**Oracle Interview Questions and Answers**

**SQL**

**Q 1. Differentiate between Primary and Unique key.**

**Ans.** A primary key uniquely identifies each row in table and does not allow NULL value. A unique key does not allow duplicate value, but permits NULL. There can be only one primary key in a table, but any number of unique keys are allowed.

**Q 2. Differentiate between TRUNCATE and DELETE command.**

**Ans.** TRUNCATE is DDL command that deletes all rows from table, resets the High Water Mark (which indicates the last block used) and does not produce rollback data, as truncate operation cannot be rolled back (due to auto commit of DDL). DELETE is DML command that allows deleting rows conditionally and the operation can be rolled back.

**Q 3. Differentiate between CHAR and VARCHAR2 data types.**

**Ans.** CHAR is fixed length meaning, if we assign a 5 character value to a CHAR (10) column, it stores the value appended with 5 blank characters, while VARCHAR2 standing for variable characters, allows storing variable length of characters as per value w/o blank characters appended, subject to max specified length.

**Q.4. Define constraint and give various types as allowed in Oracle.**

**Ans.** A constraint is a rule or condition that determines, what data is allowed to be stored in a column. There are 5 types of constraints allowed in Oracle.

1. PRIMARY – Identifies each row uniquely
2. UNIQUE – Allows only unique values in column
3. NOT NULL – Enforces presence of value in column (column can’t be empty)
4. CHECK – Specifies a condition that must be satisfied by value to be stored in column
5. REFERENCES – Specifies a foreign key that links the table to a parent table

**Q. 5. What is referential integrity?**

**Ans.** Referential integrity refers to parent child relationship between two tables by means of primary and foreign key, which ensures that the foreign key value is always present in parent, and parent row can’t be deleted if child rows exist, unless an ON DELETE rule is specified.

**Q. 6. What is ON DELETE rule?**

**Ans.** ON DELETE rule is specified with foreign key (REFERENCES) constraint which ensures removal of child rows along with parent row if CASCADE option is used, else sets the foreign key value of respective child rows to NULL if SET NULL option is used (allowing child rows to be preserved w/o parent row)

**Q. 7. What is the purpose of DEFAULT keyword?**

**Ans.** DEFAULT keyword is used to specify a default value to be inserted in a column, when a value is not explicitly given for it during INSERT command. It thus prevents the column being NULL. It is used in CREATE TABLE command as well as in INSERT and UPDATE commands.

**Q. 8. What is a subquery?**

**Ans.** A subquery, also called as inner or nested query, is a query within query that provides input to outer query using which the outer query retrieves required rows.

**Q.9. What is a co-related subquery?**

**Ans.** A co-related subquery uses a column from outer query to retrieve and provide input to outer query and fires once for each outer query row. It works as follows:

1. Outer query retrieves values and passes it to inner query
2. Inner query in turn retrieves required values and passes them back to outer query
3. Outer query then filters the row retrieved by it based on the values received from inner query
4. This continues till all rows from outer query are processed

**Q. 10. What is the use of SAVEPOINT command?**

**Ans.** SAVEPOINT is used in a transaction, where it defines a point (sort of like a bookmark) to mark the effect of DML, that needs to be retained during a ROLLBACK operation, as the ROLLBACK TO <savepoint> command will only undo effects of DML that follows the savepoint. The effects of DML before the savepoint are thus preserved.

**Q. 11. How many LONG columns can we have in a table?**

**Ans.** There can be only one LONG column per table. However LOBs are recommended now as there can be multiple columns of LOB type and storage capacity is much more than LONG.

**Q. 12.**  **Can we perform DML on views? If yes, then under what conditions?**

**Ans.** If a view is based on only one base table and selects all mandatory columns from it and is a simple view (not having calculated columns, group by etc.) then it can be updated using DML. It should also not have read only option used while creating.

