

Q. What is a Join?

A **join** is a query that combines rows from two or more tables, views, or materialized views ("snapshots"). Oracle performs a join whenever multiple tables appear in the queries FROM clause. The query's select list can select any columns from any of these tables. If any two of these tables have a column name in common, you must qualify all references to these columns throughout the query with table names to avoid ambiguity.

Q. What are join conditions?

Most join queries contain WHERE clause conditions that compare two columns, each from a different table. Such a condition is called a **join condition**. To execute a join, Oracle combines pairs of rows, each containing one row from each table, for which the join condition evaluates to TRUE. The columns in the join conditions need not also appear in the select list.

Q. What is an equijoin?

An **equijoin** is a join with a join condition containing an equality operator. An equijoin combines rows that have equivalent values for the specified columns.

Eg:

```
Select ename, job, dept.deptno, dname
      From emp, dept
      Where emp.deptno = dept.deptno;
```

Q. What are self joins?

A **self join** is a join of a table to itself. This table appears twice in the FROM clause and is followed by table aliases that qualify column names in the join condition.

Eg:

```
SELECT e.ename || ' works for ' || e2.name "Employees and their Managers"
      FROM emp e1, emp e2
      WHERE e1.mgr = e2.empno;
```

ENAME	EMPNO	MGR
BLAKE	12345	67890
KING	67890	22446

Result: BLAKE works for KING

Q. What is an Outer Join?

An outer join extends the result of a simple join. An outer join returns all rows that satisfy the join condition and those rows from one table for which no rows from the other satisfy the join condition. Such rows are not returned by a simple join. To write a query that performs an outer join of tables A and B and returns all rows from A, apply the outer join operator (+) to all columns of B in the join condition.

For all rows in A that have no matching rows in B, Oracle returns null for any select list expressions containing columns of B.

Outer join queries are subject to the following rules and restrictions:

- ❖ The (+) operator can appear only in the WHERE clause or, in the context of left correlation (that is, when specifying the TABLE clause) in the FROM clause, and can be applied only to a column of a table or view.

- ❖ If A and B are joined by multiple join conditions, you must use the (+) operator in all of these conditions. If you do not, Oracle will return only the rows resulting from a simple join, but without a warning or error to advise you that you do not have the results of an outer join.
- ❖ The (+) operator can be applied only to a column, not to an arbitrary expression. However, an arbitrary expression can contain a column marked with the (+) operator.
- ❖ A condition containing the (+) operator cannot be combined with another condition using the OR logical operator.
- ❖ A condition cannot use the IN comparison operator to compare a column marked with the (+) operator with an expression.
- ❖ A condition cannot compare any column marked with the (+) operator with a subquery.

If the WHERE clause contains a condition that compares a column from table B with a constant, the (+) operator must be applied to the column so that Oracle returns the rows from table A for which it has generated NULLs for this column. Otherwise Oracle will return only the results of a simple join.

In a query that performs outer joins of more than two pairs of tables, a single table can be the null-generated table for only one other table. For this reason, you cannot apply the (+) operator to columns of B in the join condition for A and B and the join condition for B and C.

Set Operators: UNION [ALL], INTERSECT, MINUS

Set operators combine the results of two component queries into a single result. Queries containing set operators are called compound queries.

The number and datatypes of the columns selected by each component query must be the same, but the column lengths can be different.

If you combine more than two queries with set operators, Oracle evaluates adjacent queries from left to right. You can use parentheses to specify a different order of evaluation.

Restrictions:

- ❖ These set operators are not valid on columns of type BLOB, CLOB, BFILE, varray, or nested table.
- ❖ The UNION, INTERSECT, and MINUS operators are not valid on LONG columns.
- ❖ To reference a column, you must use an alias to name the column.
- ❖ You cannot also specify the `for_update_clause` with these set operators.
- ❖ You cannot specify the `order_by_clause` in the subquery of these operators.

Table 3–9 Set Operators

Operator	Returns
UNION	All rows selected by either query.
UNION ALL	All rows selected by either query, including all duplicates.
INTERSECT	All distinct rows selected by both queries.
MINUS	All distinct rows selected by the first query but not the second.

All set operators have equal precedence. If a SQL statement contains multiple set operators, Oracle evaluates them from the left to right if no parentheses explicitly specify another order.

The corresponding expressions in the select lists of the component queries of a compound query must match in number and datatype. If component queries select character data, the datatype of the return values are determined as follows:

- ❖ If both queries select values of datatype CHAR, the returned values have datatype CHAR.

- ❖ If either or both of the queries select values of datatype VARCHAR2, the returned values have datatype VARCHAR2.

Q. What is a UNION?

The UNION operator eliminates duplicate records from the selected rows. We must match datatype (using the TO_DATE and TO_NUMBER functions) when columns do not exist in one or the other table.

Q. What is UNION ALL?

The UNION ALL operator does not eliminate duplicate selected rows.

