardless of cardinality for these reasons:

With bitmap indexes, the optimizer can efficiently answer queries that include AND, OR, or XOR. (Oracle supports dynamic B-tree-to-bitmap conversion, but it can be inefficient.)

With bitmaps, the optimizer can answer queries when searching or counting for nulls. Null values are also indexed in bitmap indexes (unlike B-tree indexes).

Most important, bitmap indexes in DSS systems support ad hoc queries, whereas B-tree indexes do not. More specifically, if you have a table with 50 columns and users frequently query on 10 of them either the combination of all 10 columns or sometimes a single column creating a B-tree index will be very difficult. If you create 10 bitmap indexes on all these columns, all the queries can be answered by these indexes, whether they are queries on all 10 columns, on 4 or 6 columns out of the 10, or on a single column. The AND\_EQUAL hint provides this functionality for B-tree indexes, but no more than five indexes can be used by a query. This limit is not imposed with bitmap indexes.

In contrast, B-tree indexes are well suited for OLTP applications in which users queries are relatively routine (and well tuned before deployment in production), as opposed to ad hoc queries, which are much less frequent and executed during nonpeak business hours. Because data is frequently updated in and deleted from OLTP applications, bitmap indexes can cause a serious locking problem in these situations.

The data here is fairly clear. Both indexes have a similar purpose: to return results as fast as possible. But your choice of which one to use should depend purely on the type of application, not on the level of cardinality.

--> Finding the Cost of an Index

Building and maintaining an index structure can be expensive, and it can consume resources such as disk space, CPU, and I/O capacity. Designers must ensure that the benefits of any index outweigh the negatives of index maintenance.

Use this simple estimation guide for the cost of index maintenance: each index maintained by an INSERT, DELETE, or UPDATE of the indexed keys requires about three times as much resource as the actual DML operation on the table. Thus, if you INSERT into a table with three indexes, then the insertion is approximately 10 times slower than an INSERT into a table with no indexes. For DML, and particularly for INSERT-heavy applications, the index design should be seriously reviewed, which might require a compromise between the query and INSERT performance.

Good cursor usage and management

Maintaining user connections is equally important to minimizing the parsing activity on the system. Parsing is the process of interpreting a SQL statement and creating an execution plan for it. This process has many phases, including syntax checking, security checking, execution plan generation, and loading shared structures into the shared pool. There are two types of parse operations

Database Indexes

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space.By default, Oracle creates B-tree indexes.

Database Indexes

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space.By default, Oracle creates B-tree indexes.

Types of Index

A)Clustered Index

B)Non Clustered Index

Clustered Index:- Clustering alters the data block into a certain distinct order to match the index, resulting in the row data being stored in order. Therefore, only one clustered index can be created on a given database table.

Typically created on primary key column.

Non Clustered Index:-The data is present in random order, but the logical ordering is specified by the index. The data rows may be randomly spread throughout the table. The non-clustered index tree contains the index keys in sorted order, with the leaf level of the index containing the pointer to the page and the row number in the data page

The physical order of the rows is not the same as the index order.

Typically created on column used in JOIN, WHERE, and ORDER BY clauses.

Difference between Clustered and non Clustered Index:-

1)Clustered indices can greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected.Non Clustered index is Good for tables whose values may be modified frequently.

2) Clustered index usually saved in side the table however non Clustered Index saved in separate tables in database.

There are multiple ways to implement Index:-

1)Bitmap Index:-A bitmap index is a special kind of index that stores the bulk of its data as bitmaps and answers most queries by performing bitwise logical operations on these bitmaps. The most commonly used index, such as B+tree are most efficient if the values it indexes do not repeat or repeat a smaller number of times

2)B-tree Index:-By default, the Oracle creates a b\_tree index. In a b-tree, you walk the branches until you get to the node that has the data you want to use. In the classic b-tree structure, there are branches from the top that lead to leaf nodes that contain the data.

3)Dense Index:- A dense index in database is a file with pairs of keys and pointers for every record in the data file. Every key in this file is associated with a particular pointer to a record in the sorted data file. In clustered indices with duplicate keys, the dense index points to the first record with that key.

4)Sparse Index:-A sparse index in databases is a file with pairs of keys and pointers for every block in the data file. Every key in this file is associated with a particular pointer to the block in the sorted data file. In clustered indices with duplicate keys, the sparse index points to the lowest search key in each block. primary key is a sparse index.

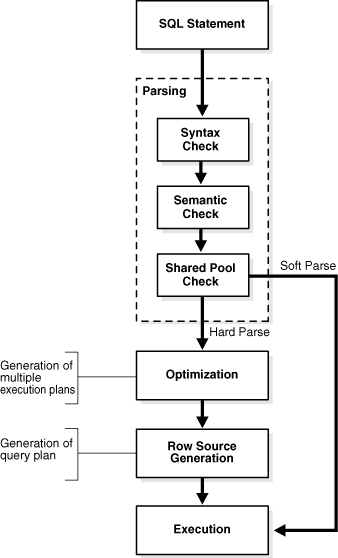
5)Reverse Index:- reverse key index reverses the key value before entering it in the index. E.g., the value 24538 becomes 83542 in the index. Reversing the key value is particularly useful for indexing data such as sequence numbers, where new key values monotonically increase.

6)Function based Index:- An index is defined on the result of a function applied to one or more columns of a single table. Functional indexes can be used to obtain fast access to data based on the result of function calls.

Ex:-This query can use an index, if one has been defined on the result of the lower(coll) operation:

***How a Query is parsed in SQL engine at high level steps.***

***Stages of SQL Processing***

****

**Oracle Database must check each SQL statement for syntactic validity.**

**A semantic check determines whether a statement is meaningful, for example, whether the objects and columns in the statement exist.**

**Shared Pool Check**

**During the parse, the database performs a shared pool check to determine whether it can skip resource-intensive steps of statement processing.**

**To this end, the database uses a hashing algorithm to generate a hash value for every SQL statement. The statement hash value is the**[**SQL ID**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-894D17CF-F10C-4BD5-B2A2-E3AC23759B78)**shown in V$SQL.SQL\_ID. This hash value is deterministic within a version of Oracle Database, so the same statement in a single instance or in different instances has the same SQL ID.**

**When a user submits a SQL statement, the database searches the**[**shared SQL area**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-577D0DD3-FB98-4B1B-BCFB-95313BB35C01)**to see if an existing parsed statement has the same hash value. The hash value of a SQL statement is distinct from the following values:**

* **Memory address for the statement**

**Oracle Database uses the SQL ID to perform a keyed read in a lookup table. In this way, the database obtains possible memory addresses of the statement.**

* **Hash value of an**[**execution plan**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-800F2BE7-A383-4FF3-A9A6-FCE786AABA3B)**for the statement**

**A SQL statement can have multiple plans in the shared pool. Typically, each plan has a different hash value. If the same SQL ID has multiple plan hash values, then the database knows that multiple plans exist for this SQL ID.**

**Parse operations fall into the following categories, depending on the type of statement submitted and the result of the hash check:**

**Hard parse**

**If Oracle Database cannot reuse existing code, then it must build a new executable version of the application code. This operation is known as a**[**hard parse**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-AB764C9E-2F03-49A9-BF8B-36A9FBD03BCE)**, or a**[**library cache miss**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-D12396D3-DCDA-4A61-8891-64AD81876EFD)**.**

* **During the hard parse, the database accesses the library cache and data dictionary cache numerous times to check the data dictionary. When the database accesses these areas, it uses a serialization device called a**[**latch**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-B1C1EE03-1E9E-45F9-BD12-3B05654F7615)**on required objects so that their definition does not change. Latch contention increases statement execution time and decreases concurrency.**
* **Soft parse**

**A**[**soft parse**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-BA91B10B-FBA3-4DF6-B59B-9AA57C683D33)**is any parse that is not a hard parse. If the submitted statement is the same as a reusable SQL statement in the shared pool, then Oracle Database reuses the existing code. This reuse of code is also called a**[**library cache hit**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-22628BDA-52D1-4349-827A-071163A13D1A)**.**

**Soft parses can vary in how much work they perform. For example, configuring the session shared SQL area can sometimes reduce the amount of latching in the soft parses, making them "softer."**

**In general, a soft parse is preferable to a hard parse because the database skips the optimization and row source generation steps, proceeding straight to execution.**

**SQL Optimization**

**During the optimization stage, Oracle Database must perform a hard parse at least once for every unique DML statement and performs the optimization during this parse.**

**The database never optimizes DDL unless it includes a DML component such as a subquery that requires optimization.**

**SQL Row Source Generation**

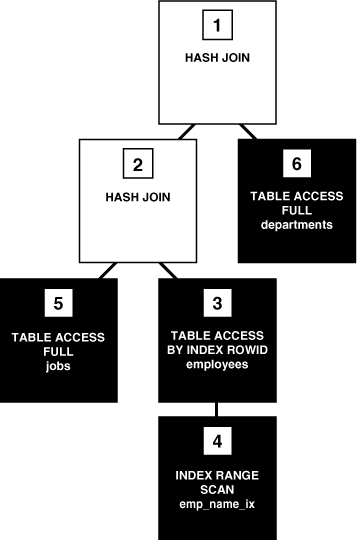
**The row source generator is software that receives the optimal execution plan from the optimizer and produces an iterative execution plan that is usable by the rest of the database.**

**The iterative plan is a binary program that, when executed by the SQL engine, produces the result set. The plan takes the form of a combination of steps. Each step returns a**[**row set**](https://docs.oracle.com/database/121/TGSQL/glossary.htm#GUID-88998F10-D77C-43B5-B148-E649BF366DC2)**. The next step either uses the rows in this set, or the last step returns the rows to the application issuing the SQL statement.**

**SQL Execution**

**During execution, the SQL engine executes each row source in the tree produced by the row source generator. This step is the only mandatory step in DML processing.**

***Row Source Tree***

****

**How does an Optimizer works? (SQL)?**

**Purpose**

**The optimizer attempts to generate the best execution plan for a SQL statement. The best execution plan is defined as the plan with the lowest cost among all considered candidate plans. The cost computation accounts for factors of query execution such as I/O, CPU, and communication.**

**It works based on two process, to execute the query faster in a efficient way**

**1. Cost-Based Optimization**

**SQL is a nonprocedural language, so the optimizer is free to merge, reorganize, and process in any order.**

**The database optimizes each SQL statement based on statistics collected about the accessed data. When generating execution plans, the optimizer considers different access paths and join methods. Factors considered by the optimizer include**

* **System resources, which includes I/O, CPU, and memory**
* **Number of rows returned**
* **Size of the initial data sets**

**2. Execution Plans**

**The plans shows the combination of the steps Oracle Database uses to execute a SQL statement. Each step either retrieves rows of data physically from the database or prepares them for the user issuing the statement.**

**An execution plan displays the cost of the entire plan, The cost is an internal unit that the execution plan only displays to allow for plan comparisons. Thus, you cannot tune or change the cost value.**

1. **Question1. What Is Pl/sql?**

**Answer :**

PL/SQL is Oracle's Procedural Language extension to SQL. The language includes object oriented programming techniques such as encapsulation, function overloading, information hiding (all but inheritance), and so, brings state-of-the-art programming to the Oracle database server and a variety of Oracle tools.

1. **Question2. Is There A Pl/sql Engine In Sql\*plus?**

**Answer :**

No. Unlike Oracle Forms, SQL\*Plus does not have a PL/SQL engine. Thus, all your PL/SQL are send directly to the database engine for execution. This makes it much more efficient as SQL statements are not stripped off and send to the database individually.

1. **Question3. Is There A Limit On The Size Of A Pl/sql Block?**

**Answer :**

Currently, the maximum parsed/compiled size of a PL/SQL block is 64K and the maximum code size is 100K. You can run the following select statement to query the size of an existing package or procedure.

SQL> select \* from dba\_object\_size where name = 'procedure\_name'

1. **Question4. Can One Read/write Files From Pl/sql?**

**Answer :**

Included in Oracle 7.3 is a UTL\_FILE package that can read and write files. The directory you intend writing to has to be in your INIT.ORA file (see UTL\_FILE\_DIR=... parameter). Before Oracle 7.3 the only means of writing a file was to use DBMS\_OUTPUT with the SQL\*Plus SPOOL command.

DECLARE

fileHandler UTL\_FILE.FILE\_TYPE;

BEGIN

fileHandler := UTL\_FILE.FOPEN('/home/oracle/tmp','myoutput','W');

UTL\_FILE.PUTF(fileHandler, 'Value of func1 is %sn',func1(1));

UTL\_FILE.FCLOSE(fileHandler);

END;

1. **Question5. What Is The Use Of Cascade Constraints?**

**Answer :**

When this clause is used with the DROP command, a parent table can be dropped even when a child table exists.

1. **Question6. Which Command Displays The Sql Command In The Sql Buffer, And Then Executes It?**

**Answer :**

RUN.

1. **Question7. What Command Is Used To Get Back The Privileges Offered By The Grant Command?**

**Answer :**

REVOKE.

1. **Question8. What Operator Performs Pattern Matching?**

**Answer :**

LIKE operator.

1. **Question9. What Operator Tests Column For The Absence Of Data?**

**Answer :**

IS NULL operator.

1. **Question10. Which Function Is Used To Find The Largest Integer Less Than Or Equal To A Specific Value?**

**Answer :**

FLOOR.

1. **Question11. Describe The Use Of %rowtype And %type In Pl/sql?**

**Answer :**

%ROWTYPE allows you to associate a variable with an entire table row. The %TYPE associates a variable with a single column type.

1. **Question12. What Packages (if Any) Has Oracle Provided For Use By Developers?**

**Answer :**

Oracle provides the DBMS\_ series of packages. There are many which developers should be aware of such as DBMS\_SQL, DBMS\_PIPE, DBMS\_TRANSACTION, DBMS\_LOCK, DBMS\_ALERT, DBMS\_OUTPUT, DBMS\_JOB, DBMS\_UTILITY, DBMS\_DDL, UTL\_FILE.

1. **Question13. Describe The Use Of Pl/sql Tables ?**

**Answer :**

PL/SQL tables help you move bulk data. They can store columns or rows of Oracle data, and they can be passed as parameters. So, PL/SQL tables make it easy to move collections of data into and out of database tables or between client-side applications and stored subprograms. You can even use PL/SQL tables of records to simulate local database tables.PL/SQL tables are scalar arrays that can be referenced by a binary integer. They can be used to hold values for use in later queries or calculations. In Oracle 8 they will be able to be of the %ROWTYPE designation, or RECORD.

1. **Question14. When Is A Declare Statement Needed ?**

**Answer :**

The DECLARE statement is used in PL/SQL anonymous blocks such as with stand alone, non-stored PL/SQL procedures. It must come first in a PL/SQL stand alone file if it is used.

1. **Question15. In What Order Should A Open/fetch/loop Set Of Commands In A Pl/sql Block Be Implemented If You Use The %notfound Cursor Variable In The Exit When Statement? Why?**

**Answer :**

OPEN then FETCH then LOOP followed by the exit when. If not specified in this order will result in the final return being done twice because of the way the %NOTFOUND is handled by PL/SQL.

1. **Question16. What Are Sqlcode And Sqlerrm And Why Are They Important For Pl/sql Developers?**

**Answer :**

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.

1. **Question17. How Can You Find Within A Pl/sql Block, If A Cursor Is Open?**

**Answer :**

Use the %ISOPEN cursor status variable.

1. **Question18. How Can You Generate Debugging Output From Pl/sql?**

**Answer :**

Use the DBMS\_OUTPUT package. Another possible method is to just use the SHOW ERROR command, but this only shows errors. The DBMS\_OUTPUT package can be used to show intermediate results from loops and the status of variables as the procedure is executed. The new package UTL\_FILE can also be used.

1. **Question19. What Are The Types Of Triggers?**

**Answer :**

There are 12 types of triggers in PL/SQL that consist of combinations of the BEFORE, AFTER, ROW, TABLE, INSERT, UPDATE, DELETE and ALL key words:

BEFORE ALL ROW INSERT

AFTER ALL ROW INSERT

BEFORE INSERT

AFTER INSERT etc.

1. **Question20. Explain The Difference Between A Hot Backup And A Cold Backup And The Benefits Associated With Each ?**

**Answer :**

A hot backup is basically taking a backup of the database while it is still up and running and it must be in archive log mode. A cold backup is taking a backup of the database while it is shut down and does not require being in archive log mode. The benefit of taking a hot backup is that the database is still available for use while the backup is occurring and you can recover the database to any point in time. The benefit of taking a cold backup is that it is typically easier to administer the backup and recovery process. In addition, since you are taking cold backups the database does not require being in archive log mode and thus there will be a slight performance gain as the database is not cutting archive logs to disk.

1. **Question21. You Have Just Had To Restore From Backup And Do Not Have Any Control Files. How Would You Go About Bringing Up This Database?**

**Answer :**

I would create a text based backup control file, stipulating where on disk all the data files where and then issue the recover command with the using backup control file clause.

1. **Question22. How Do You Switch From An Init.ora File To A Spfile?**

**Answer :**

Issue the create spfile from pfile command.