**Q. 13. Explain different types of joins.**

**Ans.** There are 4 basic types of joins:

1. EQUIJOIN – uses **= (equal to)** operator to compare common column(s) between tables to join
2. NON-EQUIJOIN – uses any **not equal to** operator for comparing column(s) and join tables
3. OUTER JOIN – uses **(+)** operator with equijoin to include all rows that match as well as do not match the joining condition(s)
4. SELF JOIN – joins table to itself, based on columns that reference each other in same table

**Q. 14. Differentiate between column and table level constraints.**

**Ans.** A column level constraint applies to a single column and can’t reference other columns from that table. A table level constraint can refer multiple columns from same table and is required to define compound keys and CHECK constraints that involve multiple columns.

**Q. 15. What are Pessimistic Lock and Optimistic Lock?**

**Ans.** Optimistic locking is a strategy where you read a record, take note of a version number and check that the version hasn’t changed before you write the record back. If the record is dirty (i.e. different version to yours), then you abort the transaction and the user can re-start it.

Pessimistic locking is when you lock the record for your exclusive use until you have finished with it. It has much better integrity than optimistic locking but requires you to be careful with your application design to avoid Deadlocks.

**Q. 16. What is the purpose of SELECT…FOR UPDATE statement?**

**Ans.** This command is used to lock the rows exclusively so that other users can’t change them until you finish the transaction with an explicit COMMIT / ROLLBACK. It allows user to obtain pessimistic lock.

**Q. 17. What is the difference between UNION and UNION ALL?**

**UNION****Ans.** The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type. With UNION, only distinct values are selected.

**UNION ALL**The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values.

The difference between UNION and UNION ALL is that UNION ALL will not eliminate duplicate rows, instead it just pulls all rows from all the tables fitting your query specifics and combines them into a table.

**Q. 18. What are the different types of index?**

**Ans.** There are two basic types as B\*Tree index and Bitmap index. In B\*Tree index, we have Regular (normal), Reverse key and Function based index types.

**Q. 19. What is the difference between HAVING and WHERE clause?**

**Ans.** HAVING clause is used with GROUP BY clause to filter groups based on a condition, and can use aggregate functions. WHERE clause is used to apply a filter condition on individual rows and can’t use aggregate functions. Also HAVING clause can only appear in SELECT while WHERE clause is supported by UPDATE and DELETE as well.

**Q. 20. What is a pseudo column? Give some examples**

**Ans.** It is a column that is not an actual column in the table, but is defined at system level and usable with any query. E.g. USER, UID, SYSDATE, ROWNUM, ROWID, NULL, AND LEVEL.

**Q. 21. What is an instance and database with respect to an Oracle server?**

**Ans.** An instance is a collection of memory structures and background processes used for accessing and managing a database. It is logical and temporary. A database is a collection of physical files that store actual data and logical structures such as block, extents, segments and tablespaces. Unlike instance, database is physical and permanent. One instance can access only one database at a time, however a single database may be accessed by multiple instances.

**Q. 22. What is an inline view?**

**Ans.** An inline view is a subquery that appears in FROM clause and acts like a table from which you can SELECT data as well as create joins. Unlike a normal view, it is not stored in database. It is useful in Top-N queries and joins requiring calculated columns and complex conditions for display/selection.

**Q. 23. Explain TIMESTAMP data type.**

**Ans.** A TIMESTAMP data type is used to store date and time data that are precise to fractional seconds. There are 3 types as TIMESTAMP, TIMESTAMP WITH TIMEZONE and TIMESTAMP WITH LOCAL TIMEZONE.

**Q.24. Explain LOBs.**

**Ans.** LOB stands for Large Objects and allows large data storage of type character and binary. There are CLOB, NCLOB, BLOB and BFILE as LOB types that allow multimedia data to be stored in database.