Note: The UNION operator returns only distinct rows that appear in either result, while the UNION ALL operator returns all rows.

Q. What is an INTERSECT?

The INTERSECT operator returns only those rows returned by both queries. It shows only the distinct values from the rows returned by both queries.

Q. What is MINUS?

The MINUS operator returns only rows returned by the first query but not by the second. It also eliminates the duplicates from the first query.

Note: For compound queries (containing set operators UNION, INTERSECT, MINUS, or UNION ALL), the ORDER BY clause must use positions, rather than explicit expressions. Also, the ORDER BY clause can appear only in the last component query. The ORDER BY clause orders all rows returned by the entire compound query.

Q. How many types of Sql Statements are there in Oracle?

There are basically 6 types of sql statements. They are:

a) Data Definition Language (DDL)

The DDL statements define and maintain objects and drop objects.

b) Data Manipulation Language (DML)

The DML statements manipulate database data.

c) Transaction Control Statements

Manage change by DML

d) Session Control

Used to control the properties of current session enabling and disabling roles and changing. E.g. Alter Statements, Set Role

e) System Control Statements

Change Properties of Oracle Instance. E.g. Alter System

f) Embedded Sql

Incorporate DDL, DML and TCS in Programming Language. E.g. Using the Sql Statements in languages such as 'C', Open, Fetch, execute and close

Q) What is a Transaction in Oracle?

A transaction is a Logical unit of work that comprises one or more SQL Statements executed by a single User. According to ANSI, a transaction begins with first executable statement and ends when it is explicitly committed or rolled back.

A transaction is an atomic unit.

Q. What are some of the Key Words Used in Oracle?

Some of the Key words that are used in Oracle are:

a) **Committing**: A transaction is said to be committed when the transaction makes permanent changes resulting from the SQL statements.

b) **Rollback**: A transaction that retracts any of the changes resulting from SQL statements in Transaction.

c) **SavePoint**: For long transactions that contain many SQL statements, intermediate markers or savepoints are declared. Savepoints can be used to divide a transaction into smaller points.

We can declare intermediate markers called savepoints within the context of a transaction. Savepoints divide a long transaction into smaller parts. Using savepoints, we can arbitrarily mark our work at any point within a long transaction. We then have the option later of rolling back work performed before the current point in the transaction but after a declared savepoint within the transaction.

For example, we can use savepoints throughout a long complex series of updates so that if we make an error, we do not need to resubmit every statement.

d) **Rolling Forward**: Process of applying redo log during recovery is called rolling forward.

e) **Cursor**: A cursor is a handle (name or a pointer) for the memory associated with a specific statement. A cursor is basically an area allocated by Oracle for executing the Sql Statement. Oracle uses an implicit cursor statement for Single row query and Uses Explicit cursor for a multi row query.

f) **System Global Area (SGA)**: The SGA is a shared memory region allocated by the Oracle that contains Data and control information for one Oracle Instance. It consists of Database Buffer Cache and Redo log Buffer. (KPIT Infotech, Pune)

g) **Program Global Area (PGA)**: The PGA is a memory buffer that contains data and control information for server process.

g) **Database Buffer Cache**: Database Buffer of SGA stores the most recently used blocks of database data. The set of database buffers in an instance is called Database Buffer Cache.

h) **Redo log Buffer**: Redo log Buffer of SGA stores all the redo log entries.

i) **Redo Log Files**: Redo log files are set of files that protect altered database data in memory that has not been written to Data Files. They are basically used for backup when a database crashes.

j) **Process**: A Process is a 'thread of control' or mechanism in Operating System that executes series of steps.

Q. What are Procedure, functions and Packages?

Procedures and functions consist of set of PL/SQL statements that are grouped together as a unit to solve a specific problem or perform set of related tasks.

Procedures do not return values while Functions return one Value.

Packages: Packages provide a method of encapsulating and storing related procedures, functions, variables and other Package Contents

Q. What are Database Triggers and Stored Procedures?

Database Triggers: Database Triggers are Procedures that are automatically executed as a result of insert in, update to, or delete from table.

Database triggers have the values old and new to denote the old value in the table before it is deleted and the new indicated the new value that will be used. DT is useful for implementing complex business rules which cannot be enforced using the integrity rules. We can have the trigger as Before trigger or After Trigger and at Statement or Row level.

e.g.: operations insert, update ,delete 3

before ,after 3*2 A total of 6 combinations

At statement level(once for the trigger) or row level(for every execution) 6 * 2 A total of 12.

Thus a total of 12 combinations are there and the restriction of usage of 12 triggers has been lifted from Oracle 7.3 Onwards.

Stored Procedures: Stored Procedures are Procedures that are stored in Compiled form in the database. The advantage of using the stored procedures is that many users can use the same procedure in compiled and ready to use format.