1. **Question23. Explain The Difference Between A Data Block, An Extent And A Segment ?**

**Answer :**

A data block is the smallest unit of logical storage for a database object. As objects grow they take chunks of additional storage that are composed of contiguous data blocks. These groupings of contiguous data blocks are called extents. All the extents that an object takes when grouped together are considered the segment of the database object.

1. **Question24. Where Would You Look For Errors From The Database Engine?**

**Answer :**

In the alert log.

1. **Question25. Compare And Contrast Truncate And Delete For A Table ?**

**Answer :**

Both the truncate and delete command have the desired outcome of getting rid of all the rows in a table. The difference between the two is that the truncate command is a DDL operation and just moves the high water mark and produces a now rollback. The delete command, on the other hand, is a DML operation, which will produce a rollback and thus take longer to complete.

**High-water Mark**   
  
This is a term used with table segments stored in the database. If you envision a table, for example, as a 'flat' structure or as a series of blocks laid one after the other in a line from left to right, the high-water mark (HWM) would be the rightmost block that ever contained data, as illustrated in Figure 10-1. 

+---- high water mark of newly created table

|

V

+--------------------------------------------------------+

| | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | |

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

high water mark after inserting 10,000 rows

|

v

+--------------------------------------------------------+

|x |x |x |x |x |x |x |x |x |x |x |x | | | | | | | |

|x |x |x |x |x |x |x |x |x |x |x |x | | | | | | | |

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

high water mark after inserting 10,000 rows

|

v

+--------------------------------------------------------+

|x |x |x |x |x |x |x | | | | | | | | | | | | |

|x |x |x |x |x |x |x | | | | | | | | | | | | |

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

Figure 10-1. Depiction of an HWM   
  
Figure 10-1 shows that the HWM starts at the first block of a newly created table. As data is placed into the table over time and more blocks get used, the HWM rises. If we delete some (or even all) of the rows in the table, we might have many blocks that no longer contain data, but they are still under the HWM, and they will remain under the HWM until the object is rebuilt, truncated, or shrunk (shrinking of a segment is a new Oracle 10g feature that is supported only if the segment is in an ASSM tablespace).   
  
The HWM is relevant since Oracle will scan all blocks under the HWM, even when they contain no data, during a full scan. This will impact the performance of a full scan¿especially if most of the blocks under the HWM are empty. To see this, just create a table with 1,000,000 rows (or create any table with a large number of rows), and then execute a SELECT COUNT(\*) from this table. Now, DELETE every row in it and you will find that the SELECT COUNT(\*) takes just as long (or longer, if you need to clean out the block! Refer to the 'Block Cleanout' section of Chapter 9) to count 0 rows as it did to count 1,000,000. This is because Oracle is busy reading all of the blocks below the HWM to see if they contain data. You should compare this to what happens if you used TRUNCATE on the table instead of deleting each individual row. TRUNCATE will reset the HWM of a table back to 'zero' and will truncate the associated indexes on the table as well. If you plan on deleting every row in a table, TRUNCATE¿if it can be used¿would be the method of choice for this reason.

1. **Question26. Give The Reasoning Behind Using An Index ?**

**Answer :**

Faster access to data blocks in a table.

1. **Question27. Give The Two Types Of Tables Involved In Producing A Star Schema And The Type Of Data They Hold ?**

**Answer :**

Fact tables and dimension tables. A fact table contains measurements while dimension tables will contain data that will help describe the fact tables.

1. **Question28. What Type Of Index Should You Use On A Fact Table ?**

**Answer :**

A Bitmap index.

1. **Question29. Give Two Examples Of Referential Integrity Constraints ?**

**Answer :**

primary key and a foreign key.

1. **Question30. A Table Is Classified As A Parent Table And You Want To Drop And Re-create It. How Would You Do This Without Affecting The Children Tables?**

**Answer :**

Disable the foreign key constraint to the parent, drop the table, re-create the table, enable the foreign key constraint.

1. **Question31. How Would You Determine The Time Zone Under Which A Database Was Operating?**

**Answer :**

select DBTIMEZONE from dual;

1. **Question32. Explain The Use Of Setting Global\_names Equal To True ?**

**Answer :**

Setting GLOBAL\_NAMES dictates how you might connect to a database. This variable is either TRUE or FALSE and if it is set to TRUE it enforces database links to have the same name as the remote database to which they are linking.

1. **Question33. What Command Would You Use To Encrypt A Pl/sql Application?**

**Answer :**

WRAP

1. **Question34. Explain The Difference Between A Function, Procedure And Package ?**

**Answer :**

A function and procedure are the same in that they are intended to be a collection of PL/SQL code that carries a single task. While a procedure does not have to return any values to the calling application, a function will return a single value. A package on the other hand is a collection of functions and procedures that are grouped together based on their commonality to a business function or application.

1. **Question35. Explain The Use Of Table Functions ?**

**Answer :**

Table functions are designed to return a set of rows through PL/SQL logic but are intended to be used as a normal table or view in a SQL statement. They are also used to pipeline information in an ETL process.

1. **Question36. Name Three Advisory Statistics You Can Collect ?**

**Answer :**

Buffer Cache Advice, Segment Level Statistics, & Timed Statistics.

1. **Question37. Where In The Oracle Directory Tree Structure Are Audit Traces Placed?**

**Answer :**

In unix $ORACLE\_HOME/rdbms/audit, in Windows the event viewer.

1. **Question38. Explain Materialized Views And How They Are Used ?**

**Answer :**

Materialized views are objects that are reduced sets of information that have been summarized, grouped, or aggregated from base tables. They are typically used in data warehouse or decision support systems.

1. **Question39. When A User Process Fails, What Background Process Cleans Up After It?**

**Answer :**

PMON

1. **Question40. What Background Process Refreshes Materialized Views?**

**Answer :**

The Job Queue Processes.

1. **Question41. How Would You Determine What Sessions Are Connected And What Resources They Are Waiting For?**

**Answer :**

Use of V$SESSION and V$SESSION\_WAIT

1. **Question42. How Would You Force A Log Switch ?**

**Answer :**

ALTER SYSTEM SWITCH LOGFILE;

1. **Question43. Give Two Methods You Could Use To Determine What Ddl Changes Have Been Made ?**

**Answer :**

You could use Log miner or Streams.

1. **Question44. What Does Coalescing A Table Space Do?**

**Answer :**

Coalescing is only valid for dictionary-managed table spaces and de-fragments space by combining neighboring free extents into large single extents.

1. **Question45. What Is The Difference Between A Temporary Tablespace And A Permanent Tablespace?**

**Answer :**

A temporary tablespace is used for temporary objects such as sort structures while permanent tablespaces are used to store those objects meant to be used as the true objects of the database.

1. **Question46. Name A Tablespace Automatically Created When You Create A Database ?**

**Answer :**

The SYSTEM tablespace.

1. **Question47. When Creating A User, What Permissions Must You Grants To Allow Them To Connect To The Database?**

**Answer :**

Grant the CONNECT to the user.

1. **Question48. How Do You Add A Data File To A Tablespace?**

**Answer :**

ALTER TABLESPACE ADD DATAFILE SIZE ;

1. **Question49. How Do You Resize A Data File?**

**Answer :**

ALTER DATABASE DATAFILE RESIZE ;

1. **Question50. What View Would You Use To Look At The Size Of A Data File?**

**Answer :**

DBA\_DATA\_FILES

1. **Question51. What View Would You Use To Determine Free Space In A Tablespace?**

**Answer :**

DBA\_FREE\_SPACE

1. **Question52. How Would You Determine Who Has Added A Row To A Table?**

**Answer :**

Turn on fine grain auditing for the table.

1. **Question53. How Can You Rebuild An Index?**

**Answer :**

ALTER INDEX REBUILD;

1. **Question54. Explain What Partitioning Is And What Its Benefit Is ?**

**Answer :**

Partitioning is a method of taking large tables and indexes and splitting them into smaller, more manageable pieces.

1. **Question55. You Have Just Compiled A Pl/sql Package But Got Errors, How Would You View The Errors?**

**Answer :**

SHOW ERRORS;

1. **Question56. How Can You Gather Statistics On A Table?**

**Answer :**

The ANALYZE command.

1. **Question57. How Can You Enable A Trace For A Session?**

**Answer :**

Use the DBMS\_SESSION.SET\_SQL\_TRACE or   
Use ALTER SESSION SET SQL\_TRACE = TRUE;

1. **Question58. What Is The Difference Between The Sql\*loader And Import Utilities?**

**Answer :**

These two Oracle utilities are used for loading data into the database. The difference is that the import utility relies on the data being produced by another Oracle utility EXPORT while the SQL\*Loader utility allows data to be loaded that has been produced by other utilities from different data sources just so long as it conforms to ASCII formatted or delimited files.

1. **Question59. Name Two Files Used For Network Connection To A Database ?**

**Answer :**

TNSNAMES.ORA and SQLNET.ORA

1. **Question60. Explain The Difference Between Archivelog Mode And Noarchivelog Mode And The Benefits And Disadvantages To Each ?**

**Answer :**

***ARCHIVELOG***mode is a mode that you can put the database in for creating a backup of all transactions that have occurred in the database so that you can recover to any point in time

***NOARCHIVELOG*** mode is basically the absence of ARCHIVELOG mode and has the disadvantage of not being able to recover to any point in time. NOARCHIVELOG mode does have the advantage of not having to write transactions to an archive log and thus increases the performance of the database slightly.

1. **Question61. What Command Would You Use To Create A Backup Control File?**

**Answer :**

Alter database backup control file to trace.

1. **Question62. Give The Stages Of Instance Startup To A Usable State Where Normal Users May Access It?**

**Answer :**

* + ***STARTUP NOMOUNT*** - Instance startup
  + ***STARTUP MOUNT*** - The database is mounted
  + ***STARTUP OPEN*** - The database is opened

1. **Question63. What Column Differentiates The V$ Views To The Gv$ Views And How?**

**Answer :**

The INST\_ID column which indicates the instance in a RAC environment the information came from.

1. **Question64. How Would You Go About Increasing The Buffer Cache Hit Ratio?**

**Answer :**

Use the buffer cache advisory over a given workload and then query the v$db\_cache\_advice table. If a change was necessary then I would use the alter system set db\_cache\_size command.

1. **Question65. Explain An Ora-01555 ?**

**Answer :**

You get this error when you get a snapshot too old within rollback. It can usually be solved by increasing the undo retention or increasing the size of rollbacks. You should also look at the logic involved in the application getting the error message.

1. **Question66. Explain The Difference Between $oracle\_home And $oracle\_base ?**

**Answer :**

ORACLE\_BASE is the root directory for oracle. ORACLE\_HOME located beneath ORACLE\_BASE is where the oracle products reside.

1. **Question67. What Is A Mutating Table Error And How Can You Get Around It?**

**Answer :**

This happens with triggers. It occurs because the trigger is trying to update a row it is currently using. The usual fix involves either use of views or temporary tables so the database is selecting from one while updating the other.

1. **Question68. How Do You Generate File Output From Sql?**

**Answer :**

By use of the SPOOL command

1. **Question69. How Do You Prevent Oracle From Giving You Informational Messages During And After A Sql Statement Execution?**

**Answer :**

The SET options FEEDBACK and VERIFY can be set to OFF.

1. **Question70. How Do You Prevent Output From Coming To The Screen?**

**Answer :**

The SET option TERMOUT controls output to the screen. Setting TERMOUT OFF turns off screen output. This option can be shortened to TERM.

1. **Question71. What Is The Difference Between Sql And Pl/sql?**

**Answer :**

***Structured Query Language (SQL)*** is a non-procedural language that interacts with the database, and is used for database manipulation using the Data Definition Language (DDL) and Data Manipulation Language (DML) statements. Control statements cannot be used in SQL, which is compiled and executed statement by statement at the runtime (late-binding).  
  
***PL/SQL*** is a programming language that allows the usage of Cursor Control statements and Transaction Control statements, such as if...then...else. It integrates with SQL functions and statements to interact and manipulate with the database. Unlike SQL, PL/SQL statements are compiled and processed as a block of code into the machine-readable code, which is executed at runtime (early binding); and therefore, improves the performance.

1. **Question72. What Are The Advantages Of Pl/sql?**

**Answer :**

PL/SQL is a transaction processing procedural language that has the following advantages:

* + ***Integration with databas***e —PL/SQL supports SQL statements, thus enabling integration between procedural language capabilities and the database technology.
  + ***Better performance*** —SQL statements are grouped within a PL/SQL block and sent together to database for parsing; therefore, it reduces the network traffic and increases the performance.
  + ***Higher productivity*** —-PL/SQL helps the package reusable code within the well-defined modular blocks; and therefore, increases the productivity.
  + ***Portability*** —Programs packages and libraries written in PL/SQL can be reused in different environments.

1. **Question73. What Are The Mandatory Keywords In A Pl/sql Program Block?**

**Answer :**

The BEGIN and END keywords are mandatory for any PL/SQL block.

1. **Question74. Can Arrays Be Used Within A Pl/sql Program Block?**

**Answer :**

Yes, arrays can be defined and used in a PL/SQL block using the composite datatypes, such as RECORD, TABLE, NESTED TABLE, and VARRAY.

1. **Question75. Can A Variable Be Assigned A Value From Database? If Yes, Then How?**

**Answer :**

Yes, a variable can be assigned a value from a database. You can fetch the value from the database and assign it to the variable using the into keyword, as shown in the following code:

select emp\_sal \* 0.20 into var\_increment from

t\_emolovee where emo\_no = emo\_id

1. **Question76. Does A Not Null Constraint Of A Database Column Apply To %type Variables ?**

**Answer :**

No, a NOT NULL constraint does not apply to variables declared using the %TYPE attribute. These variables can be assigned a null value in a PL/SQL block, as shown in the following code snippet:

DECLARE

var\_emp\_code t\_employee.emp\_codeflPE;

BEGIN

var\_emp\_code NULL;

1. **Question77. Can Sql Group Functions Be Used Within The Procedural Statements?**

**Answer :**

The SQL group functions, such as AVG, MIN, MAX, COUNT, SUM, STDDEV, and VARIANCE, cannot be directly used within PL/SQL. However, they can be embedded within SQL statements and then can be utilized in the procedural statements.

1. **Question78. What Is The Maximum Length Of An Identifier That Can Be Used?**

**Answer :**

The maximum length of an identifier in a PL/SQL block is 30.

1. **Question79. What Is The Equivalent Of Decode Function In The Procedural Statements?**

**Answer :**

The if...then...else statement is the equivalent of DECODE function within the procedural statements.

1. **Question80. Can Ddl Commands Be Used Within A Procedure? If So, Then How?**

**Answer :**

DDL commands, such as create, alter, revoke, and grant, cannot be used directly within the procedural language block. They can only be called using built-in Native Dynamic SQL or Dynamic SQL Package, which allows the execution of the DDL statements at runtime.

1. **Question81. What Is An Anonymous Block And How Is It Executed?**

**Answer :**

Anonymous blocks, as the name suggests, are PL/SQL blocks that are not given any name and cannot be stored in the database. These blocks can call another blocks but they cannot be called by any other block, as they do not have a name. An anonymous block is executed by either storing the block code in a file or writing the block code at the SQL prompt.

1. **Question82. What Is A Bind Variable And How Is It Used?**

**Answer :**

A bind variable is a variable within a SQL statement that has to be replaced by a valid value on the successful execution of the statement. This value can be referenced into the PL/SQL block. Bind variable is always preceded by a colon (:), as shown in the following code:

Declare

y\_emp\_age number;

Begin

Select emp-age into :v\_empLage from t\_employee

where emp\_code.121;

End;

In the preceding code, v\_emp\_age is a bind variable.

1. **Question83. How Is Global Variable Declared Within A Nested Block?**

**Answer :**

Any variable declared within a block is local to the block and global to the nested blocks. The variables declared within the nested blocks are unknown to the outer blocks. They override the references to any outer declared variables with the same name unless they are used with the block label name.

1. **Question84. What Is The Difference Between Implicit Cursor And Explicit Cursor?**

**Answer :**

A cursor is a SQL memory work area. Any SQL statement always uses a cursor for the execution of statements and fetching of results. This is implicitly defined and is called implicit cursor.  
When a programmer defines a cursor for fetching and parsing results of a SQL query, which returns more than one row, it is called an explicit cursor.

1. **Question85. Why Does Sql%isopen Always Returns The False Value?**

**Answer :**

SQL is the name of the implicit cursor, which is always opened when the database executes the SQL statement. SQL%ISOPEN always evaluates to FALSE as ORACLE closes the implicit cursor as soon as it executes the query.

1. **Question86. What Are Open, Fetch, And Close Cursor Commands? Are They Used With The Implicit Or Explicit Cursor?**

**Answer :**

These commands are used with explicit cursors. First, the cursor is opened to make it active using the OPEN command. Then, the FETCH command is used to retrieve the rows from the active cursor. After all the rows are retrieved, the cursor is closed using the CLOSE command that can again be reopened. Any other operation on a closed cursor returns the predefined exception, INVALID\_CURSOR. However, these commands cannot be used to control the SQL implicit cursors.