**Q. 25. Explain CUBE and ROLLUP operators.**

**Ans.** CUBE and ROLLUP are used in GROUP BY clause to produce super aggregates. The CUBE operation groups the selected rows based on the values of all possible combinations of expressions in the specification, and returns a single row of summary information for each group. You can use the CUBE operation to produce cross-tabulation values. The ROLLUP operation in the groups the selected rows based on the values of the first n, n-1, n-2, ... 0 expressions in the GROUP BY specification, and returns a single row of summary for each group. You can use the ROLLUP operation to produce subtotal values by using it with the SUM function. When used with SUM, ROLLUP generates subtotals from the most detailed level to the grand total. Aggregate functions such as COUNT can be used to produce other kinds of super aggregates.

**Q. 26. Explain hierarchical query.**

**Ans.** A hierarchical query is useful to display hierarchical data such as manager and subordinates, categories and products etc. It is created using a CONNECT BY clause in SELECT.

**Q. 27. What is a transaction?**

**Ans.** A transaction is a set of SQL statements between any two COMMIT and ROLLBACK statements.

**Q. 28. Explain PL/SQL block.**

**Ans.** A PL/SQL block contains various sections and combines PL and SQL statements to perform logical operations on data as a program. There are two types: Anonymous and Named. Anonymous have no name and are stored and executed locally on a client machine. Named blocks are stored centrally in a database and are executed when a client calls them. A PL/SQL block has 3 sections: DECLARE, executable (BEGIN..END) and EXCEPTION. Only executable section is mandatory.

**Q. 29. What is implicit cursor and how is it used by Oracle?**

**Ans.** An implicit cursor is a cursor which is internally created and managed by Oracle. It is created by Oracle for each individual SQL statement executed within a PL/SQL block. It is named as SQL and you can refer its attributes %FOUND, %NOTFOUND, and %ROWCOUNT to check the result of a DML statement.

**Q. 30. Explain various types of exceptions.**

**Ans.** There are 3 types of exceptions that are, PRE-DEFINED, NON-PRE-DEFINED and USER DEFINED. PRE-DEFINED are having Oracle defined names for most common exceptions such as no data found and are implicitly raised. NON-PRE-DEFINED are implicitly raised as errors, but have no names, so require programmer defined name and a PRAGMA to associate the error number. USER DEFINED are explicitly named, raised and handled by programmer.

**Q. 31. What is the difference between a procedure and a function?**

**Ans.** Functions always return a single value, whereas procedures can return multiple values via OUT parameters if required. Functions must be called as part of some executable statement while procedures can be called via EXECUTE/CALL statement. Logically functions are used to compute and return result; while procedures are used for performing some actions.

**Q. 32. What are various types of parameter passing modes in procedures and functions?**

**Ans.** There are 3 modes of parameter passing, IN, OUT and IN OUT. IN is used to pass read only values as literals and variables, OUT is used to pass write only variables (no literals) to be passed back to calling environment and IN OUT are used for passing to and fro a value through variable.

**Q. 33. What is a package?**

**Ans.** A package is a collection of data types, procedures and functions that are logically grouped as an application. A package consists of a specification and a body, where specification includes all public elements that are visible to package users and the body consists of all private elements that are not visible to package users including the logic implementation of procedures and functions of package.

**Q. 34. What are the advantages of package?**

1. **Ans.** Package offers following advantages:
2. Let you grant privileges more efficiently.
3. Let you modify package objects without recompiling dependent schema objects.
4. Enable Oracle Database to read multiple package objects into memory at once.
5. Can contain global variables and cursors that are available to all procedures and functions in the package.
6. Let you overload procedures or functions. Overloading a procedure means creating multiple procedures with the same name in the same package, each taking arguments of different number or datatype.

**Q. 35. Explain Explicit Cursor.**

**Ans.** An explicit cursor is programmer defined and managed and allows to capture output of SELECT query that can be processed one row at a time. Essentially, cursor acts like a pointer using which you can point to a context area where output of SELECT is stored.

**Q. 36. What are the steps to use Explicit Cursor?**

**Ans.** The steps are:

1. Declare cursor using CURSOR <name> IS SELECT
2. Define a row type variable as per cursor
3. Open cursor
4. Set up a loop to fetch rows from cursor until all rows are processed
5. Close cursor

Alternately you can use a CURSOR FOR LOOP to process cursor with automatic open, fetch and close operations.