Q. How many Integrity Rules are there and what are they?

There are Three Integrity Rules. They are as follows:

- a) Entity Integrity Rule: The Entity Integrity Rule enforces that the Primary key cannot be Null
- b) Foreign Key Integrity Rule: The FKIR denotes that the relationship between the foreign key and the primary key has to be enforced. When there is data in Child Tables the Master tables cannot be deleted.
- c) Business Integrity Rules: The Third Integrity rule is about the complex business processes which cannot be implemented by the above 2 rules.

Q. What are the Various Master and Detail Relationships?

The various Master and Detail Relationship are

- a) No Isolated : The Master cannot be deleted when a child is existing
- b) Isolated : The Master can be deleted when the child is existing
- c) Cascading : The child gets deleted when the Master is deleted.

Q. What are the Various Block Coordination Properties?

The various Block Coordination Properties are:

- a) Immediate

Default Setting. The Detail records are shown when the Master Record are shown.

b) Deferred with Auto Query

Oracle Forms defer fetching the detail records until the operator navigates to the detail block.

c) Deferred with No Auto Query

The operator must navigate to the detail block and explicitly execute a query

Q. What are the Different Optimization Techniques?

The Various Optimization techniques are:

a) Execute Plan: we can see the plan of the query and change it accordingly based on the indexes

b) Optimizer_hint: set_item_property ('DeptBlock',OPTIMIZER_HINT,'FIRST_ROWS');
Select /*+ First_Rows */ Deptno,Dname,Loc,Rowid from dept where (Deptno > 25)

c) Optimize_Sql: By setting the Optimize_Sql = No, Oracle Forms assigns a single cursor for all SQL statements. This slows down the processing because for every time the SQL must be parsed whenever they are executed.

f45run module = my_firstform userid = scott/tiger optimize_sql = No

d) Optimize_Tp:

By setting the Optimize_Tp= No, Oracle Forms assigns separate cursor only for each query SELECT statement. All other SQL statements reuse the cursor.

f45run module = my_firstform userid = scott/tiger optimize_Tp = No

Q. How do you implement the If statement in the Select Statement?

We can implement the if statement in the select statement by using the Decode statement.

e.g select DECODE (EMP_CAT,'1','First','2','Second', Null);

Q. How many types of Exceptions are there?

There are 2 types of exceptions. They are:

a) System Exceptions

e.g. When no_data_found, When too_many_rows

b) User Defined Exceptions

e.g. My_exception exception

When My_exception then

Q. What are the inline and the precompiler directives?

The inline and precompiler directives detect the values directly.

Q. How do you use the same lov for 2 columns?

We can use the same lov for 2 columns by passing the return values in global values and using the global values in the code.

Q. How many minimum groups are required for a matrix report?

The minimum number of groups in matrix report is 4.

Q. What is the difference between static and dynamic lov?

The static lov contains the predetermined values while the dynamic lov contains values that come at run time.

Q. What are the OOPS concepts in Oracle?

Oracle does implement the OOPS concepts. The best example is the Property Classes. We can categorize the properties by setting the visual attributes and then attach the property classes for the objects. OOPS supports the concepts of objects and classes and we can consider the property classes as classes and the items as objects

Q. What is the difference between candidate key, unique key and primary key?

Candidate keys are the columns in the table that could be the primary keys and the primary key is the key that has been selected to identify the rows. Unique key is also useful for identifying the distinct rows in the table.

Q. What is concurrency?

Q. Concurrency is allowing simultaneous access of same data by different users.

Locks useful for accessing the database are:

a) Exclusive

The exclusive lock is useful for locking the row when an insert, update or delete is being done. This lock should not be applied when we do only select from the row.

b) Share lock

We can do the table as Share_Lock and as many share_locks can be put on the same resource.

Q. What are Privileges and Grants?

Privileges are the right to execute a particular type of SQL statements.

E.g. Right to Connect, Right to create, Right to resource

Grants are given to the objects so that the object might be accessed accordingly. The grant has to be given by the owner of the object.

Q. What are Table Space, Data Files, Parameter File and Control Files?

Table Space: The table space is useful for storing the data in the database.

When a database is created two table spaces are created.

a) System Table space: This data file stores all the tables related to the system and dba tables

b) User Table space: This data file stores all the user related tables

We should have separate table spaces for storing the tables and indexes so that the access is fast.

Data Files: Every Oracle Data Base has one or more physical data files. They store the data for the database. Every data file is associated with only one database. Once the Data file is created the size cannot change. To increase the size of the database to store more data we have to add data file.

Parameter Files: Parameter file is needed to start an instance. A parameter file contains the list of instance configuration parameters.

e.g.

db_block_buffers = 500

db_name = ORA7

db_domain = u.s.acme lang

Control Files: Control files record the physical structure of the data files and redo log files. They contain the Db name, name and location of dbs, data files, redo log files and time stamp.

Q. Some of the terms related to Physical Storage of the Data.

The finest level of granularity of the data base is the data blocks.

Data Block : One Data Block correspond to specific number of physical database space

Extent : Extent is the number of specific number of contiguous data blocks.