1. **Question87. What Are Transaction Control Statements? Can They Be Used Within The Pl/sql Block?**

**Answer :**

Transaction Control statements are the COMMIT and REVOKE commands that control the logic of transactions within a database. These statements are valid within a PL/SQL block. The COMMIT command terminates the active transaction and makes the changes permanent to the database. The ROLLBACK command terminates the active transaction but cancels any changes that were made to the database.

1. **Question88. What Is A Savepoint Command?**

**Answer :**

The SAVEPOINT command is used to set a point within a transaction to which you may rollback. This command helps in cancelling the portions of the current transaction. Using ROLLBACK with the SAVEPOINT TO clause, a transaction can be undone in parts rather than rolling back the entire transaction.

1. **Question89. What Are Pseudo-columns? How Can You Use Pseudo Columns In Procedural Statements?**

**Answer :**

Pseudo columns are not actual columns in a table but they behave like columns. They are used in SQL statements to retrieve specific information. PL/SQL recognizes pseudo-columns as a part of SQL statements but they cannot be used directly in a procedural language. The following pseudo-columns are used in SQL:

* + ROWID
  + ROWNUM
  + LEVEL
  + CURRVAL
  + NEXTVAL

1. **Question90. How Can You Check If An Update Statement In Pl/sql Is Successfully Executed?**

**Answer :**

You can use the SQL%NOTFOUND attribute to check if the UPDATE statement has successfully updated any rows or not. This attribute returns the TRUE value if the last executed SQL statement has not affected any rows.

1. **Question91. What Is A Record Datatype?**

**Answer :**

A RECORD is a collection of data items, which differ from each in other in datatype, but are logically related. For example, an employee record with name, age, address, and telephone number. These are all part of employee record but have different datatypes.  
In PL/SQL, the record type is defined first and then the record of that type is declared as follows:

TYPE emp\_record\_tyPe is RECORD

(emp\_name varchar2(50),

emp\_age number(2),

emp\_address varchar2(100).

emp\_tel\_no number(10));

emp\_record emp\_record\_type;

We have defined the record type by the name emp\_record\_type and then declared a record, emp\_record, of that type.

1. **Question92. What Is The Function Of Open Cursor Command?**

**Answer :**

The following operations occur when the OPEN cursor command is executed for a cursor:

* + Memory area for processing is allocated
  + The SELECT statement is parsed
  + The input variables are assigned the values by obtaining the memory addresses
  + The active set of rows that fulfill the select criteria are identified
  + The pointer is set just before the first row of the active set

1. **Question93. What Is The Advantage Of Using The %rowtype Datatype?**

**Answer :**

The advantages of using the %ROWTYPE datatype to declare variables are given as follows:

* + It is useful to retrieve an entire row from a table. If you do not use the %ROWTYPE datatype, then you have to declare variables for each column separately.
  + It can be used even if datatype of the table columns is not known
  + It ensures that datatype of the variable changes dynamically if the underlying table is altered.

1. **Question94. How Can Explicit Cursors Be Used To Fetch Multiple Rows?**

**Answer :**

A loop must be defined to fetch multiple rows from an explicit cursor. It will fetch one row from the active cursor set on each iteration of the loop, as shown in the following code:

Declare

v\_emp\_code

t\_employees.emp\_code%TYPE;

v\_emp\_name t\_employees.emp\_name%TYPE;

v\_emp\_salary

t\_employees.emp\_salary%TWE:

v\_increment number (10.2);

Cursor emp\_cursor IS

Select emp\_code, emp\_name,

emp\_salary from t\_employees:

Begin

Open emp\_cursor;

LOOP FETCH emp\_cursor into v\_emp\_code,

v\_emp\_name, v\_emp\_salary:

EXIT WHEN emp\_cursoMBOutOuNT > 10 or

emp\_cursor%NOTPOuND or emp\_cursor%NOTFOuND IS NULL;

V\_increment := v\_emp\_salary • 0.20 DBMS\_OUTPUT.

PUTLINE(v\_emp\_code II' 'Ilv\_emp\_name It " II v\_increment);

END LOOP;

CLOSE emp\_cursor;

END;

1. **Question95. Can You Define A Record Type Based On An Explicit Cursor?**

**Answer :**

Yes, It is possible to define a record type based on the selected columns of an explicit cursor. This helps in retrieving the entire row together into the defined record and further processing. The values of the row are directly loaded into the fields of the record.

1. **Question96. What Is A Cursor For Loop And How Does It Differ From An Explicit Cursor?**

**Answer :**

Cursor FOR loop is a shortcut to use explicit cursors. It does not require explicit opening, fetching, and closing of the cursor. In other words, the cursor FOR loop implicitly opens the cursor, fetches the record, and closes the cursor. In the cursor FOR loop, the cursor is defined and its attributes are checked within the processing of the FOR loop. The records are also implicitly defined within the loop, as shown in the following code:

Declare

v\_increment number(10.2);

Cursor em P\_cursor IS

Select emp\_code. emp\_name. emp\_salary from t\_employeeS;

Begin

FOR emp\_record IN em p\_cursor

v\_increment emp\_record.emp\_salary • 0.20

D13MS\_OUTPUT.PUTLINE

(emp\_FREOFthent\_COOR 11 'Ilemp\_record.emp\_name II " II v\_increment);

END LOOP;

END;

It is also possible to use a cursor FOR loop with a subquery, where the cursor name is not defined. In this case, the attributes of the cursor cannot be checked as no explicit cursor name is defined.

1. **Question97. What Is Select For Update And How Can It Be Used In A Cursor?**

**Answer :**

The SELECT FOR UPDATE clause is used to lock rows in a table when an update or delete transaction is performed on the active set. It selects the rows of the table that have to be modified and locks each row in the result set. This prevents any other user from accessing the same set of records while modification is being executed on the table. You should add the NOWAIT clause to the SELECT FOR UPDATE statement to prevent the procedure from waiting indefinitely in case the lock cannot be acquired. If the NOWAIT clause is used, the control will return to the program in case the requested rows have been locked by another user.

1. **Question98. How Is It Possible To Fetch A Current Row From A Cursor Without Using Rowid?**

**Answer :**

The WHERE CURRENT OF clause is used to reference the current row of an active explicit cursor. When this clause is used, ROWID is not needed to access the current row for modifications.

1. **Question99. Which Datatypes Can Be Used With A Record Type?**

**Answer :**

The datatype of a REORD Type can be a NUMBER, VRCHAR2, %TYPE, or %ROWTYPE; however, it cannot be a'REF CURSOR.

1. **Question100. Is It Possible To Label A Loop? If Yes, Then What Is Its Benefit?**

**Answer :**

Yes, It is possible to label a loop similar to a PL/SQL block. Loop labels are undeclared identifiers enclosed within the double angle brackets. This helps in better readability in case of nested loops. In addition, this helps in using the EXIT statement with the loop label in the nested loops. Label of a loop should appear before the LOOP statement, as shown in the following code snippet:

«first»

LOOP

-----

«second»

LOOP

-------

EXIT second WHEN ------

END LOOP second;

END LOOP first;

1. **Question101. In A For-loop, Can The Counter Value Be Assigned A Value Within The Loop?**

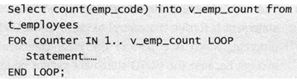
**Answer :**

The counter of a FOR-LOOP can only be used as a constant within the loop. It cannot be assigned a value inside the loop, as that would change the bounds of the loop, which is logically not possible.

1. **Question102. Is It Possible To Assign A Dynamic Value To A Loop Counter?**

**Answer :**

Yes, it is possible to assign a dynamic value to the loop counter at the time of execution, as shown in the following code snippet:

  
In the preceding code snippet, the employee count is derived from the table at runtime and the counter is used to iterate the FOR-LOOP.

1. **Question103. What Will Be The Consequence If The Upper Bound Of The For-loop Counter Gets A Smaller Value Than The Lower Bound?**

**Answer :**

If the upper bound of a FOR-LOOP is assigned a value smaller than the lower bound of the loop, then the statements within the loop are not executed and the control is passed to the statement after the loop.

1. **Question104. Can Index-by Table Be Stored In A Database? If Not, Then Which Type Of Collection Can Be Used To Store In A Database?**

**Answer :**

No, Index-By table cannot be stored in a database. The nested tables and VARRAYs collection types can be used to store in database.

1. **Question105. What Is The Difference Between Nested Tables And Varrays?**

**Answer :**

Both nested tables and VARRAYs are collection types that can be stored in the database. They differ in two properties, bounding and sparsity. Bounding refers to the limit on the number of elements that a collection can have; whereas, sparsity means whether there can be gaps within the subscripts or not.

Nested tables are not bounded and can be sparse; whereas, VARRAYs are bounded and cannot be sparse. It means that VARRAYs have a limit on the number of elements and are bounded; whereas, nested tables does not have a limit on the number of elements and are unbounded. Nested tables can become sparse if elements are deleted; whereas, VARRAYs can never become sparse.

1. **Question106. What Is The Difference In The Usage Of Select Into And Fetch?**

**Answer :**

The SELECT INTO clause should be used when the result of the query is known to return a single row. When more than one row is to be returned by the SQL query, then the FETCH clause should be used. If the FETCH clause is not used and the query returns more than one row, then the runtime error is generated in the code.

1. **Question107. How Can Two Pl/sql Records Be Compared?**

**Answer :**

Two records can be compared by using the equality (comparison) operator for each field of the record. However, the entire record cannot be compared as a whole.

1. **Question108. Which Datatype Is Not Allowed In The Definition Of Pl/sql Record?**

**Answer :**

A collection type cannot be used for defining the datatype of fields; only scalar datatype can be used for a PL/SQL record.

1. **Question109. What Is The Value Of Sql%found Attribute After And Before An Insert Statement Is Executed?**

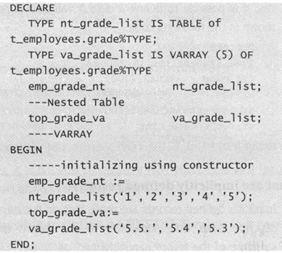
**Answer :**

The value of SQL%F0UND attribute will be NULL before an INSERT statement is executed and TRUE after the successful insertion into a table.

1. **Question110. What Is A Constructor And Why Is It Used?**

**Answer :**

A constructor is a special built-in function that is used to initialize nested tables and VARRAYs. Constructor has the same name as the collection and can take any number of arguments, as shown in the following code:



1. **Question111. How Are Elements Added To An Index-by Table?**

**Answer :**

Index-By tables are instantiated automatically. No special constructor is required to initialize an Index-By table. Whenever an element of the collection is referred, it is initialized.

1. **Question112. Can An If-then-else If Block Have Infinite Number Of Else Clauses?**

**Answer :**

No, the IF-THEN-ELSIF block can have only one ELSE clause at the end of the it, which is executed if no other condition is evaluated to TRUE.

1. **Question113. What Will Be The Value Of %n0tf0und Attribute After A Cursor Is Opened But Before A Fetch Statement Is Executed?**

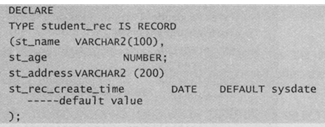
**Answer :**

The value of %NOTFOUND attribute will be NULL after the cursor is opened but before the FETCH statement is executed. After the FETCH statement, the value of %NOTFOUND attribute will be TRUE OR FALSE depending on the result of the FETCH statement.

1. **Question114. Is It Possible To Define A Default Value For A Record Field? If Yes, Then How?**

**Answer :**

Yes, it is possible to define a default value for a record field, as shown in the following code:



1. **Question115. What Are Implicitly Defined Records?**

**Answer :**

Implicitly defined records are those that do not have the need to describe each field separately in the record definition. The structure of the record is not defined using the TYPE statement; instead, the %ROWTYPE attribute is used to define the implicit record to have the same structure as the database record. Following code shows the declaration of the implicit defined records:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/implicitly%20defined%20record.png

1. **Question116. What Is The Advantage Of Defining Implicit Records?**

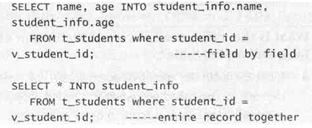
**Answer :**

Implicit records are useful as there is no hard-code description needed. As they are based on a database table record, any change in the database table record will be automatically reflected in the record.

1. **Question117. How Can An Implicitly Defined Record Be Populated?**

**Answer :**

The implicitly defined record can be populated field by field or the entire record can be populated in one statement, as shown in the following code:



1. **Question118. Can A Record Definition Have Another Record As Datatype?**

**Answer :**

Yes, a record definition can have scalar datatypes or another record as a datatype.

1. **Question119. Is It Possible To Define A Not Null Field In A Record?**

**Answer :**

Yes, it is possible to define a NOT NULL field in a record so that it will always require a value.

1. **Question120. Is It Possible To Define A Constant Value In A Record?**

**Answer :**

No, a CONSTANT value is not allowed in a record.

1. **Question121. Is It Possible To Copy Contents Of One Record To Another?**

**Answer :**

Yes, the contents of one record can be copied to another by copying either each field using the assignment operator or whole record using the assignment operator.

1. **Question122. Do Records Have Methods?**

**Answer :**

No, records do not have methods.

1. **Question123. Can A Varray Be Used In A Record Definition?**

**Answer :**

No, VARRAY is a collection and collections cannot be used for record definition.

1. **Question124. Can An Uninitialized Nested Table Be Referenced?**

**Answer :**

No, an uninitialized nested table cannot be referenced. It can only be checked for nullity.

1. **Question125. What Is The Difference Between Initialization Of Index-by Table And Nested Table?**

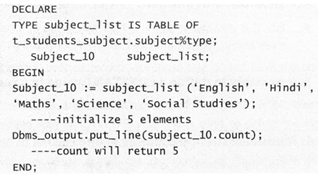
**Answer :**

Index-By table is initialized automatically by referencing an element in the collection; whereas, nested tables have to be initialized using a built-in function called constructor.

1. **Question126. Do Collections Have Methods? If Yes, Give An Example ?**

**Answer :**

Yes, collections have built-in methods, which are used to examine and manipulate the attributes of the collection, as shown in the following code:



1. **Question127. Can Collections Be Compared With Boolean Operators?**

**Answer :**

No, collections cannot be compared with Boolean operators.

1. **Question128. Can The Exists Method Be Used For Index-by Table?**

**Answer :**

Yes, the EXISTS method can be used for all collection types. It returns a Boolean value, TRUE, if the collection element is initialized and FALSE, if the collection element is not initialized.

1. **Question129. What Will The Result Of The Next() Method, If X Is The Last Element Of The Collection?**

**Answer :**

The NEXT (<x>) method will return NULL if the last element of the collection is reached.

1. **Question130. What Is The Difference In Accessing Objects Of Lob And Long Datatypes?**

**Answer :**

The LOB datatype supports random access; whereas, the LONG datatype supports only sequential access of objects.

1. **Question131. What Is The Difference Between Lob Objects Stored In-line And Out-of-line?**

**Answer :**

The LOB objects can be stored in-line or out-of-line. The in-line storage means that objects are stored along with the row; whereas, the out-of-line storage means that objects are stored outside the row.

1. **Question132. Which Package Is Used To Manipulate Lob Objects?**

**Answer :**

Oracle-supplied package called DBMS.J\_OB is used to manipulate the LOB objects.

1. **Question133. What Are Lob Locators?**

**Answer :**

LOB locators are values, which specify the location of the large object.

1. **Question134. What Are The System Privileges That Are Required By A Schema Owner (user) To Create A Trigger On A Table?**

**Answer :**

A user must be able to alter a table to create a trigger on the table. The user must own the table and either have the ALTER TABLE privilege on that table or have the ALTER ANY TABLE system privilege. In addition, the user must have the CREATE TRIGGER system privilege. User should have the CREATE ANY TRIGGER system privilege to be able to create triggers in any other user account or schema.

A database-level event trigger can be created if the user has the ADMINISTER DATABASE TRIGGER system privilege.

1. **Question135. What Are The Different Types Of Triggers?**

**Answer :**

There are following two types of triggers:

* + ***Database triggers*** are executed implicitly whenever a Data Manipulation Language (DML) statement is carried out on a database table or a Data Definition Language (DDL) statement, such as CREATE OR ALTER, is performed on the database. They may also be executed when a user or database event occurs, such as a user logs on or a database is shutdown.
  + ***Application triggers*** are executed implicitly whenever a DML event takes place within an application, such as WHEN\_NEW\_FORM\_INSTANCE in the Oracle Forms application.

1. **Question136. Which Data Manipulation Operations Raise A Trigger?**

**Answer :**

Data manipulation operations such as INSERT, UPDATE, and DELETE on a table or a view are the main triggering events that cause a trigger to execute.

1. **Question137. Can Triggers Reference Other Tables Through A Role?**

**Answer :**

Triggers may require referencing other tables while triggering an event. Therefore, privileges required to reference other tables must be granted directly to the owner of the trigger, as the trigger cannot reference other user tables through roles.

For example, consider a trigger, which is used for any audit table. To insert a row in the audit table, the owner of the trigger must have the insert privilege on that audit table.