**Q. 37. What is a REF Cursor or Cursor Variable?**

**Ans.** Cursor variables are like pointers to result sets. You use them when you want to perform a query in one subprogram, and process the results in a different subprogram (possibly one written in a different language). A cursor variable has datatype REF CURSOR, and you might see them referred to informally as REF CURSORs. Unlike an explicit cursor, which always refers to the same query work area, a cursor variable can refer to different work areas. You cannot use a cursor variable where a cursor is expected, or vice versa.

**Q. 38. Differentiate between weak and strong REF cursor.**

**Ans.** A weak REF cursor does not specify a RETURN row type, so is flexible to go with any query. A strong REF cursor specifies a RETURN row type and so can go with only queries that return specific row type. Weak cursor produces error at runtime during mismatch of fetch while strong one produces error at compile time.

**Q. 39. What are SQLCODE and SQLERRM and why are they important?**

**Ans.** 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.

**Q. 40. What is the use of EXECUTE IMMEDIATE statement?**

**Ans.** This statement is used for executing Native Dynamic SQL in PL/SQL block. It lets you avoid using DBMS\_SQL procedural approach to executing dynamic SQL.

**Q. 41. What are collections in PL/SQL?**

**Ans.** A collection allows you to store multiple elements of same type and access them by means of an index value. There are 3 types of collections in PL/SQL, INDEX BY TABLE, VARRAY and NESTED TABLE.

**Q. 42. What is the difference between INDEX BY TABLE and VARRAY?**

**Ans.** An INDEX BY TABLE uses range of BINARY\_INTEGER as index, thus allowing virtually unlimited storage of values, supports negative index values and is a sparse structure, meaning there can be gaps in index value. A VARRAY on the other hand is a dense structure that does not allow gaps in index value, is fixed in size and supports no negative index value.

**Q. 43. What are persistent v/s transient objects?**

**Ans.** Persistent objects are created in SQL using CREATE command and are permanently stored in schema, until removed. Once created they can be used anywhere needed. Transient objects are created in PL/SQL block and exist for the life and scope of the defining block.

**Q. 44. What is the size limit of VARCHAR2 variable in PL/SQL?**

**Ans.** In PL/SQL, VARCHAR2 size limit is 32767 characters.

**Q. 45. What is a trigger?**

**Ans.** A trigger is a procedure that fires automatically in response to an event. Events can be DML (INSERT, UPDATE, DELETE), DDL (CREATE, ALTER, DROP), database events (LOGON/LOGOFF) or system events (STARTUP/SHUTDOWN).

**Q. 46. How many triggers can you create for a table?**

**Ans.** We can create total 12 triggers for a table in DML type as 3 events (INSERT/UPDATE/DELETE), a BEFORE and AFTER for each event and statement or row level for each timing.

**Q. 47. What are :NEW and :OLD values?**

**Ans.** :NEW and :OLD values are accessible only in row level triggers and are used to access values in statement and table respectively. For INSERT, only :NEW values exist, for UPDATE, both :NEW and :OLD values exist, while for DELETE, only :OLD values exist. They are used for validation and calculation of column values as per event and timing.

**Q. 48. How do you encrypt PL/SQL code to prevent it from being viewed or changed?**

**Ans.** Use the WRAP utility to compile a .SQL or .PLS file in text format to produce a .PLB file which is binary. When .PLB file is executed, the source code in USER\_SOURCE view is encrypted so is not readable and modifiable.

**Q. 49. What is mutating table error?**

**Ans.** When a row level trigger tries to read a table being modified by DML on which the trigger has fired, it results in mutating table error as read consistency cannot be guaranteed.

**Q. 50. How do you resolve mutating table error?**

**Ans.** This can be solved using a statement level trigger that, with the help of packaged variables can read and store values from mutating table, which can then be used by row level trigger. Another method is to use AUTONOMOUS transaction within row level trigger to start an independent transaction which can read mutating table w/o error.