Segments : Set of Extents allocated for Extents. There are three types of Segments.

- a) Data Segment: Non Clustered Table has data segment data of every table is stored in cluster data segment
- b) Index Segment: Each Index has index segment that stores data
- c) Roll Back Segment: Temporarily store 'undo' information

Q. What are the Pct Free and Pct Used?

Pct Free is used to denote the percentage of the free space that is to be left when creating a table. Similarly Pct Used is used to denote the percentage of the used space that is to be used when creating a table

E.g. Pctfree 20, Pctused 40

Q. What is Row Chaining?

The data of a row in a table may not be able to fit the same data block. Data for row is stored in a chain of data blocks.

Q. What is a 2 Phase Commit?

Two Phase commit is used in distributed data base systems. This is useful to maintain the integrity of the database so that all the users see the same values. It contains DML statements or Remote Procedural calls that reference a remote object.

There are basically 2 phases in a 2 phase commit.

- a) Prepare Phase: Global coordinator asks participants to prepare
- b) Commit Phase: Commit all participants to coordinator to Prepared, Read only or abort Reply

A two-phase commit mechanism guarantees that all database servers participating in a distributed transaction either all commit or all roll back the statements in the transaction. A two-phase commit mechanism also protects implicit DML operations performed by integrity constraints, remote procedure calls, and triggers.

Q. What is the difference between deleting and truncating of tables?

Deleting a table will not remove the rows from the table but entry is there in the database dictionary and it can be retrieved But truncating a table deletes it completely and it cannot be retrieved.

Q. What are mutating tables?

When a table is in state of transition it is said to be mutating. E.g. If a row has been deleted then the table is said to be mutating and no operations can be done on the table except select.

Q. What are Codd Rules?

Codd Rules describe the ideal nature of a RDBMS. No RDBMS satisfies all the 12 codd rules and Oracle Satisfies 11 of the 12 rules and is the only RDBMS to satisfy the maximum number of rules.

Q. What is Normalization?

Normalization is the process of organizing the tables to remove the redundancy. There are mainly 5 Normalization rules.

- | | |
|-----------------|---|
| 1 Normal Form | - A table is said to be in 1st Normal Form when the attributes are atomic |
| 2 Normal Form | - A table is said to be in 2nd Normal Form when all the candidate keys are dependant on the primary key |
| 3rd Normal Form | - A table is said to be third Normal form when it is not dependant transitively |

Q. What is the Difference between a post query and a pre query?

A post query will fire for every row that is fetched but the pre query will fire only once.

Q. How can we delete the duplicate rows in the table?

We can delete the duplicate rows in the table by using the Rowid.

Q. Can U disable database trigger? How?

Yes. With respect to table
`ALTER TABLE TABLE
[DISABLE all_trigger]`

Q. What are pseudocolumns? Name them?

A pseudocolumn behaves like a table column, but is not actually stored in the table. You can select from pseudocolumns, but you cannot insert, update, or delete their values. This section describes these pseudocolumns:

- * CURRVAL
- * NEXTVAL
- * LEVEL
- * ROWID
- * ROWNUM

Q. How many columns can table have?

The number of columns in a table can range from 1 to 254.

Q. Is space acquired in blocks or extents?

In extents.

Q. What is clustered index?

In an indexed cluster, rows are stored together based on their cluster key values. Can not be applied for HASH.

Q. What are the datatypes supported By oracle (INTERNAL)?

Varchar2, Number, Char, MLSLABEL.

Q. What are attributes of cursor?

%FOUND , %NOTFOUND , %ISOPEN,%ROWCOUNT

Q. Can you use select in FROM clause of SQL select ?

Yes.

Q. Describe the difference between a procedure, function and anonymous pl/sql block.

Candidate should mention use of DECLARE statement, a function must return a value while a procedure doesn't have to.

Q. What is a mutating table error and how can you get around it?

This happens with triggers. It occurs because the trigger is trying to modify 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.

Q. Describe the use of %ROWTYPE and %TYPE in PL/SQL.

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

Q. What packages (if any) has Oracle provided for use by developers?

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. If they can mention a few of these and describe how they used them, even better. If they include the SQL routines provided by Oracle, great, but not really what was asked.

Q. Describe the use of PL/SQL 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.

Q. When is a declare statement needed?

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 standalone file if it is used.

Q. 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?

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.

Q. What are SQLCODE and SQLERRM and why are they important for PL/SQL developers?

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

Q. How can you find within a PL/SQL block, if a cursor is open?

Use the %ISOPEN cursor status variable.

Q. How can you generate debugging output from PL/SQL?

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.

Q. What are the types of triggers?

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.