1. **Question138. What Is The Difference Between Row-level Triggers And Statement-level Triggers?**

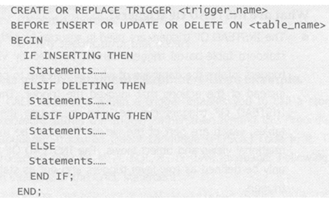
**Answer :**

* + Row-level triggers are activated once for each row, which is modified by a DML statement. For example, consider a table, STUDENT\_MARKS\_AUDIT, used for the auditing of the STUDENT\_MARKS table. A row-level trigger would insert a row in the STUDENT\_MARKS\_AUDIT table every time any value is modified in any row of the STUDENT\_MARKS table.
  + Statement-level triggers are executed only once for each DML statement. For example, if there is an INSERT statement, which inserts 100 rows in a STUDENT\_MARKS table, then the statement-level trigger would be executed only once on that table; whereas, a row-level trigger would be executed 100 times.
  + Row-level triggers are generally used for data auditing applications; whereas, statement-level triggers are used to enforce security features for the types of actions that may be performed on a table.

1. **Question139. Is It Possible To Combine Several Triggering Events Into One Trigger?**

**Answer :**

Yes, it is possible to combine various triggering events and create one trigger. Special conditional predicates such as INSERTING, UPDATING, and DELETING are used within the trigger body to represent the triggering events, as shown in the following code snippet:



1. **Question140. How Can Triggers Be Used For The Table Auditing?**

**Answer :**

Triggers can be used to track values for data operations on tables. This is done using the old and new qualifiers within the trigger code. These two clauses help keep track of the data that is being inserted, updated, or deleted in the table; and therefore, facilitate in application auditing of DML statements. The audit trail can be written to a user-defined table and audit records can be generated for both row-level and statement-level triggers.

1. **Question141. What Are Instead Of Triggers?**

**Answer :**

The INSTEAD OF triggers are used in association with views. The standard table-based triggers cannot be used by views. These triggers inform the database of what actions are to be performed instead of the actions that invoked the trigger. Therefore, the INSTEAD OF triggers can be used to update the underlying tables, which are part of the views. They can be used on both relational views and object views. The INSTEAD OF triggers can only be defined as row-level triggers and not as statement-level triggers.

1. **Question142. What Is The Difference Between Database Trigger And Stored Procedure?**

**Answer :**

* + The main difference between database trigger and stored procedure is that the trigger is invoked implicitly and stored procedure is invoked explicitly.
  + Transaction Control statements, such as COMMIT, ROLLBACK, and SAVEPOINT, are not allowed within the body of a trigger; whereas, these statements can be included in a stored procedure.

1. **Question143. How Can The Performance Of A Trigger Be Improved?**

**Answer :**

The performance of a trigger can be improved by using column names along with the UPDATE clause in the trigger. This will make the trigger fire when that particular column is updated  and therefore, prevents unnecessary action of trigger when other columns are being updated.

1. **Question144. What Are The Events On Which A Database Trigger Can Be Based?**

**Answer :**

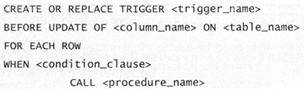
Database triggers are based on system events and can be defined at database or schema level. The various events on which a database trigger can be based are given as follows:

* + Data definition statement on a database or schema object
  + Logging off or on of a specific user
  + Database shutdown or startup
  + On any specific error that occurs

1. **Question145. What Is A Call Statement? Explain With An Example ?**

**Answer :**

A CALL statement within a trigger enables you to call a stored procedure within the trigger rather than writing the Procedural Language/Structured Query Language (PL/SQL) code in it, The procedure may be in PL/SQL, C, or Java language. Following is an example of the CALL statement:



1. **Question146. What Is A Mutating Table?**

**Answer :**

A mutating table is a table, which is in the state of transition. In other words, it is a table, which is being updated at the time of triggering action. If the trigger code queries this table, then a mutating table error occurs, which causes the trigger to view the inconsistent data.

1. **Question147. What Are Schema-level Triggers?**

**Answer :**

Schema-level triggers are created on schema-level operations, such as create table, alter table, drop table, rename, truncate, and revoke. These triggers prevent DDL statements, provide security, and monitor the DDL operations.

1. **Question148. What Is A Database Event Trigger?**

**Answer :**

Trigger that is executed when a database event, such as startup, shutdown, or error, occurs is called a database event trigger. It can be used to reference the attributes of the event and perform system maintenance functions immediately after the database startup.

1. **Question149. In What Condition Is It Good To Disable A Trigger?**

**Answer :**

It is good to disable triggers during data load operations. This improves the performance of the data loading activities. The data modification and manipulation that the trigger would have performed has to be done manually after the data loading.

1. **Question150. Which Column Of The Userjtriggers Data Dictionary View Displays The Database Event That Will Fire The Trigger?**

**Answer :**

The Description column of the USERJTRIGGERS view combines information from many columns to display the trigger header, which includes the database event.

1. **Question151. What Is The Meaning Of Disabling A Trigger?**

**Answer :**

When a trigger is disabled, it does not mean that it is deleted. The code of the trigger is still stored in the data dictionary but the trigger will not have any effect on the table.

1. **Question152. Can Triggers Stop A Dml Statement From Executing On A Table?**

**Answer :**

Yes, triggers have the capability of stopping any DML statement from execution on a table. Any logical business rule can be implemented using PL/SQL to block modification on table data.

1. **Question153. Can A View Be Mutating? If Yes, Then How?**

**Answer :**

No, a view cannot be mutating like a table. If an UPDATE statement fires an INSTEAD OF trigger on a view, the view is not considered to be mutating. If the UPDATE statement had been executed on a table, the table would have been considered as mutating.

1. **Question154. Can A Commit Statement Be Executed As Part Of A Trigger?**

**Answer :**

No, A COMMIT statement cannot be executed as a part of a trigger because it is a Transaction Control statement, which cannot be executed within a trigger body. Triggers fire within transactions and cannot include any Transaction Control statement within its code.

1. **Question155. What Is The Difference Between Alter Trigger And Drop Trigger Statements?**

**Answer :**

An ALTER TRIGGER statement is used to recompile, disable, or enable a trigger; whereas, the DROP TRIGGER statement is used to remove the trigger from the database.

1. **Question156. Do Triggers Have Restrictions On The Usage Of Large Datatypes, Such As Long And Long Raw?**

**Answer :**

Triggers have restrictions on the usage of large datatypes as they cannot declare or reference the LONG and LONG RAW datatypes and cannot use them even if they form part of the object with which the trigger is associated. Similarly, triggers cannot modify the CLOB and BLOB objects as well; however, they can reference them for read-only access.

1. **Question157. Are Ddl Triggers Fired For Ddl Statements Within A Pl/sql Code Executed Using The Dbms.sql Package?**

**Answer :**

No, DDL triggers are not executed for DDL statements, which are executed within the PL/SQL code using the DBMS\_SQL package.

1. **Question158. Does A User\_objects View Have An Entry For A Trigger?**

**Answer :**

Yes, the USER\_OBJECTS view has one row entry for each trigger in the schema.

1. **Question159. How Can You View The Errors Encountered In A Trigger?**

**Answer :**

The USERJERRORS view can be used to show all the parsing errors that occur in a trigger during the compilation until they are resolved.

1. **Question160. Does User-triggers Have Entry For Triggers With Compilation Errors?**

**Answer :**

Yes, USER\_TRIGGERS have entries for all triggers that are created in the schema with or without errors.

1. **Question161. Is It Possible To Pass Parameters To Triggers?**

**Answer :**

No, it is not possible to pass parameters to triggers. However, triggers fired by INSERT and UPDATE statements can reference new data by using the mew prefix. In addition, the triggers fired in response to UPDATE and DELETE statements can reference old, modified, or deleted data using the :old prefix.

1. **Question162. Can A Select Statement Fire A Trigger?**

**Answer :**

No, a SELECT statement cannot fire a trigger. DML statements, such as INSERT, UPDATE, and DELETE, can cause triggers to fire.

1. **Question163. Can Cursors Be Part Of A Trigger Body?**

**Answer :**

Yes, cursors can be a part of code in trigger.

1. **Question164. Is It Possible To Create Startup Or Shutdown Trigger For On-schema?**

**Answer :**

No, It is not possible to create STARTUP or SHUTDOWN triggers for ON-SCHEMA.

1. **Question165. What Does The Base\_object\_type Column Shows In The User.triggers Data Dictionary View?**

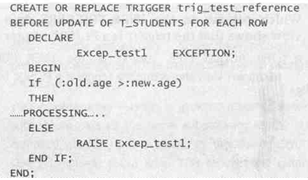
**Answer :**

The BASE\_OBJECT\_TYPE column identifies the type of database object with which the trigger is associated. It shows whether the object of database is a TABLE, VIEW, SCHEMA, or DATABASE.

1. **Question166. Is It Possible To Define An Exception Within A Trigger?**

**Answer :**

Yes, it is possible to define and use an exception within a trigger, as shown in the following code snippet:



1. **Question167. Can Instead Of Triggers Be Used To Fire Once For Each Statement On A View?**

**Answer :**

No, INSTEAD OF triggers cannot be used for each statement; however, It can only be used for each row on a view.

1. **Question168. Is It Possible To Include An Insert Statement On The Same Table To Which The Trigger Is Assigned?**

**Answer :**

If an INSERT statement is used on the same table to which the trigger is associated, then this will give rise to a mutating table, as it is not possible to change the same table that is already in the process of being changed.

1. **Question169. What Are Conditional Predicates?**

**Answer :**

Triggers use conditional predicates, such as INSERTING, UPDATING, and DELETING, to determine which particular event will cause the trigger to fire. All the three predicates have Boolean values and are useful in triggers, such as AFTER INSERT or UPDATE.

1. **Question170. Write The Alter Statement To Enable All The Triggers On The T.students Table ?**

**Answer :**

The ALTER statement is given as follows:

ALTER TABLE T\_STUDENTS ENABLE ALL TRIGGERS;

1. **Question171. Which Column In The User.triggers Data Dictionary View Shows That The Trigger Is A Pl/sql Trigger?**

**Answer :**

The ACTION\_TYPE column of the USER\_TRIGGERS data dictionary view shows that the trigger is a PL/SQL trigger.

1. **Question172. What Is A Subprogram? What Are Its Advantages?**

**Answer :**

A named Procedural Language/Structured Query Language (PL/SQL) block is called a subprogram. It has three sections: declarative, executable, and exception handling (optional). The main advantage of a subprogram is that it can be compiled and stored in a database. In addition, it has the following advantages:

* + ***Modularization*** —Refers to a property, which allows you to break the large programs into smaller modules and make the program easy to write and debug.
  + ***Easy maintenance***—Refers to a property, which allows you to make the modifications in a code because the code is modular and written in one place.
  + ***Reusability***—Refers to a property, which allows you to reuse the same subprogram within any application. The code of the subprogram need not be re-written each time.
  + ***Data integrity and security*** —Refers to a property, which prevents unauthorized users from accessing the subprogram without the proper rights.
  + ***Better performance*** —Helps in avoiding repeated parsing on subsequent calls to the same subprogram as the code is compiled, parsed, and available in the shared Structured Query Language (SQL) area. This reduces the number of calls to the database; and therefore, increases the network performance.
  + ***Code clarity***—Allows you to give proper names to the PL/SQL blocks. As a result, the code becomes simpler to understand and gets better clarity.

1. **Question173. What Is An Anonymous Block?**

**Answer :**

Any PL/SQL block, which does not have a name, is called an anonymous block. It is directly written in the application code and compiled by the PL/SQL engine at the time of execution.

1. **Question174. What Are The Two Different Types Of Subprograms In Pl/sql?**

**Answer :**

The two different types of subprograms in PL/SQL are procedures and functions. Both can be called from an external environment or application. In addition, they can accept parameters.

1. **Question175. What Is The Basic Difference Between Procedure And Function?**

**Answer :**

A***procedure*** is executed as a PL/SQL statement. It can accept more than one parameter as an input from the calling environment and may return none, one, or more than one value.  
  
A***function*** is invoked as a part of expression. It can have more than one parameter as an input from the calling environment and it should return a single value to the calling environment using the RETURN statement.

1. **Question176. What Is The Difference Between The Code Syntax Of An Anonymous Block And Named Subprogram Block?**

**Answer :**

A subprogram has two parts: subprogram specification and subprogram body.

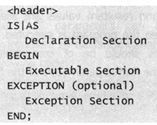
* + The subprogram specification is the header section, which contains the details such as program name, subprogram type, parameter list, and return clause (for functions).
  + The subprogram body is a PL/SQL block, which has the declaration, executable, and exception sections.

An anonymous block has no specification or header section. It only consists of a PL/SQL block in the body that contains the declaration, executable, and exception sections.

1. **Question177. Give The Basic Block Structure Of A Subprogram ?**

**Answer :**

The block structure of a subprogram is given as follows:



1. **Question178. What Is A Procedure? Can It Be Stored In The Database?**

**Answer :**

A procedure is a named PL/SQL block (subprogram) that can have more than one parameter as an input from the calling environment and it may return none, one, or more than one value.  
Yes, it can be compiled and stored in the database as a schema object and used for execution at any point of time.

1. **Question179. What Does The Replace Option Indicates In The Procedure Definition?**

**Answer :**

The REPLACE option specifies that if the procedure with the same name exists in the database, it will be dropped and recreated in the database as a new version.

1. **Question180. Can The Size Of The Parameter Passed To The Procedure Be Restricted In The Procedure Definition?**

**Answer :**

No, the size of the parameter passed to the procedure cannot be restricted in the procedure declaration. The datatype for the parameter has to be specified only.

1. **Question181. What Is The Difference Between Formal And Actual Parameters In A Subprogram Specification?**

**Answer :**

Formal parameters are variables declared as the parameter list in the subprogram specification.  
Actual parameters are the values that are passed to the subprogram from the calling environment. The actual parameter values are evaluated and resultant values are assigned to the formal parameters when the program is executed. Both formal and actual parameters should be of compatible datatypes. Following code shows the formal parameters of the function:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/actual%20and%20formal%20parameters.png  
In the preceding function, p\_student\_id and p\_percentage are the formal parameters.  
In the following function code, v\_student\_id and v\_percentage are the actual parameters:  
proc\_disp\_percentage (v\_student\_id, v\_percentage);

1. **Question182. Which Datatypes Can Be Used For Parameters Within A Procedure?**

**Answer :**

Explicit datatypes, %TYPE and %ROWTYPE, without size specification can be used for parameters in a procedure.

1. **Question183. What Are The Different Parameter Modes, Which Can Be Used In The Procedure Specification?**

**Answer :**

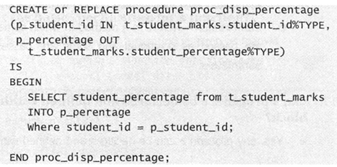
There are three parameter modes for a procedure, given as follows:

* + ***IN***—Refers to a constant value, which is assigned to this parameter from the calling environment. The IN parameter value cannot be modified within a procedure. It is a default mode.
  + ***OUT***— Refers to a return value, which is passed from the procedure back to the calling environment.
  + ***IN OUT*** —Refers to a value, which is assigned from the calling environment to this parameter and a same or different value is passed back from the procedure to the calling environment using this parameter.

1. **Question184. How Can The Out Parameters Be Viewed On The Execution Of Procedure In The Sql Plus Environment?**

**Answer :**

The OUT parameters can be viewed in the SQL PLUS environment by declaring variables for the same, as shown in the following code:

  
On the SQL PLUS prompt, variable is created and the procedure is executed, as shown in the following code:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/out%20parameter%201.png

1. **Question185. How Can A Parameter Be Initialized In A Procedure?**

**Answer :**

The IN OUT parameter mode can be used to initialize a parameter in a procedure, as any value can be initially assigned to this parameter when the procedure is called from the calling environment. The value of this parameter can be changed during the execution of the procedure and the result can be returned back to the calling environment from the procedure.

1. **Question186. Can Default Values Be Assigned To In Out Parameters In A Procedure?**

**Answer :**

No, default values cannot be assigned to IN OUT parameters; only IN parameters can be assigned default values.

1. **Question187. What Are The Different Methods For Passing Parameters To A Procedure?**

**Answer :**

There are following three different methods to pass values to a procedure:

* + ***Positional*** —Refers to a method in which actual parameters of a procedure are passed in the same order as the formal parameters.
  + ***Named***—Refers to a method in which actual parameters of a procedure are passed in any arbitrary order but are associated with their corresponding formal parameters.
  + ***Combination***—Refers to a method in which some parameters are passed as positional and some are passed as named parameters.

1. **Question188. Can A Procedure Be Declared And Defined Within A Pl/sql Block?**

**Answer :**

Yes, any procedure can be declared and defined within a PL/SQL block.

1. **Question189. What Is The Scope Of The Procedure Defined Within A Pl/sql Block?**

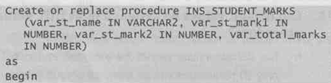
**Answer :**

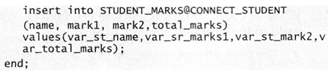
The scope of any procedure that is defined within the PL/SQL block will be limited as it can be called only within the PL/SQL block and not by any other procedure or calling environment.

1. **Question190. How Can You Reference Remote Tables In Procedures?**

**Answer :**

Remote tables can be referenced within procedures through SQL statements. A remote table can be queried through a database link in the procedure. The following code inserts a record in the STUDENT\_MARKS table, which is in another schema, using the CONNECT\_STUDENT database link:





1. **Question191. What Happens If An Exception Is Unhandled Within A Procedure?**

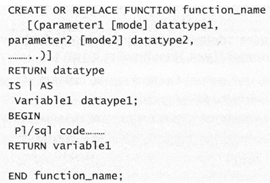
**Answer :**

If an exception is unhandled within a procedure, the control is passed back to the calling environment. Any DML statements, which were executed within the procedure, are rolled back.

1. **Question192. What Is The Basic Specification For A Function?**

**Answer :**

The basic specification for a function is given as follows:



1. **Question193. Can There Be Multiple Return Statements Within A Function?**

**Answer :**

Yes, there can be multiple RETURN statements within a function though only one is executed. After the value is retuned, the control passes back to the calling environment and function processing is stopped.

1. **Question194. Is It Possible To Define A Return Statement In The Exception Part Of A Function?**

**Answer :**

Yes, it is possible to define RETURN statement in the exception section. This RETURN statement would be executed if an exception is raised and control is passed to the exception section of the code.

1. **Question195. What Are The Different Parameter Modes, Which Can Be Used In The Function Specification?**

**Answer :**

The three different modes, IN, OUT, and IN OUT, can be used in the function specification. However, it is recommended to use only the IN mode. This is because functions are supposed to accept values from the calling environment and return a single value.

1. **Question196. Can A Function Be Defined Without A Return Statement?**

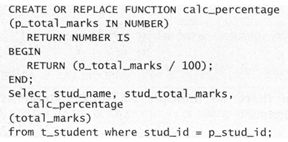
**Answer :**

No, a function cannot be defined without a RETURN statement because it has to return a value back to the calling environment. In case a RETURN statement is not defined, runtime error is raised.

1. **Question197. Can User Defined Functions Be Called Within Sql Statements? If Yes, Then How?**

**Answer :**

Yes, user defined functions can be called in SQL statements as shown in the following code:



1. **Question198. Can Functions Be Stored Within A Database As Database Objects?**

**Answer :**

Yes, functions can be stored as database objects in the database.

1. **Question199. Which Datatypes Can Be Used Within Functions That Can Be Invoked In Sql Statements?**

**Answer :**

Stored functions can use valid SQL datatypes, such as NUMBER, VARCHAR2, and DATE. However, they cannot use PL/SQL datatypes, such as BOOLEAN, RECORD, and TYPE.

1. **Question200. What Restrictions Can Be Applied On Stored Functions, Which Are Called From Sql Statements?**

**Answer :**

The main restrictions on stored functions that are called from SQL statements are:

* + These functions can use only positional methods for passing the parameters.
  + These functions can use only SQL specific datatypes. PL/SL datatypes, such as RECORD or TYPE, cannot be used.
  + User should have EXECUTE privilege on the function to use it within SQL statements
  + These functions cannot have the OUT and IN OUT parameter modes

1. **Question201. What Is The Difference Between Create Or Replace Function And Drop Function Commands?**

**Answer :**

The CREATE OR REPLACE FUNCTION command creates the function if it does not exist and replaces it with the new version if it already exists. The DROP FUNCTION command deletes the function altogether from the database. Any privilege assigned on the function remains the same when the CREATE OR REPLACE FUNCTION command is used; however, in the case of the DROP FUNCTION command, all the privileges are revoked.

1. **Question202. Can A Role Be Used To Define Privileges To Access Database Objects Of Another Schema In Procedures And Functions?**

**Answer :**

Database objects of another schema cannot be accessed in a procedure or function through roles; however, explicit privileges have to be granted to the user for those database objects.

1. **Question203. How Can A Compiled Object Code Be Viewed?**

**Answer :**

Compiled object code cannot be viewed, as it is not accessible to the user.

1. **Question204. What Is The Use Of The User\_errors Data Dictionary View?**

**Answer :**

The USER\_ERRORS data dictionary view is used to view the compilation errors of the subprograms.

1. **Question205. Write The Sql Query To View All The Procedures Within A Schema ?**

**Answer :**

SQL query to view all the procedures within a schema is as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/sql%20query.png

1. **Question206. How Can The Text Of A Procedure Or Function Specification Be Viewed?**

**Answer :**

The text of a procedure or function specification can be viewed using the USER\_SOURCE data dictionary view.

1. **Question207. Write The Syntax For The Show\_error Command ?**

**Answer :**

The syntax for the SHOW\_ERROR command is as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/show%20error%20command.png

1. **Question208. How Can The Parameter List For A Function Or Procedure Be Displayed?**

**Answer :**

The parameter list for a function or procedure can be displayed using the DESCRIBE command, given as follows:

DESCRIBE subprogram\_name

1. **Question209. What Are The Advantages Of Using Stored Procedures And Functions?**

**Answer :**

Stored procedures are a collection of SQL, PL/SQL, and Java statements that allow a programmer to shift the code from the application side to the database side. This enables the business rules and application logic to be stored as procedures within the database. The main advantages of using stored procedures and functions are as follows:

* + The complex business rules processing is performed within the database on the server side. Therefore, this change of processing from the application on the client side to the database on the server side enhances and improves the performance of the application.
  + The procedural code stored within the database can be reused anywhere within the application.
  + When a stored procedure is executed for the first time, the shared SQL area in the System Global Area (SGA) of the database stores the parsed versions of the executed commands. After this, whenever the command is executed, it takes the advantage of the parsing, which was performed the first time, and improves the performance of the procedure's execution.
  + Developmental efforts are reduced by keeping the code in a stored procedure. As the business rules are part of the database, they need not be re-written in each code module of the application; and therefore, saves the creation time of the application.
  + As the code is stored in a common place, any change can be easily managed; and therefore, reduces the maintenance effort.

1. **Question210. What Is A Package And How Does It Differ From Procedure And Function?**

**Answer :**

A package is a group of logically related PL/SQL subprograms bundled together with PL/SQL types and other associated objects. It consists of the following two parts:

* + ***Package specification*** —Includes declaration of all the objects and subprograms that are part of the package.
  + ***Package body***—Includes the total definition of all the parts of a package that have been declared in the specification.

The whole package is fully loaded into the memory when any package construct is called for the first time. Therefore, it does not require any disk input/output on later calls to constructs in the same package.

1. **Question211. What Is The Difference Between Private Package Construct And Public Package Construct?**

**Answer :**

* + Public package constructs are declared in the package specification but defined in the package body and can be invoked from any other object in the oracle environment.
  + Private package constructs are those that are declared and defined only within the package body. They are not declared in the package specification and can only be referenced by other constructs, which are part of the same package.
  + Private package construct is always defined before a public package construct within the package body.

1. **Question212. Can A Complete Package Be Called?**

**Answer :**

No, a complete package is not possible to call, invoke, or parameterize; however, any construct from the package can be called or invoked.

1. **Question213. What Is The Difference Between Local And Global Variables Within A Package?**

**Answer :**

A ***local variable*** is defined within a package body and is only available to the constructs within the same package.  
  
A ***global variable*** is one that is declared in the package specification. This can be referenced outside the package and is visible, to external objects that have access privilege on the package.

1. **Question214. Is It Possible To Write A Package Specification Without A Package Body?**

**Answer :**

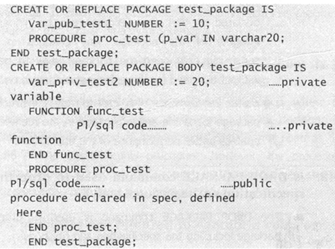
Yes, it is possible to write a package specification without a package body but not the vice versa. This is specifically used if there is a package, which is only used for the declaration of public variables, types, constants, and exceptions. In such case, there is no need for a package body and only package specification is enough.

1. **Question215. What Is Written As Part Of The Package Specification And The Package Body?**

**Answer :**

The package specification consists of the declaration of all the global variables, cursors, types, procedures, and functions that are public and required to be called from within or outside the package.

The package body consists of the definition of all the public constructs that are declared within the package specification and the definition of all the private variables, cursors, types, procedures, and functions that are required only within the package. Following code snippet shows the implementation of the package specification and the package body:



1. **Question216. Suppose A Procedure, A\_proc, Is Defined With A Number Parameter As Input Within A Package Called A\_package In The A\_user Schema. Write The Statement To Invoke The A\_proc Procedure On The Sql Prompt From The User,b\_user ?**

**Answer :**

The statement is given as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/query.png

1. **Question217. When Does A Package Need To Be Recompiled?**

**Answer :**

A package needs to be recompiled completely whenever any changes are made in the package specification. Any change to the package body does not require recompilation of the package.

1. **Question218. What Are The Advantages Of Using Packages?**

**Answer :**

The main advantages of using packages are given as follows:

* + ***Encapsulation***—Packages provide encapsulation of all the logically related constructs together in a single module, which is easy to code and understand.
  + ***Better application design***—Both modularity and encapsulation leads to better designing of applications. Separate package specification and body helps in simplifying the coding and better definition of the subprograms.
  + ***Hiding*** —Public and private constructs can be defined and declared so that any definition can be hidden to protect integrity of the package. Any changes in the private constructs only need the package body to be recompiled without the need for the entire application to be recompiled.
  + ***Better performance*** —A call to any construct within the package loads the entire package in the memory; therefore, enables better performance of the application.

1. **Question219. Is Drop Package Command Used To Drop The Package Specification Or The Package Body?**

**Answer :**

The DROP PACKAGE command is used to drop the entire package including the specification and the body.

1. **Question220. Is It Possible To Drop The Package Body Without Dropping The Package Specification?**

**Answer :**

Yes, it is possible to drop the package body without dropping the package specification using the DROP PACKAGE BODY command.

1. **Question221. How The Concept Of Overloading Is Implemented Within A Package?**

**Answer :**

PL/SQL allows you to overload the subprogram names and type methods. The package can include subprograms that have the same name as long as they have different formal parameters. PL/SQL determines which subprogram to be called depending on the formal parameters. In PL/SQL, only packaged subprograms can be overloaded.

1. **Question222. What Is Forward Declaration And How Can It Be Achieved Within A Package?**

**Answer :**

PL/SQL does not allow the use of any subprogram before it is declared and defined. However, in a package, forward declaration is allowed in case the subprogram specification is terminated by a semi-colon. This enables the subprogram to be defined in a logical manner and groups the subprograms in a package. It also helps in defining mutually recursive programs that call each other directly or indirectly.

1. **Question223. What Is A One-time-only Procedure And How Is It Declared Within A Package?**

**Answer :**

A One-Time-Only procedure is executed only once, when the package is called for the first time. While declaring a One-Time-Only procedure the keyword END is not used at the end of the one-time-only procedure; however, the procedure uses the package body's END clause, as the procedure's code is placed at the bottom of the package body.

1. **Question224. Which Privileges Are Required To Execute A Subprogram Owned By Another User?**

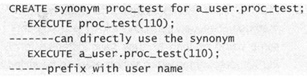
**Answer :**

To execute a subprogram owned by another user, the user must be granted either the EXECUTE privilege on a procedure or function or the EXECUTE ANY PROCEDURE system privilege.

1. **Question225. How Can A Procedure Or Function Be Executed Which Is Owned By Another User?**

**Answer :**

A procedure or function, which is owned by another user, may be executed by either creating a synonym for the subprogram or prefixing it with the user name, as shown in the following code snippet:



1. **Question226. What Is The Significance Of Authid Property While Creating A Procedure?**

**Answer :**

The AUTHID property affects the name resolution and privilege checking of SQL statements at runtime; however, it does not affect the compilation, and has no meaning for blocks that have no code, such as collection types.  
  
Any procedure may be created with its AUTHID property set to either the DEFINER or the INVOKER rights. If a procedure is created with the DEFINER rights, then the user executing the procedure need not have an access to database objects, which the procedure is accessing. However, if a procedure is created with the INVOKER rights, then the user must have access rights to all the objects that the procedure is accessing.

1. **Question227. How Can You Bug A Procedure Or Function For Errors?**

**Answer :**

The procedures or functions can be debugged using the SHOW\_ERROR command or the DBMS\_OUTPUT command within the subprogram code.

1. **Question228. When Does A Subprogram Need Recompilation?**

**Answer :**

When any database object associated with the subprogram changes, the program needs to be recompiled. When the program is executed the next time, after the object is changed, the system will automatically perform runtime recompilation. To avoid this performance overhead, it is better to explicitly recompile the subprogram using the ALTER [PROCEDURE | FUCNTION] command, as shown in the following statement:

ALTER PROCEDURE proc\_get\_marks compile;

1. **Question229. When Should A Procedure And A Function Be Used?**

**Answer :**

Procedure is used when there are more than one return values; however, if there is only one return value, then a function should be used. Although functions may have more than one OUT parameters, it is considered as a poor programming style. In addition, when a subprogram has to be invoked within an SQL statement, function should be used.

1. **Question230. Can Stand-alone Programs Be Overloaded?**

**Answer :**

No, stand-alone programs cannot be overloaded; however, packaged sub-programs can be overloaded within the same package.

1. **Question231. What Are Function Purity Levels?**

**Answer :**

Function purity levels define what type of data structures can be read or modified by a function. There are four function purity levels, which are given as follows:

* + ***WNDS***—Writes no database state. The function does not modify any database table using any DML statement.
  + ***RNDS***—Reads no database state. The function does not read any database tables using the SELECT statement.
  + ***WNPS*** —Writes no package state. The function does not modify any packaged variables.
  + ***RNPS***— Reads no package state. The function does not read any packaged variables.

1. **Question232. What Are The Restrictions On Functions That Are Called Within Sql Statements?**

**Answer :**

The functions that are called within SQL statements have the following restrictions:

* + They must be stored in a database.
  + They cannot modify any database table.
  + They can only take the IN parameter modes. The OUT and IN OUT parameter modes are not allowed in a function.
  + They can only use valid SQL datatypes, such as NUMBER, VARCHAR2, and DATE. However, they cannot use PL/SQL datatypes, such as BOOLEAN, RECORD, and TYPE.
  + The return type of a function should be of SQL datatype.
  + They cannot have the COMMIT and ROLLBACK statements.
  + They cannot have the ALTER SESSION and ALTER SYSTEM commands.

1. **Question233. Suppose A Procedure Proc\_\_calc\_student\_marks Inserts The Values Into The Student\_total\_marks Table. If The Table Is Modified By Adding Another Column, What Needs To Be Done To Make The Procedure Valid?**

**Answer :**

Oracle recompiles the procedure automatically when the procedure is invoked. Therefore, no action has to be taken if the INSERT statement is using column names. However, if the INSERT statement is just adding values without using the column names, then the procedure has to be modified, as another column has been added.

1. **Question234. Suppose A Procedure Is Created Using The Create Procedure Command; However, It Gives A Syntax Error. Which Data Dictionary Views Will Have The Source Code Of The Procedure?**

**Answer :**

The USER\_SOURCE, ALL\_SOURCE, and DBA\_SOURCE data dictionary views will have the source code, as a procedure is stored in the database even if it has a syntax error.

1. **Question235. Can A Parameter Use A Sequence Generator For The Assignment Of Default Value To A Parameter?**

**Answer :**

No, a sequence generator cannot be used to assign a default value to a parameter.

1. **Question236. Can An Out Parameter, V\_totalmarks, Be Referenced Within The Procedural Code Without Being Assigned A Value?**

**Answer :**

Any OUT parameter cannot be referenced without being assigned a value, as the OUT parameter cannot be assigned a default value. Therefore, the v\_total\_marks parameter cannot be referred within the code until the executable statements of the procedure have begun; the OUT parameter is assigned a value using the PL/SQL assignment statements.

1. **Question237. Can An In Parameter Be Assigned A Value Within A Procedure?**

**Answer :**

No, an IN parameter cannot be assigned values within a procedure.

1. **Question238. Suppose A Procedure Proc\_get\_highest\_rank Is Stored In The User Schema, User\_a. How Can You Access The Procedure Within This Schema?**

**Answer :**

The proc\_get\_highest\_rank procedure can be accessed within the user\_a schema from another stored procedure. It can also be accessed either from an anonymous PL/SQL block executed on the SQL prompt logged in as user\_a or directly at the SQL prompt using the EXECUTE command. In addition, it can be accessed by other users if they have the EXECUTE privilege.

1. **Question239. Can Boolean Datatype Be Used In Functions That Are Called From Sql Statements?**

**Answer :**

No, BOOLEAN datatype cannot be used in functions that are called from SQL statements.

1. **Question240. Is It Possible To Make Procedure Or Function Part Of The If Then Else Block?**

**Answer :**

Procedure cannot be called from within the IF THEN ELSE block; however, functions can become a part of the IF THEN ELSE block because functions return the value in such a manner that the function call in the block acts like a variable.

1. **Question241. Can You Invoke A Stored Function Or Procedure From The Oracle Forms In The Database?**

**Answer :**

Yes, a stored procedure or function can be invoked from the Oracle Forms and the result can be used for further programming.