Q. How can variables be passed to a SQL routine?

By use of the & or double && symbol. For passing in variables numbers can be used (&1, &2,...&8) to pass the values after the command into the SQLPLUS session. To be prompted for a specific variable, place the ampersanded variable in the code itself:

“select * from dba_tables where owner=&owner_name;” . Use of double ampersands tells SQLPLUS to resubstitute the value for each subsequent use of the variable, a single ampersand

will cause a reprompt for the value unless an ACCEPT statement is used to get the value from the user.

Q. You want to include a carriage return/linefeed in your output from a SQL script, how can you do this?

The best method is to use the CHR() function (CHR(10) is a return/linefeed) and the concatenation function "||". Another method, although it is hard to document and isn't always portable is to use the return/linefeed as a part of a quoted string.

Q. How can you call a PL/SQL procedure from SQL?

By use of the EXECUTE (short form EXEC) command. You can also wrap the call in a BEGIN END block and treat it as an anonymous PL/SQL block.

Q. How do you execute a host operating system command from within SQL?

By use of the exclamation point "!" (in UNIX and some other OS) or the HOST (HO) command.

Q. You want to use SQL to build SQL, what is this called and give an example?

This is called dynamic SQL. An example would be:

```
set lines 90 pages 0 termout off feedback off verify off
spool drop_all.sql
select 'drop user '||username||' cascade;' from dba_users
where username not in ('SYS','SYSTEM');
spool off
```

Essentially you are looking to see that they know to include a command (in this case DROP USER...CASCADE;) and that you need to concatenate using the '||' the values selected from the database.

Q. What SQLPlus command is used to format output from a select?

This is best done with the COLUMN command.

**Q. You want to group the following set of select returns, what can you group on?
Max(sum_of_cost), min(sum_of_cost), count(item_no), item_no**

The only column that can be grouped on is the "item_no" column, the rest have aggregate functions associated with them.

Q. What special Oracle feature allows you to specify how the cost based system treats a SQL statement?

The COST based system allows the use of HINTs to control the optimizer path selection. If they can give some example hints such as FIRST ROWS, ALL ROWS, USING INDEX, STAR, even better.

Q. You want to determine the location of identical rows in a table before attempting to place a unique index on the table, how can this be done?

Oracle tables always have one guaranteed unique column, the rowid column. If you use a min/max function against your rowid and then select against the proposed primary key you can squeeze out the rowids of the duplicate rows pretty quick. For example:

```
select rowid from emp e
      where e.rowid > (select min(x.rowid)
                      from emp x
                      where x.emp_no = e.emp_no);
```

In the situation where multiple columns make up the proposed key, they must all be used in the where clause.

Q. What is a Cartesian product?

A Cartesian product is the result of an unrestricted join of two or more tables. The result set of a three table Cartesian product will have $x * y * z$ number of rows where x, y, z correspond to the number of rows in each table involved in the join. This occurs if there are not at least n-1 joins where n is the number of tables in a SELECT.

Q. You are joining a local and a remote table, the network manager complains about the traffic involved, how can you reduce the network traffic?

Push the processing of the remote data to the remote instance by using a view to pre-select the information for the join. This will result in only the data required for the join being sent across.

Q. What is the default ordering of an ORDER BY clause in a SELECT statement?

Ascending

Q. What is tkprof and how is it used?

The tkprof tool is a tuning tool used to determine cpu and execution times for SQL statements. You use it by first setting timed_statistics to true in the initialization file and then turning on tracing for either the entire database via the sql_trace parameter or for the session using the ALTER SESSION command. Once the trace file is generated you run the tkprof tool against the trace file and then look at the output from the tkprof tool. This can also be used to generate explain plan output.

Q. What is explain plan and how is it used?

The EXPLAIN PLAN command is a tool to tune SQL statements. To use it you must have an explain_table generated in the user you are running the explain plan for. This is created using the utlxplan.sql script. Once the explain plan table exists you run the explain plan command giving as its argument the SQL statement to be explained. The explain_plan table is then queried to see the execution plan of the statement. Explain plans can also be run using tkprof.

Q. How do you set the number of lines on a page of output? The width?

The SET command in SQLPLUS is used to control the number of lines generated per page and the width of those lines, for example SET PAGESIZE 60 LINESIZE 80 will generate reports that are 60 lines long with a line width of 80 characters. The PAGESIZE and LINESIZE options can be shortened to PAGES and LINES.

Q. How do you prevent output from coming to the screen?

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

Q. How do you prevent Oracle from giving you informational messages during and after a SQL statement execution?

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

Q. How do you generate file output from SQL?

By use of the SPOOL command.

Data Modeler:

Q. Describe third normal form?

Expected answer: Something like: In third normal form all attributes in an entity are related to the primary key and only to the primary key

Q. Is the following statement true or false? Why or why not?

“All relational databases must be in third normal form”

False. While 3NF is good for logical design most databases, if they have more than just a few tables, will not perform well using full 3NF. Usually some entities will be denormalized in the logical to physical transfer process.

Q. What is an ERD?

An ERD is an Entity-Relationship-Diagram. It is used to show the entities and relationships for a database logical model.