1. **Question242. What Is The Meaning Of The Following Error Statement: Ora-06575: Package Or Function Func\_test Is In An Invalid State?**

**Answer :**

This error is displayed if the function is executed when it is in the invalid state. The ORA-06575 error indicates that for some reason the function is not valid and needs to be checked and compiled again.

1. **Question243. Suppose A Function Named Func\_test\_a Calls Another Function, Func\_test\_b. If The Func\_test\_b Function Is Dropped, What Will Be The Effect On The Func\_test\_a Function?**

**Answer :**

The calling function func\_\_test\_a becomes invalid. It has to be edited and compiled again to become valid.

1. **Question244. Can Functions Be Called As Stand-alone Programs Like Procedures?**

**Answer :**

No, functions cannot be called as stand-alone programs like procedures. They have to be a part of an expression or they should be invoked as part of SQL statements.

1. **Question245. Can Functions Stored In Oracle Forms Or Reports Be Invoked From Sql Statements?**

**Answer :**

No, functions stored on the client side in Oracle Forms or Reports cannot be invoked from SQL statements. They can only be used as PL/SQL expressions in the program units of the same client program.

1. **Question246. Can A Function Be Compiled Without A Return Statement?**

**Answer :**

Yes, a function can be compiled without a RETURN statement, but it cannot be executed.

1. **Question247. Can A Return Statement In The Function Code Be Followed By Other Valid Pl/sql Statements?**

**Answer :**

It is advisable to keep the RETURN statement at the end of the function code. This is because the function processes the statements until the RETURN statement is encountered and then the control is passed back to the calling environment.

1. **Question248. Suppose A Function Is Stored In An Oracle Form On The Client Side. Can The Function Be Called From Another Program Unit In Another Oracle Form?**

**Answer :**

It is not possible to call a function stored in one Oracle Form from another Oracle Form unless it is defined and declared in a PL/SQL library, which is shared between the two Oracle Forms.

1. **Question249. How Does A Package Support Information Hiding?**

**Answer :**

A package consists of two parts: package specification and package body. The package specification contains the names of procedures and functions, their parameters, and return types; whereas, the subprogram code is part of the package body. Developers who require calling the procedures or functions just need to have access on the package specification. Therefore, the package owner can grant privilege on the package specification while hiding the package body. In this way, the developers can use the package without seeing the actual code. In addition, package owners can create procedures and functions that are private for the package and need not be part of the package specification. In this way, information hiding is achieved within a package.

1. **Question250. Where Can A Package Be Stored?**

**Answer :**

A package can be stored on the server side within a schema in the database. It can also be stored on the client side in the form of Oracle Reports, Oracle Forms, or as part of PL/SQL library.

1. **Question251. Does A Package Is Recompiled If The Database Objects That The Package Uses Are Modified?**

**Answer :**

Yes, if the database objects associated with the package are modified, the package specification and the package body become invalid; and therefore, the package needs recompilation using the following syntax:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/package.png

1. **Question252. Are Global Constructs Declared In Packages Available To The Entire User Session?**

**Answer :**

Global constructs once declared in package are available throughout the user session. Until the user session is terminated, the global construct retains its value across the PL/SQL blocks. This is applicable for all the packaged global constructs, such as constants, exceptions, and cursors. For example, a globally declared cursor may be opened in one PL/SQL block, fetched in another block, and closed in some other block.

1. **Question253. Where Can A Constant Be Referenced If The Constant Is Declared In A Package Specification?**

**Answer :**

The constant declared in a package specification can be referenced in the same package where it is declared, stored procedure, stored function, or an anonymous b : c.

1. **Question254. Can A Variable Be Referenced In A Package If It Is Declared In The Package Body Of Another Package?**

**Answer :**

Variables declared in a package body are private variables. Therefore, these cannot be referenced in other packages.

1. **Question255. How Many Entries Does The User\_objects Data Dictionary Have For A Package?**

**Answer :**

Two separate entries are created for package specification and package body in the USER\_OBJECTS data dictionary.

1. **Question256. What Is The Minimum Requirement To Create A Package?**

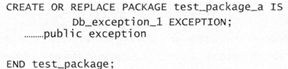
**Answer :**

Package specification, which declares constructs, such as variables, constants, user-defined exceptions, and cursors, is enough to create a complete package. Package body is not required if there are no procedures and functions to be defined in the package.

1. **Question257. How Can A Public Exception, Which Is Global In Nature, Be Created In A Package?**

**Answer :**

An exception is called a public exception if it is declared in a package specification as a separate construct, as shown in the following code:



1. **Question258. Write A Statement To Display The Text Of The Test\_packl Package From Data Dictionary View ?**

**Answer :**

The statement is given as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/query%201.png

1. **Question259. What Is An Exception?**

**Answer :**

An exception is an identifier, which defines an error condition within the Procedural Language/Structured Query Language (PL/SQL) code. When an error occurs, it is trapped and the program control unconditionally branches to the exception section of the PL/SQL block.

1. **Question260. How Is An Exception Raised And Handled?**

**Answer :**

When a runtime error occurs, an exception is raised. The exception is handled when this error is successfully handled by the PL/SQL code and is not allowed to propagate to the outward environment. Compilation error cannot be handled within the program; only runtime errors can be handled within the PL/SQL code.  
An exception can be raised:

* + By a runtime Oracle error
  + By the RAISE statement
  + By calling the RAISE\_APPLICATION\_ERROR procedure

After an exception is raised, the control is passed to the exception section, where the code is written about how to handle the error that has occurred. The control cannot pass back to the executable section of the same block after the exception is handled.

1. **Question261. When Is An Exception Propagated To The Outside Environment?**

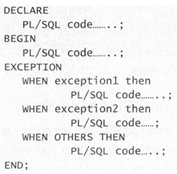
**Answer :**

An exception is raised within the program and if there is no corresponding exception handler, the PL/SQL block terminates unsuccessfully and the exception is propagated to the calling environment.

1. **Question262. Give The Block Structure Of An Exception Section Within The Pl/sql Code ?**

**Answer :**

The structure of an exception section within the PL/SQL code is given as follows:



1. **Question263. What Are The Different Types Of Exceptions?**

**Answer :**

The three types of exceptions that can be handled at runtime are as follows:

* + ***Predefined Oracle error*** —Refers to an undeclared error, which is raised by the Oracle server implicitly.
  + ***Non-predefined Oracle error***—Refers to an error declared in the declarative section of the code that is raised by the Oracle server implicitly.
  + ***User-defined error*** —Refers to an error declared in the declarative section of the code that is raised explicitly.

1. **Question264. Describe The Others Exception Handler ?**

**Answer :**

The exception section handles only those errors that are specified; however, other errors that are not specified cannot be handled. To handle these errors explicitly, the OTHERS exception handler is used that is always defined as the last exception handler.

1. **Question265. Can The Pl/sql Block Process More Than One Exception At A Time?**

**Answer :**

No, the PL/SQL block cannot handle more than one exception. It can process only one exception handler before terminating.

1. **Question266. Is It Possible To Have More Than One Others Clause?**

**Answer :**

No, it is not possible to have more than one OTHERS clause in the exception section.

1. **Question267. Give A Few Pre-defined Oracle Errors ?**

**Answer :**

Pre-defined Oracle errors are as follows:

* + ***NO\_DATA\_FOUND*** —Refers to an error when a SELECT statement does not return any result set.
  + ***DUP\_VAL\_ON\_INDEX*** —Refers to an error when a program tries to store duplicate values in an indexed column.
  + ***TOO\_MANY\_ROWS*** —Refers to an error when a SELECT INTO statement returns more than one row.
  + ***VALUE\_ERROR*** —Refers to an error when an arithmetic, conversion, truncation, or size-constraints error occurs.
  + ***ROWTYPE\_MISMATCH*** —Refers to an error when the actual and formal parameters do not have the same datatypes

1. **Question268. What Is The Advantage Of Others Exception Handler?**

**Answer :**

The OTHERS exception handler ensures that no exception goes unhandled and the program terminates successfully.

1. **Question269. Describe The Sqlcode And Sqlerrm Functions ?**

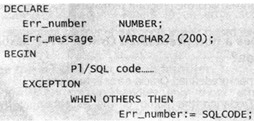
**Answer :**

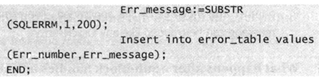
The SQLCODE function returns the Oracle error number for internal exceptions. This error number can be passed to the SQLERRM function to get the message associated with the error number. The SQLCODE function returns the value +1; whereas, the SQLERRM function returns the User-Defined Exception message for the user-defined exceptions.

1. **Question270. How Can The Sqlcode And Sqlerrm Functions Be Used For Exception Handling?**

**Answer :**

These functions can be used with the OTHERS exception handler to get the error number and the error message that has caused the OTHERS exception to be raised, as shown in the following code snippet;

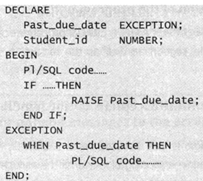




1. **Question271. How Are User-defined Exceptions Trapped?**

**Answer :**

User-defined exceptions are defined in the declarative section of the PL/SQL code and raised explicitly using the RAISE statement, as shown in the following code snippet:



1. **Question272. What Happens To The Program Control When The Raise\_application\_error Procedure Is Encountered?**

**Answer :**

When the RAISE\_APPLICATION\_ERROR procedure is encountered, it ends the program unit and returns the error number and message to the application. The error number and message can be trapped like any Oracle error.

1. **Question273. What Happens After A Sub-block Handles An Exception?**

**Answer :**

When a sub-block handles an exception, the control passes to the next statement in the executable section of the enclosing block immediately after the END statement of the sub-block.

1. **Question274. Can Raise\_application\_error Be Called In The Executable Section Of The Program Unit?**

**Answer :**

Yes, the RAISE\_APPLICATION\_ERROR procedure can be called in the executable section as well as the exception section of the PL/SQL code.

1. **Question275. What Happens If An Exception Is Not Handled In A Sub- Block?**

**Answer :**

If an exception is not handled in a sub-block, the exception propagates to the enclosing blocks until the associated handler is found. If none of the blocks has a handler for the exception, an unhandled exception error is passed back to the host environment as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/sub%20block.png

1. **Question276. Can Processing Be Resumed From The Point Exception Was Raised After The Exception Is Handled?**

**Answer :**

After the exception is handled, processing cannot be resumed within the executable section of the current block from where the exception was raised. The current block where the exception handler is declared will be terminated. The control will pass to the statement in the enclosing executable block. If there is no enclosing block, control will pass back to the calling host environment.

1. **Question277. How Can A Transaction Be Retried After An Exception?**

**Answer :**

The following method can be used to retry a transaction after an exception is raised:

* + The transaction should be written within a sub-block
  + The sub-block can be placed inside a loop that repeats the transaction.
  + Mark a savepoint before starting the transaction. If the transaction succeeds, then commit and exit from the loop. If the transaction fails, the control is transferred to the exception handler where the transaction is rolled back to the savepoint to undo the changes, and then try to fix the problem.

1. **Question278. Does Raise\_application\_error Overwrite The Error Stack Or Append Messages To The Error Stack?**

**Answer :**

The default behavior of the RAISE\_APPLICATION\_ERROR procedure is to overwrite the error message stack. However, if the parameter is passed, then the messages can be appended to the error stack as well. Following is the syntax of the RAISE\_APPLICATION\_ERROR procedure:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/msg.png

In the preceding syntax, TRUE adds the messages to the error stack; whereas, FALSE overwrites the error stack.

1. **Question279. What Causes The Invalid\_cursor Exception?**

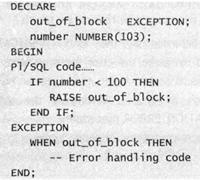
**Answer :**

This exception is raised when the program attempts to perform an illegal operation, such as closing an unopened cursor.

1. **Question280. How Can A Current Exception Be Re-raised?**

**Answer :**

The RAISE statement can be used to re-raise the current exception. It stops the normal execution of a PL/SQL program and transfers the control to an exception handler, as shown in the following code:



1. **Question281. Where Will The Control Be Transferred If An Exception Is Raised Within An Exception Section Of The Block?**

**Answer :**

When an exception is raised within an exception section of a block, the control is transferred to the exception section of the enclosing block.

1. **Question282. When Is The Cursor\_already\_open Exception Raised?**

**Answer :**

This exception is raised when the program tries to open an already opened cursor. The cursor should be closed before it can be opened again.

1. **Question283. When Is The Row\_type\_mismatch Exception Raised?**

**Answer :**

The ROW\_\_TYPE\_MISMATCH exception is raised when a host cursor variable and a PL/SQL cursor variable involved in an assignment have incompatible return types.

1. **Question284. What Is The Advantage Of Having A Separate Exception Section Within A Pl/sql Code?**

**Answer :**

The advantages of having a separate exception section within a PL/SQL code are as follows:

* + It is easier to understand the program logic as it is easily visible and readable
  + Any error in any statement will be detected and handled easily

1. **Question285. What Happen When There Is No Handler For A Raised Exception?**

**Answer :**

If an exception is raised and there is no handler in the current block, the exception would propagate to the enclosing block. If no handler is found in the enclosing block, the exception will be propagated to the calling environment as an unhandled exception.

1. **Question286. What Is The Difference Between User-defined Exceptions And Oracle-defined Exceptions?**

**Answer :**

User-defined exceptions are explicitly raised; whereas, Oracle- defined exceptions are raised implicitly.

1. **Question287. What Is The Difference Between Raise And Raise\_application\_error?**

**Answer :**

* + ***RAISE***is used to call pre-defined exceptions declared in the declarative section of the PL/SQL block.
  + ***RAISE\_APPLICATION\_ERROR*** is used to call user-defined ORA - error messages from stored subprograms.

1. **Question288. What Is The Meaning Of Pragma Keyword?**

**Answer :**

The PRAGMA keyword signifies that the statement is a compiler directive, which is not processed when the PL/SQL is executed. It is a pseudo-instruction that tells the compiler to interpret all the occurrences of exception name within the block with the associated Oracle server number.

1. **Question289. What Are Oracle Supplied Packages?**

**Answer :**

Oracle server provides the Oracle Supplied Packages to allow Procedural Language/Structured Query Language (PL/SQL) to access certain features of SQL, which are not available in PL/SQL. They also help in extending the functionality of the database.

1. **Question290. What Is Catproc.sql?**

**Answer :**

The catproc.sql is a SQL file, which has the script to create packages in the database, This file has to be executed on the SQL prompt for the packages to be available in the database.

1. **Question291. What Is Dynamic Sql And What Is Its Usage?**

**Answer :**

Dynamic SQL is used by PL/SQL to execute Data Definition Language (DDL) statements, Data Control (DCL) statements, or Transaction Control statements within PL/SQL blocks. These statements are not stored within the source code but are stored as character variables in the program. The SQL statements are created dynamically at runtime by using variables. This is used either using native dynamic SQL or through the DBMS\_SQL package.

1. **Question292. How Is An Sql Statement Processed? What Are The Steps That An Sql Statement Undergoes For Compilation?**

**Answer :**

SQL statements are compiled using the following steps:

* + ***Parse***—Refers to a step in which statements are checked for existence of all objects used and the syntax is validated.
  + ***Bind***— Refers to a step In which the values for the variables are obtained. This process is also called binding variables.
  + ***Execute*** —Refers to a step in which the statement is executed by Oracle.
  + ***Fetch*** —Refers to a step in which rows are selected and retrieved one by one until last row is fetched.

1. **Question293. What Is The Use Of Dbms\_sql Package?**

**Answer :**

The DBMS\_SQL package is used to access dynamic SQL and dynamic PL/SQL from within a PL/SQL program. It allows you to execute statements that are otherwise impossible from within PL/SQL programs, which includes DDL, DCL, or Transaction Control statements.

1. **Question294. Does Dynamic Sql Support All Sql Data Types?**

**Answer :**

Yes, dynamic SQL supports all SQL data types. It is possible to define variables and bind arguments of collection type, LOBs, and REFs.

1. **Question295. Are The Pl/sql Specific Datatypes Supported By Dynamic Sql To Define Variables And Bind Arguments?**

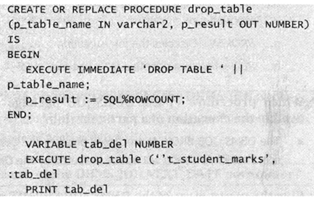
**Answer :**

No, PL/SQL datatypes are not supported by dynamic SQL; only PL/SQL RECORD datatype can be used in the INTO clause. No other PL/SQL datatype, such as BOOLEAN, can be used within dynamic SQL.

1. **Question296. What Is The Use Of Execute Immediate Command? Explain With Example ?**

**Answer :**

The native dynamic SQL can be executed within a PL/SQL block using the EXECUTE IMMEDIATE command, as shown in the following example:



1. **Question297. Give The Usage Of Dbms\_ddl Package ?**

**Answer :**

The DBMSJDDL package provides access to certain DDL statements within PL/SQL blocks. This package runs with the privileges of the calling user and not the owner of the SYS package. Some of the procedures included in this package are as follows:

* + DBMS\_DDL.ALTER\_COMPILE object\_type, owner, object\_\_name)
  + DBMS\_DDL.ANALYZE\_OBJECT(object\_type, owner, object\_name, method)

1. **Question298. Which Oracle Package Is Used For Scheduling Job?**

**Answer :**

The DBMS\_JOB package is used for scheduling and executing PL/SQL programs.

1. **Question299. Which Are The Subprograms Of The Dbms\_job Package?**

**Answer :**

The subprograms of the DBMS\_JOB package are as follows:

* + ***SUBMIT***—Submits a job to the job queue
  + ***REMOVE***—Removes the specified job from the job queue
  + ***CHANGE*** —Alters the specified job that has already been submitted to the job queue
  + ***WHAT***—Alters the specified job
  + ***NEXT\_\_DATE***—Alters the execution time
  + ***INTERVAL*** —Alters the interval between execution of specified job
  + ***BROKEN***—Disables the job execution
  + ***RUN***—Forces a specified job to be executed

1. **Question300. Which Procedure Of The Dbms\_job Package Is Used To Disable The Execution Of A Particular Job?**

**Answer :**

The DBMS\_JOB.BROKEN procedure disables the execution of the specified job. This job will not be executed by the Oracle server.

1. **Question301. Give The Parameters Of The Dbms\_job.submit Procedure ?**

**Answer :**

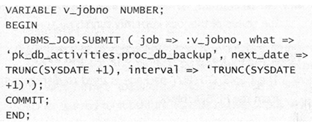
The DBMS\_JOB.SUBMIT procedure adds a new job to the job queue. It takes four parameters as input and returns the job number as the output though the JOB parameter. The parameters of the DBMSJOB.SUBMIT procedure are as follows:

* + ***JOB***— Refers to the OUT parameter that gives the output of the job number, which is the unique identifier for the job
  + ***WHAT***— Refers to the IN parameter, which provides the PL/SQL code to be executed
  + ***NEXT\_DA TE*** —Refers to the IN parameter, which provides the next execution date of the job
  + ***INTERVAL*** —Refers to the IN parameter for the interval used to calculate the next execution date
  + ***NO\_PARSE*** —Refers to the IN parameter, which is a Boolean flag indicating whether to parse the job at the time of job submission

1. **Question302. Provide A Program Code In Which The Dbms\_job.submit Procedure Is Used To Schedule A Database Backup Job In The Queue For Midnight ?**

**Answer :**

The following program code is used to schedule a database backup job using the procedure pk\_db\_activities.proc\_db\_backup in the queue for midnight:



1. **Question303. When Is An Exception Raised By The Dbms\_job Package?**

**Answer :**

An exception is raised if the time specified for the job does not evaluate to a job in future.

1. **Question304. What Is Use Of The Dbms\_job.next\_date Parameter?**

**Answer :**

The DBMS\_JOB.NEXT\_DATE parameter is used to reschedule a particular job. This parameter is used when the next execution date for a particular job needs to be changed.

1. **Question305. Give A Statement In Which The Dbms\_job.change Procedure Is Used To Change The Backup Job Submitted To Morning 8am And Above ?**

**Answer :**

The job can be rescheduled as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/query%20_%201.png

1. **Question306. Is It Possible To Run A Job Scheduled For A Particular Time To Be Run Immediately? If Yes, Then How?**

**Answer :**

Yes, It is possible to run a job scheduled for a particular time immediately using the DBMS\_JOB.RUN procedure.

1. **Question307. Which Data Dictionary View Is Used To See The Status Of Submitted Jobs?**

**Answer :**

The DBA\_JOBS data dictionary view is used to display the status of the submitted jobs.

1. **Question308. Which Data Dictionary View Is Used To See The Status Of Currently Running Jobs?**

**Answer :**

The DBA\_JOBS\_RUNNING data dictionary view is used to display the status of the jobs currently running.

1. **Question309. What Is The Use Of Dbms\_output Package?**

**Answer :**

The DBMS\_OUTPUT package is used to display output messages from the PL/SQL block.

1. **Question310. How Can The Dbms\_output Package Used To Get An Array Of Lines From Buffer Into The Procedure?**

**Answer :**

The DBMS\_OUTPUT package uses the GET\_LINES procedure to get an array of lines from output buffer into the procedure.

1. **Question311. What Is The Utl\_file Package?**

**Answer :**

The UTL\_FILE package provides text files input/output functionality within PL/SQL. The security on client-side is handled using normal operating system file permissions; whereas, server-side security is handled though restrictions on the directories that are accessed.

1. **Question312. What Is Essential To Be Set In The Init.ora File Before Using The Utl\_file Package?**

**Answer :**

It is essential to initialize the UTL\_FILE\_DIR parameter in the init.ora file before using the UTL\_FILE package. This parameter is initialized to the directory that is to be accessed as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/util_file.png

1. **Question313. How Can All The Directories Be Made Available To The Utl\_file Package?**

**Answer :**

The following settings will turn off all the database permissions on the directories that are accessible to the database processes and make all of them available to the UTL\_FILE package:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/util_file%20package.png

1. **Question314. What Is The Difference Between The Put\_line And Putf Functions Of The Utl\_file Package?**

**Answer :**

The difference between the PUT\_LINE and PUTF functions of the UTL\_FILE package is as follows:

* + ***PUT\_LINE*** —Refers to a function used to write text string, which is stored in the buffer, to the opened file with a line terminator.
  + ***PUTF***— Refers to a function used to write text string, which is stored in the buffer, to the opened file but in a formatted form with format specifiers - %s and n. The %s format specifier is used to substitute a value in the output string and n is a new line character.

1. **Question315. What Is The Use Of Fflush Procedure Of The Utl\_file Package?**

**Answer :**

The FFLUSH procedure is used to write all data buffered in memory to a file.

1. **Question316. Name A Few Exceptions Of The Utl\_file Package ?**

**Answer :**

A few exceptions of the UTL\_FILE package are as follows:

* + ***INVALID\_PATH*** —  Specifies that a file location or filename is invalid
  + ***INVALID\_MODE*** — Specifies that the OPEN\_MODE parameter in the FOPEN procedure is incorrect
  + ***READ\_\_ERROR*** — Specifies that an operating system error has occurred during the read operation
  + ***WRITE\_ERROR***  — Specifies that an operating system error has occurred during the write operation
  + ***INVALID\_\_OPERATION*** — Specifies that a file cannot be opened or used

1. **Question317. Can The Utl\_file Package Raise Predefined Pl/sql Error?**

**Answer :**

Yes, the UTL\_FILE package can raise predefined exceptions, such as NO\_DATA\_FOUND or VALUE\_ERROR.

1. **Question318. What Is The Difference Between Fopen And Is\_open Functions?**

**Answer :**

The difference between FOPEN and IS\_OPEN functions is given as follows:  
  
***FOPEN*** —Opens a file for input/output and returns a file handle used in I/O operations. Following is the syntax of the FOPEN function:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/f%20open.png

***IS\_OPEN*** — Returns a Boolean value when the file handle is passed as a parameter to the function to an open file. Following is the syntax of the IS\_OPEN function:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/is%20open.png

1. **Question319. Explain The Functionality Of Utl\_http Package ?**

**Answer :**

The UTL\_HTTP package enables PL/SQL and SQL to access the Internet data using the REQUEST and REQUEST\_PIECES functions. These functions take the Uniform Resource Locator (URL) of the website as input and return the HyperText Markup Language (HTML) data from the website.

1. **Question320. What Is The Size Of Data That Can Be Retrieved For The Database Using The Utl\_http Package?**

**Answer :**

The size of the data that can be accessed from the Internet is based on the functions used. The REQUEST function returns up to the first 2000 bytes of data retrieved from the website. The REQUEST\_PIECES function returns a PL/SQL table of 2000 bytes from the accessed URL.

1. **Question321. Which Exception Is Raised If The Utl\_http Package Fails Due To Wrong Url Specified?**

**Answer :**

If wrong URL is specified in the UTL\_HTTP package call, the REQUEST\_FAILED exception is raised.

1. **Question322. What Is The Utility Of Utl\_tcp Package? How Is The Connection Opened And Closed?**

**Answer :**

The UTL\_TCP package is used by PL/SQL applications to access external Transmission Control Protocol/Internet Protocol (TCP/IP) based servers using the TCP/IP protocol.  
  
The connection is opened using the OPEN\_CONNECTION function, which opens a TCP/IP connection when the remote host, local host, and both the port numbers are specified. It returns a connection of PL/SQL RECORD type.  
  
The connection is closed using the CLOSE\_CONNECTION function, which closes the connection of the port whose details are provided as input.

1. **Question323. In Which Situations Exceptions Are Raised While Using The Utl\_tcp Package?**

**Answer :**

An exception is raised using the UTL\_TCP package in the following situations:

* + The buffer size is too small for the input
  + A network error occurs
  + No more data is available to read from the connection
  + Invalid arguments are passed to the function

1. **Question324. What Is The Difference Between Close\_connection And Close\_all\_connections Procedure Of The Utl\_tcp Package?**

**Answer :**

The CLOSE\_CONNECTION procedure closes the connection whose details are provided as input where as the CLOSE\_ALL\_CONINJECTIONS procedure closes all the opened connections.

1. **Question325. Which Oracle Package Is Used To Check The Status Of A Schema And Analyze The Objects Of The Schema?**

**Answer :**

The DBMS\_UTILITY package is used to check the status of a schema and analyze the objects of the schema.

1. **Question326. What Is The Use Of Dbms\_transaction Package?**

**Answer :**

The DBMS\_TRANSACTION package controls logical transactions and improves the performance of short and non-distributed transactions.

1. **Question327. Is It Possible To Access Sql Session Information?**

**Answer :**

Yes, it is possible to access the SQL sessions using the DBMS\_SESSION package.

1. **Question328. Is There A Built-in Random Number Generator In Oracle?**

**Answer :**

Yes, there is a built-in random number generator in the DBMS\_RANDOM package.

1. **Question329. Which Oracle Package Is Used To Manage The Oracle Lock Management Services?**

**Answer :**

The DBMS\_LOCK package is used to manage the Oracle Lock Management services. This package is used to request, convert, and release the locks using the Oracle Lock Management services.

1. **Question330. What Is The Utility Of The Dbms\_logmnr\_d Package?**

**Answer :**

This package queries the dictionary tables of the current database and creates text based file with their contents.

1. **Question331. What Is The Difference Between Blob And Clob?**

**Answer :**

The BLOB datatype is used to store a binary large object, such as a video image file; whereas, the CLOB datatype is used to store a character large object.

1. **Question332. Is It Possible To Convert Long Objects To Lob Objects?**

**Answer :**

Yes, it is possible to convert LONG and LONG RAW objects to LOB objects using the TO\_LOB function.

1. **Question333. What Are The Two Components Of Lob Datatype?**

**Answer :**

The two components of the LOB datatype are given as follows:

* + ***LOB locator***— Refers to a locator, which points to the location in the database where the value is stored. This value is stored along with the record in the table row and is similar to a pointer to the actual location of LOB value.
  + ***LOB value*** — Refers to an actual image or value of the LOB datatype.

1. **Question334. Can Lob Datatype Be Used For Pl/sql Variables Or Parameters?**

**Answer :**

Yes, a LOB datatype can be used as a PL/SQL variable or parameter.

1. **Question335. What Is The Difference Between Internal And External Lobs?**

**Answer :**

LOB datatype can be internal or external. Internal LOBs are CLOB, NCLOB, and BLOB that are stored in the database; whereas, external LOBs are BFILEs that are stored outside the database as external files. Although both LOB types have the pointer to the location, the BFILEs are stored externally across hard disks or on external storage; whereas, internal LOBs have to be stored in the different location within the database itself.

1. **Question336. Can Bfiles Be Used For Transactions?**

**Answer :**

No, BFILEs are read-only and cannot participate in transactions.

1. **Question337. Are The Bfiles Deleted When Lob Column Record Is Deleted?**

**Answer :**

When the LOB record is deleted, the BFILE type of external file is not actually deleted; only the locator to the location in the database table is deleted.

1. **Question338. How Many Bfiles Can Be Opened In A Session?**

**Answer :**

The number of BFILEs that can be opened in a session is dependent on the parameter SESSION\_MAX\_OPEN\_FILES, which is set in the init.ora file. Its default value is 10. The maximum value is either 50 or the value set in the parameter.

1. **Question339. Which Constraints Are Allowed For Lob Objects?**

**Answer :**

The NULL and NOT NULL constraints are allowed for LOB objects.

1. **Question340. What Is The Difference Between To\_blob And To\_clob Functions?**

**Answer :**

The ***TO\_CLOB*** function is used to convert the LONG, VARCHAR2, and CHAR datatypes to the CLOB datatype.

The ***TO\_BLOB*** function is used to convert the LONG RAW and RAW datatype to the BLOB datatype.

1. **Question341. Is It Possible To Convert Clob To Char Type?**

**Answer :**

Yes, it possible to convert CLOB to CHAR type using the TO\_CHAR function.

1. **Question342. What Is The Usage Of Dbms\_lob Package?**

**Answer :**

The DBMS\_LOB package helps PL/SQL programs to access and manipulate the LOB objects.

1. **Question343. Does The Dbms\_lob Package Support Concurrency Control?**

**Answer :**

No, the DBMS\_LOB package does not support concurrency control. The user is responsible for locking the row explicitly before any transactions. The DBMS\_LOB package does not provide implicit locking of the row with LOB values.

1. **Question344. What Are Mutators And Observers In Relation With The Dbms\_lob Package?**

**Answer :**

The DBMS\_LOB package has functions and procedures to manipulate LOB objects. These can be categorized into mutators and observers. Mutators can modify LOB values, such as APPEND, COPY, WRITE, FILECLOSE, and FILEOPEN; whereas, observers can only read the LOB values, such as COMPARE, INSTR, GETLENGTH, FILEEXISTS, and FILEGETNAME.

1. **Question345. How Can A Lob Column Be Added To An Existing Table T\_students?**

**Answer :**

A LOB column can be added using the following syntax:

ALTER TABLE t\_students ADD(stud\_id\_file BLOB);

1. **Question346. Is It Possible To Query And See The Data In Blob And Bfile Format Using The Select Statement?**

**Answer :**

No, the BLOB and BFILE data cannot be seen using the SELECT statement query. Relevant software, such as Oracle Forms or Reports, should be used for this purpose.

1. **Question347. Is It Possible To Update Blob And Clob Columns Without Setting It To An Empty Locator?**

**Answer :**

No, the BLOB and CLOB columns cannot be updated until the value is set using the EMPTY\_BLOB or EMPTY\_CLOB function through an UPDATE statement.

1. **Question348. Does Oracle Maintain Data Integrity And Concurrency For Lob Datatypes?**

**Answer :**

Yes, Oracle maintains data integrity and concurrency for internal LOBs (BLOBs and CLOBs) but not for BFILEs, which are external LOBs.

1. **Question349. What Is The Benefit Of Storing The Lob Datatype Out-of- Line?**

**Answer :**

This allows the database to avoid scanning LOB data each time it reads rows from the table. The LOB data will be read only when it is required; however, at other times, only the LOB locator values are read.

1. **Question350. Is It Possible To Set Null Values To Bfile Columns?**

**Answer :**

Yes, BFILE can be set to NULL values; whereas, BLOB and CLOB cannot be set to NULL values. They have to be initialized using the EMPTY\_BLOB or EMPTY\_CLOB function.

1. **Question351. Which Functions Should Be Used Before Performing Data Manipulation On Bfiles?**

**Answer :**

Following are the procedures or functions of the DBMS\_LOB package that should be used before manipulating the BFILE data:

* + FILEOPEN— Opens files
  + FILEGETNAME—GeXs the filename
  + FILEEXISTS —Checks whether the file exists
  + FILEISOPEN— Checks whether the file is open

These procedures are used, as Oracle does not maintain the integrity of the external file; only the pointer is maintained.

1. **Question352. Can Lob Values Be Deleted From The Database?**

**Answer :**

When Internal LOB datatype (BLOB, CLOB, and NCLOB) column values are deleted, both locator and values are deleted. If an external LOB (BFILE) value is deleted, then only the locator value is deleted. The external file has to be manually deleted from the server file system.

1. **Question353. What Is The Difference Between Referenced Object And Dependent Object?**

**Answer :**

If the definition of an object A references object B, then A is the dependent object and B is the referenced object. If a table is being queried in a procedure, then the table is the referenced object and the procedure is the dependent object. However, if the definition of the table is modified, the procedure may or may not execute correctly.

1. **Question354. Does Oracle Automatically Record The Dependencies Among The Schema Objects?**

**Answer :**

Yes, Oracle automatically records the dependencies among objects through the USER\_OBJECTS data dictionary view, where the schema objects have a valid or invalid status.

1. **Question355. Which Schema Objects Can Be Referenced Directly Or Indirectly?**

**Answer :**

Any procedure or function can be referenced directly or indirectly using the following objects:

* + Tables
  + Views
  + Procedures
  + Functions
  + Sequences
  + Package procedures or functions

1. **Question356. What Is Indirect Reference Of Schema Objects?**

**Answer :**

When any procedure or function references another schema object through an intermediate procedure, function, or view, then the reference is called indirect reference.