Q. Why are recursive relationships bad? How do you resolve them?

A recursive relationship (one where a table relates to itself) is bad when it is a hard relationship (i.e. neither side is a “may” both are “must”) as this can result in it not being possible to put in a top or perhaps a bottom of the table (for example in the EMPLOYEE table you couldn’t put in the PRESIDENT of the company because he has no boss, or the junior janitor because he has no subordinates). These type of relationships are usually resolved by adding a small intersection entity.

Q. What does a hard one-to-one relationship mean (one where the relationship on both ends is “must”)?

This means the two entities should probably be made into one entity.

Q. How should a many-to-many relationship be handled?

By adding an intersection entity table

Q. What is an artificial (derived) primary key? When should an artificial (or derived) primary key be used?

A derived key comes from a sequence. Usually it is used when a concatenated key becomes too cumbersome to use as a foreign key.

Q. When should you consider denormalization?

Whenever performance analysis indicates it would be beneficial to do so without compromising data integrity.

Q. What is a Schema?

Associated with each database user is a schema. A schema is a collection of schema objects. Schema objects include tables, views, sequences, synonyms, indexes, clusters, database links, snapshots, procedures, functions, and packages.

Q. What do you mean by table?

Tables are the basic unit of data storage in an Oracle database. Data is stored in rows and columns.

A row is a collection of column information corresponding to a single record.

Q. Is there an alternative of dropping a column from a table? If yes, what?

Dropping a column in a large table takes a considerable amount of time. A quicker alternative is to mark a column as unused with the SET UNUSED clause of the ALTER TABLE statement. This makes the column data unavailable, although the data remains in each row of the table. After marking a column as unused, you can add another column that has the same name to the table. The unused column can then be dropped at a later time when you want to reclaim the space occupied by the column data.

Q. What is a rowid?

The rowid identifies each row piece by its location or address. Once assigned, a given row piece retains its rowid until the corresponding row is deleted, or exported and imported using the Export and Import utilities.

Q. What is a view? (KPIT Infotech, Pune)

A view is a tailored presentation of the data contained in one or more tables or other views. A view takes the output of a query and treats it as a table. Therefore, a view can be thought of as a stored query or a virtual table.

Unlike a table, a view is not allocated any storage space, nor does a view actually contain data. Rather, a view is defined by a query that extracts or derives data from the tables that the view references. These tables are called base tables. Base tables can in turn be actual tables or can be views themselves (including snapshots). Because a view is based on other objects, a view requires no storage other than storage for the definition of the view (the stored query) in the data dictionary.

Q. What are the advantages of having a view?

The advantages of having a view are:

- ❖ To provide an additional level of table security by restricting access to a predetermined set of rows or columns of a table
- ❖ To hide data complexity
- ❖ To simplify statements for the user
- ❖ To present the data in a different perspective from that of the base table
- ❖ To isolate applications from changes in definitions of base tables
- ❖ To save complex queries

For example, a query can perform extensive calculations with table information.

By saving this query as a view, you can perform the calculations each time the view is queried.

Q. What is a Materialized View? (Honeywell, KPIT Infotech, Pune)

Materialized views, also called snapshots, are schema objects that can be used to summarize, precompute, replicate, and distribute data. They are suitable in various computing environments especially for data warehousing.

From a physical design point of view, Materialized Views resembles tables or partitioned tables and behave like indexes.

Q. What is the significance of Materialized Views in data warehousing?

In data warehouses, materialized views are used to precompute and store aggregated data such as sums and averages. Materialized views in these environments are typically referred to as summaries because they store summarized data. They can also be used to precompute joins with or without aggregations.

Cost-based optimization can use materialized views to improve query performance by automatically recognizing when a materialized view can and should be used to satisfy a request. The optimizer transparently rewrites the request to use the materialized view. Queries are then directed to the materialized view and not to the underlying detail tables or views.

Q. Differentiate between Views and Materialized Views? (KPIT Infotech, Pune)

Q. What is the major difference between an index and Materialized view?

Unlike indexes, materialized views can be accessed directly using a SELECT statement.

Q. What are the procedures for refreshing Materialized views?

Oracle maintains the data in materialized views by refreshing them after changes are made to their master tables.

The refresh method can be:

- a) incremental (fast refresh) or
- b) complete

For materialized views that use the fast refresh method, a materialized view log or direct loader log keeps a record of changes to the master tables.

Materialized views can be refreshed either on demand or at regular time intervals.

Alternatively, materialized views in the same database as their master tables can be refreshed whenever a transaction commits its changes to the master tables.

Q. What are materialized view logs?

A materialized view log is a schema object that records changes to a master table's data so that a materialized view defined on the master table can be refreshed incrementally. Another name for materialized view log is snapshot log.

Each materialized view log is associated with a single master table. The materialized view log resides in the same database and schema as its master table.

Q. What is a synonym?

A synonym is an alias for any table, view, snapshot, sequence, procedure, function, or package. Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary.