1. **Question357. How Does An Object Get An Invalid Status?**

**Answer :**

If there is any change in any of the referenced object, the dependent object will get an invalid status. For example, if the structure of base table of a view is modified, the view will get an invalid status.

1. **Question358. What Happens When An Object Has An Invalid Status?**

**Answer :**

Whenever an invalid object is called again, the Oracle server automatically recompiles the object and it gets a valid status. For example, if a view with an invalid status is queried, it will be recompiled before the results are displayed.

1. **Question359. Which Data Dictionary Views Are Used To Display The Dependencies Between The Schema Objects?**

**Answer :**

The USER\_DEPENDENCIES, DBA\_DEPENDENCIES, and ALL\_DEPENDENCIES data dictionary views display the dependencies among the schema objects.

1. **Question360. Which Column In The User\_dependencies View Indicates The Type Of The Reference Object?**

**Answer :**

The REFERENCED\_TYPE column of the USER\_DEPENDENCIES view indicates the type of the referenced object.

1. **Question361. Write A Query To Display All Object Types That Are Dependent On Other Objects In The Database ?**

**Answer :**

The following query will display the object types that are referenced by other objects in the database:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/order.png

1. **Question362. Suppose The Student\_grades View Is Based On The Student\_marks Table. If The Length Of The Student\_name Column Is Modified, Will The View Become Invalid?**

**Answer :**

If the STUDENT\_GRADES view Includes the STUDENT\_NAME column then it will become invalid; and therefore, needs to be recompiled. If the view does not include the column, which is modified, then it is valid.

1. **Question363. What Is Cascading Invalidation?**

**Answer :**

Suppose object A references object B, which in turn references object C, then A is an indirect dependent of object C. If object C is changed, then object B is invalidated; and therefore, invalidates object A as well. This is called cascading invalidation.

1. **Question364. What Is The Effect On The Dependent Objects If An Object Is Renamed?**

**Answer :**

All the dependent objects become invalid if an object, which the dependent objects reference, is renamed.

1. **Question365. What Is The Use Of Deptree And Ideptree Views?**

**Answer :**

These views contain information on the object direct and indirect dependency tree.

1. **Question366. How Do You Create The Deptree And Ideptree Views?**

**Answer :**

These views are created by executing the utldtree.sql file which is found in the $ORACLE\_HOME/rdbms/admin folder.

1. **Question367. What Is The Significance Of The Nested\_level Column In The Deptree View?**

**Answer :**

The NESTED\_LEVEL column shows the level of nesting of a particular object within the dependency tree. The parent object has a NESTED\_LEVEL of 0 and all other objects are numbered as per their level in relation to the parent object.

1. **Question368. What Is The Meaning Of Nested\_level 2 For An Object In The Dependency Tree?**

**Answer :**

NESTED\_LEVEL 2 for an object in the dependency tree means that the object is dependent on another object, which has NESTED\_LEVEL 1.

1. **Question369. What Is The Ideptree View And How Is It Different From The Deptree View?**

**Answer :**

The IDEPTREE view is a pre-indented representation of the hierarchical structure of the schema objects, It is a sorted version of the DEPTREE view.

1. **Question370. Which Procedure Has To Be Executed Before Using The Ideptree View?**

**Answer :**

The DEPTREE\_FILL procedure needs to be executed before the IDEPTREE view can be used.

1. **Question371. What Information Is Stored In The Ideptree View?**

**Answer :**

The IDEPTREE view has the hierarchical information of the dependency tree in a single column called DEPENDENCIES. This column stores a single string of information showing the dependent objects indented as per the hierarchy.

1. **Question372. Which View Is Used To Track The Security Dependencies?**

**Answer :**

The security dependencies can be tracked using the USER\_TAB\_PRIVS data dictionary view.

1. **Question373. What Is The Effect On The Dependent Object When A Package Body Is Modified?**

**Answer :**

When a package body is modified, there is no change in the dependent objects.

1. **Question374. Is A View Invalidated When A New Column Is Added To The Table It References?**

**Answer :**

No, a view is not invalidated if a new column is added to the table it references.

1. **Question375. What Is The Advantage Of Referring Tables Indirectly Using Views?**

**Answer :**

Referencing tables indirectly using views allows:

* + Adding columns to the table without invalidating dependent views or dependent PL/SQL objects
  + Modifying or deleting columns of the table that are not referenced by the view without invalidating the dependent objects

1. **Question376. How Does Oracle Manage Dependencies Across Remote Databases On Distributed Networks?**

**Answer :**

Oracle does not manage dependencies across remote schema objects on distributed networks other than local-procedure-to- remote procedure dependencies. It only manages dependencies among stored procedures (procedures, functions, and packages) using time-stamped checking or signature checking. If there is a local view based on a remote table, and the remote table definition is altered, the local view is not invalidated.

1. **Question377. Does A Change To A Remote Procedure Automatically Change The Status Of Dependent Objects In Local Database To Invalid?**

**Answer :**

No, a change to a remote procedure does not automatically change the status of dependent objects in local database to invalid.

1. **Question378. How Is Time-stamped Checking Used For The Management Of Remote Dependencies?**

**Answer :**

Time-stamped checking is based on the fact that every program unit carries a time stamp, which is set when it is created or recompiled. Whenever a program unit or a schema object is altered, all its dependent units are invalidated and must be recompiled before they can be executed. The time stamp is recorded in the p-code of the procedure.  
  
The time stamp of the remote procedure is also stored in the p-code of the local procedure when it is compiled. A local procedure, which refers to a remote procedure, is invalidated if the remote procedure is recompiled after the local procedure is compiled. When the time stamp of the remote procedure, which is stored in the local procedure, is compared with the time stamp of the recompiled remote procedure, both the time stamps differ. This invalidates the local procedure and returns a runtime error.

1. **Question379. What Parameter Is Used To Determine Whether The Mode Of Checking For The Management Of Remote Dependencies Is Time-stamped Or Signature Checking?**

**Answer :**

The parameter used to determine if the mode of checking for the management of remote dependencies is time-stamped or signature checking is set in the init.ora parameter file as REMOTE\_DEPENDENCIES\_MODE = value, where value can be TIMESTAMP or SIGNATURE.

1. **Question380. What Is The Meaning Of Signature Of A Procedure?**

**Answer :**

The signature of a procedure refers to:

* + The name of the procedure
  + The datatype of the parameters
  + The modes of the parameters
  + The number of parameters
  + The datatype of the return value in case of function

1. **Question381. What Is The Disadvantage Of Time-stamped Checking?**

**Answer :**

The disadvantage of time-stamped checking is that it causes unnecessary recompilation of the dependent objects across networks when it may not be required, leading to the degradation of system performance.

1. **Question382. How Is Signature Checking Better Than The Time-stamped Checking?**

**Answer :**

Signature checking eliminates the problem of unnecessary recompilation of the program units across networks. The signature of the remote procedure is stored in the p-code of the local procedure. In signature checking, the signature of the remote procedure is compared and not the time stamp. If the remote program is modified and recompiled but its signature does not change, then the local procedure can be executed without any runtime error. Therefore, signature checking does not degrade the system performance by repetitive compilation of the local procedure.

1. **Question383. What Happens If The Time-stamped Checking Is Used For Dependency Checking Across Remote Databases In Different Time Zones?**

**Answer :**

The time stamp of the remote procedure and the local procedure will not match and the dependency checking will not be successful. Therefore, while working on remote databases across time zones, signature checking should be used.

1. **Question384. What Happens If A Private Referenced Table Is Dropped, But A Public Table With The Same Name Exists?**

**Answer :**

When a private table, which is referenced by a dependent procedure, is dropped, the dependent procedure is invalidated. When the procedure is recompiled and a public table of the same name exists with all the columns required by the procedure, then the procedure will recompile successfully but refer to the public table.

1. **Question385. Give A Few Guidelines To Minimize The Dependency Failures ?**

**Answer :**

A few guidelines that would minimize the dependency failures are as follows:

* + Refer to tables indirectly using views
  + Include column list with INSERT statements
  + Query with SELECT \* notation
  + Declare records with the %ROWTYPE attribute
  + Declare variables with the %TYPE attribute
  + Use packages for procedures and functions

1. **Question386. Can A Table Or Synonym Be Invalidated?**

**Answer :**

A table or synonym can never be invalidated; only dependent objects are invalidated.

1. **Question387. Suppose A Schema 2 Has Granted Necessary Privileges To The Schema 1 Through Roles. Is It Possible For A Program Unit In The Schema 1 To Refer The Database Objects In The Schema 2 ?**

**Answer :**

No, a program unit in the schema 1 cannot reference the database objects in the schema 2, even if the schema 2 has granted necessary privileges to the schema 1 through roles. This is because granting privileges through roles does not work for program units. They can reference other schemp objects only if the necessary privilege has been granted directly to the schema.

1. **Question388. Suppose A Schema A Has The Create Any Trigger Privilege And The Select Privilege On The Table X In The Schema B. Can The Schema A Create A Trigger On The Table X?**

**Answer :**

Yes, schema A can create a trigger on the table X of schema B.

1. **Question389. Why Should Column Names Be Used In The Insert Statements?**

**Answer :**

The INSERT statements should have column names so that if any unrelated change to the table is made after the program unit is compiled, automatic recompilation should succeed.

1. **Question390. In The Time-stamped Checking Mode, Which Value Of The Two Program Units Is Compared To Determine The Dependency Checking?**

**Answer :**

The LAST\_DDL\_TIME column of the USER\_OBJECTS data dictionary view records the last time a SQL statement was issued to change the structure of the objects. This column is compared to determine if the referenced program unit has been changed since the dependent unit was last compiled.

1. **Question391. If The Return Type Of A Remote Function Is Changed And The System Is In The Signature Mode Of The Dependency Checking, Then Does It Cause The Local Procedure To Be Invalidated?**

**Answer :**

Yes, the status of the local procedure will be set to invalid if the RETURN type of the remote function is changed because the signature of the function is changing.

1. **Question392. What Is A System Privilege?**

**Answer :**

A system privilege is the right to perform a particular action in the database. Privileges are granted to a schema. They may be granted to a role, which in turn is granted to schemas.

1. **Question393. What Is The Meaning Of Granting A Privilege To Public?**

**Answer :**

Granting a privilege to PUBLIC means it is granted to all the schemas in the database that already exist or created in the future.

1. **Question394. What System Privileges Are Required By An End User?**

**Answer :**

The end user may not require any system privilege except the CREATE ANY SESSION, which grants them the privilege to log in as an Oracle user.

1. **Question395. What Is The Difference Between The Create Table And Create Any Table Privileges?**

**Answer :**

The CREATE TABLE privilege grants the schema to create tables in its own schema; however, CREATE ANY TABLE grants the privilege to create any table owned by any other schema as well.

1. **Question396. What Is The Meaning Of Granting System Privilege With Admin Option?**

**Answer :**

Granting system privilege WITH ADMIN OPTION means that the schema can in turn grant the same privilege to any other schema.

1. **Question397. Suppose A Schema A Grants The Create Any Trigger With Admin Option Privilege To A Schema B And The Schema B Grants The Create Any Trigger Privilege To A Schema C. If The Schema A Revokes This Privilege On The Schema B, Then Does The Privilege On Schema C Be Revoked As Well?**

**Answer :**

No, the privilege on the schema C will not be revoked automatically; only the privilege on the schema B is revoked.

1. **Question398. What Is The Advantage Of Using Roles For System Privilege?**

**Answer :**

Roles are easy to manage and helpful in granting multiple system privileges to schemas. Any privilege can be added to a role or removed from a role. This would automatically affect all the schemas with that role.

1. **Question399. What Does Create Any Procedure System Privilege Grant To A Schema?**

**Answer :**

The CREATE ANY PROCEDURE system privilege allows the schema to create any package, procedure, or function within its schema or owned by any other schema.

1. **Question400. Can A Schema A With The Create Any Procedure Privilege Alter Or Modify A Package X In A Schema B?**

**Answer :**

No, a schema A with the CREATE ANY PROCEDURE privilege cannot alter or modify a package X in the schema B. The schema must have the ALTER ANY PROCEDURE system privilege for altering or modifying any program unit.

1. **Question401. What Is The Difference Between The Owner Rights Program Unit And The Invoker Rights Program Unit?**

**Answer :**

A program unit created as an owner rights program unit executes according to the privileges that the owner of the program unit have at the time of execution; whereas, a program unit created as an invoker rights program unit executes according to the privileges that the invoker of the program unit have at the time of execution.

1. **Question402. Which Keyword Is Used To Define The Program Unit As An Invoker Rights Program Unit?**

**Answer :**

The keyword AUTHID CURRENT\_USER is used to define a program unit as an invoker rights program unit.

1. **Question403. Write A Statement To Remove Create Any View Privilege On A Schema Sample\_schema ?**

**Answer :**

The statement to remove the CREATE ANY VIEW privilege on the schema SAMPLE\_SCHEMA is as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/schema.png

1. **Question404. What Are Indirectly Granted Privileges?**

**Answer :**

Indirectly granted privileges are those privileges that have been granted to a role, which in turn granted to a schema through a role.

1. **Question405. Which Data Dictionary View Is Used To View Information About All The Privileges That Have Been Made By And To The Current Schema?**

**Answer :**

The USER\_TAB\_PRIVS data dictionary view is used to view information about all the privileges that have been made by and to the current schema.

1. **Question406. Which Data Dictionary View Shows All Directly And Indirectly Granted System Privileges In Effect To A Schema?**

**Answer :**

The SESSION\_PRIVS data dictionary view shows all the directly and indirectly granted system privileges to a schema.

1. **Question407. Suppose A Package Pack.a With A Procedure Proc\_1 Exists In The Schema Sample\_one\_schema. If The Schema Sample\_two\_schema Needs To Be Given Privilege To Execute The Procedure Proc\_1, Which Privilege Has To Be Granted ?**

**Answer :**

The statement is as follows:

GRANT EXECUTE ON PACK\_A TO SAMPLE\_TWO\_SCHEMA

The privilege is always granted on the whole package and not on the individual procedures or functions.

1. **Question408. Which Privilege Is Required To Be Able To Drop A Trigger?**

**Answer :**

The CREATE TRIGGER privilege is required to create, alter, and drop triggers in a schema.

1. **Question409. Suppose The Trig\_1 Trigger On The Tab\_1 Table Exists In The Schema A, And The Schema B Has Privilege To Access The Tab\_1 Table. What Additional Privilege Is Required By Schema B To Be Able To Execute The Trig\_1 Trigger?**

**Answer :**

No additional privilege is required to execute the trigger as the schema B already has access privilege to the triggered table, TAB\_1.

1. **Question410. Write A Pl/sql Statement To Show The Privileges On A Procedure Proc\_1 Within The Schema A ?**

**Answer :**

The statement is given as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/proc.png

1. **Question411. Which Column In The User\_sys\_privs View Determines Whether The System Privilege Is With Admin Option ?**

**Answer :**

The ADMIN OPTION column in the USER\_SYS\_PRIVS view is a YES/NO column that indicates if the privilege was granted WITH ADMIN OPTION.

1. **Question412. Which Data Dictionary View Will Give The Owner Name Along With The Privileges On A Schema Object?**

**Answer :**

The USER\_TABS\_PRIVS data dictionary view gives the list of system and object privileges on the object along with the owner name of the object.

1. **Question413. Explain The Grant And Revoke Dcl Statements ?**

**Answer :**

DCL statements control the security of the database. The two statements, GRANT and REVOKE, are used to give permissions to users to access the different schema objects; therefore, enabling security and access control on the database.

1. **Question414. Suppose A Procedure Proc\_1 Refers To A Procedure Proc\_2 Within The Pack\_1 Package. If The Definition Of The Proc\_2 Procedure Is Changed And It Is Recompiled Again, Then Does The Proc\_1 Procedure Is Invalidated ?**

**Answer :**

If the declaration of the proc\_2 procedure is not altered in the package specification, then the proc\_1 procedure is not invalidated. However, if the specification of the procedure is altered, then the dependent procedure gets the invalid status.

1. **Question415. Give The Sql Statement To Drop The Package Specification For Pack\_a Without Dropping The Package Body ?**

**Answer :**

It Is not possible to drop a package specification and leave the package body. If the package is dropped, the whole package is removed from a database.

1. **Question416. Is It Possible To Create A Package Body And Store It In A Database?**

**Answer :**

Yes, it is possible to create a package body and store it in the database, although it would be in an invalid state and cannot be executed.

1. **Question417. Give A Sql Statement To Grant Access On The Procedure Proc\_sample In The Schema, Schema\_a.**

**Answer :**

The SQL statement is given as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/grant.png

1. **Question418. Give A Sql Statement So That The Procedure Proc\_1 In The Package Pack\_sample Is Granted Access To All The Users In The Database ?**

**Answer :**

The SQL statement is given as follows:

https://www.wisdomjobs.com/userfiles/image/interview%20question%20images/pl%20sql/query%203.png

1. **Question419. Do Stored Procedures And Functions Within The Same Schema Share The Same Namespace?**

**Answer :**

Yes, stored procedures and functions within the same schema share the same namespace.