Q. What are the advantages of having synonyms?

Synonyms are often used for security and convenience.

For example, they can do the following:

1. Mask the name and owner of an object
2. Provide location transparency for remote objects of a distributed database
3. Simplify SQL statements for database users

Q. What are the advantages of having an index? Or What is an index?

The purpose of an index is to provide pointers to the rows in a table that contain a given key value. In a regular index, this is achieved by storing a list of rowids for each key corresponding to the rows with that key value. Oracle stores each key value repeatedly with each stored rowid.

Q. What are the different types of indexes supported by Oracle?

The different types of indexes are:

- a. B-tree indexes
- b. B-tree cluster indexes
- c. Hash cluster indexes
- d. Reverse key indexes
- e. Bitmap indexes

Q. Can we have function based indexes?

Yes, we can create indexes on functions and expressions that involve one or more columns in the table being indexed. A function-based index precomputes the value of the function or expression and stores it in the index.

You can create a function-based index as either a B-tree or a bitmap index.

Q. What are the restrictions on function based indexes?

The function used for building the index can be an arithmetic expression or an expression that contains a PL/SQL function, package function, C callout, or SQL function. The expression cannot contain any aggregate functions, and it must be DETERMINISTIC. For building an index on a column containing an object type, the function can be a method of that object, such as a map method. However, you cannot build a function-based index on a LOB column, REF, or nested table column, nor can you build a function-based index if the object type contains a LOB, REF, or nested table.

Q. What are the advantages of having a B-tree index?

The major advantages of having a B-tree index are:

1. B-trees provide excellent retrieval performance for a wide range of queries, including exact match and range searches.
2. Inserts, updates, and deletes are efficient, maintaining key order for fast retrieval.
3. B-tree performance is good for both small and large tables, and does not degrade as the size of a table grows.

Q. What is a bitmap index? (KPIT Infotech, Pune)

The purpose of an index is to provide pointers to the rows in a table that contain a given key value. In a regular index, this is achieved by storing a list of rowids for each key corresponding to the rows with that key value. Oracle stores each key value repeatedly with each stored rowid. In a bitmap index, a bitmap for each key value is used instead of a list of rowids.

Each bit in the bitmap corresponds to a possible rowid. If the bit is set, then it means that the row with the corresponding rowid contains the key value. A mapping function converts the bit position to an actual rowid, so the bitmap index provides the same functionality as a regular index even though it uses a different representation internally. If the number of different key values is small, then bitmap indexes are very space efficient.

Bitmap indexing efficiently merges indexes that correspond to several conditions in a WHERE clause. Rows that satisfy some, but not all, conditions are filtered out before the table itself is accessed. This improves response time, often dramatically.

Q. What are the advantages of having bitmap index for data warehousing applications?

(KPIT Infotech, Pune)

Bitmap indexing benefits data warehousing applications which have large amounts of data and ad hoc queries but a low level of concurrent transactions. For such applications, bitmap indexing provides:

1. Reduced response time for large classes of ad hoc queries
2. A substantial reduction of space usage compared to other indexing techniques
3. Dramatic performance gains even on very low end hardware
4. Very efficient parallel DML and loads

Q. What is the advantage of bitmap index over B-tree index?

Fully indexing a large table with a traditional B-tree index can be prohibitively expensive in terms of space since the index can be several times larger than the data in the table. Bitmap indexes are typically only a fraction of the size of the indexed data in the table.

Q. What is the limitation/drawback of a bitmap index?

Bitmap indexes are not suitable for OLTP applications with large numbers of concurrent transactions modifying the data. These indexes are primarily intended for decision support in data warehousing applications where users typically query the data rather than update it.

Bitmap indexes are not suitable for high-cardinality data.

Q. How do you choose between B-tree index and bitmap index?

The advantages of using bitmap indexes are greatest for low cardinality columns: that is, columns in which the number of distinct values is small compared to the number of rows in the table. If the values in a column are repeated more than a hundred times, then the column is a candidate for a bitmap index. Even columns with a lower number of repetitions and thus higher cardinality, can be candidates if they tend to be involved in complex conditions in the WHERE clauses of queries.

For example, on a table with one million rows, a column with 10,000 distinct values is a candidate for a bitmap index. A bitmap index on this column can out-perform a B-tree index, particularly when this column is often queried in conjunction with other columns.

B-tree indexes are most effective for high-cardinality data: that is, data with many possible values, such as CUSTOMER_NAME or PHONE_NUMBER. A regular Btree index can be several times larger than the indexed data. Used appropriately, bitmap indexes can be significantly smaller than a corresponding B-tree index.

Q. What are clusters?

Clusters are an optional method of storing table data. A cluster is a group of tables that share the same data blocks because they share common columns and are often used together.

For example, the EMP and DEPT table share the DEPTNO column. When you cluster the EMP and DEPT tables, Oracle physically stores all rows for each department from both the EMP and DEPT tables in the same data blocks.

Q. What is partitioning? (KPIT Infotech, Pune)

Partitioning addresses the key problem of supporting very large tables and indexes by allowing you to decompose them into smaller and more manageable pieces called partitions. Once partitions are defined, SQL statements can access and manipulate the partitions rather than entire tables or indexes. Partitions are especially useful in data warehouse applications, which commonly store and analyze large amounts of historical data.

Q. What are the different partitioning methods?

Two primary methods of partitioning are available:

1. range partitioning, which partitions the data in a table or index according to a range of values, and
2. hash partitioning, which partitions the data according to a hash function.

Another method, composite partitioning, partitions the data by range and further subdivides the data into sub partitions using a hash function.

Q. What is the necessity to have table partitions?

The need to partition large tables is driven by:

- Data Warehouse and Business Intelligence demands for ad hoc analysis on great quantities of historical data
- Cheaper disk storage
- Application performance failure due to use of traditional techniques

Q. What are the advantages of storing each partition in a separate tablespace?

The major advantages are:

1. You can contain the impact of data corruption.
2. You can back up and recover each partition or subpartition independently.
3. You can map partitions or subpartitions to disk drives to balance the I/O load.

Q. What are the advantages of partitioning?

Partitioning is useful for:

1. Very Large Databases (VLDBs)
2. Reducing Downtime for Scheduled Maintenance
3. Reducing Downtime Due to Data Failures
4. DSS Performance
5. I/O Performance
6. Disk Striping: Performance versus Availability
7. Partition Transparency

Q. What is Range Partitioning? (KPIT Infotech, Pune)

Range partitioning maps rows to partitions based on ranges of column values. Range partitioning is defined by the partitioning specification for a table or index:

PARTITION BY RANGE (column_list)

and by the partitioning specifications for each individual partition:

VALUES LESS THAN (value_list)

Q. What is Hash Partitioning?

Hash partitioning uses a hash function on the partitioning columns to stripe data into partitions. Hash partitioning allows data that does not lend itself to range partitioning to be easily partitioned for performance reasons such as parallel DML, partition pruning, and partition-wise joins.

Q. What are the advantages of Hash partitioning over Range Partitioning?

Hash partitioning is a better choice than range partitioning when:

- a) You do not know beforehand how much data will map into a given range
- b) Sizes of range partitions would differ quite substantially
- c) Partition pruning and partition-wise joins on a partitioning key are important

Q. What are the rules for partitioning a table?

A table can be partitioned if:

- It is not part of a cluster
- It does not contain LONG or LONG RAW datatypes

Q. What is a global partitioned index?

In a global partitioned index, the keys in a particular index partition may refer to rows stored in more than one underlying table partition or subpartition. A global index can only be range-partitioned, but it can be defined on any type of partitioned table.

Q. What is a local index?

In a local index, all keys in a particular index partition refer only to rows stored in a single underlying table partition. A local index is created by specifying the LOCAL attribute.

Q. What are CLOB and NCLOB datatypes? (Mascot)

The CLOB and NCLOB datatypes store up to four gigabytes of character data in the database. CLOBs store single-byte character set data and NCLOBs store fixed-width and varying-width multibyte national character set data (NCHAR data).

Q. What is PL/SQL?

PL/SQL is Oracle's procedural language extension to SQL. PL/SQL enables you to mix SQL statements with procedural constructs. With PL/SQL, you can define and execute PL/SQL program units such as procedures, functions, and packages.

PL/SQL program units generally are categorized as anonymous blocks and stored procedures.

Q. What is an anonymous block?

An anonymous block is a PL/SQL block that appears within your application and it is not named or stored in the database.

Q. What is a Stored Procedure?

A stored procedure is a PL/SQL block that Oracle stores in the database and can be called by name from an application. When you create a stored procedure, Oracle parses the procedure and stores its parsed representation in the database.

Q. What is a distributed transaction?

A distributed transaction is a transaction that includes one or more statements that update data on two or more distinct nodes of a distributed database.

Q. What are packages? (KPIT Infotech, Pune)

A package is a group of related procedures and functions, together with the cursors and variables they use, stored together in the database for continued use as a unit.

While packages allow the administrator or application developer the ability to organize such routines, they also offer increased functionality (for example, global package variables can be declared and used by any procedure in the package) and performance (for example, all objects of the package are parsed, compiled, and loaded into memory once).

Q. What are procedures and functions? (KPIT Infotech, Pune)

A procedure or function is a schema object that consists of a set of SQL statements and other PL/SQL constructs, grouped together, stored in the database, and executed as a unit to solve a specific problem or perform a set of related tasks. Procedures and functions permit the caller to provide parameters that can be input only, output only, or input and output values.

Q. What is the difference between Procedure and Function?

Procedures and functions are identical except that functions always return a single value to the caller, while procedures do not return values to the caller.

Q. What is a DML and what do they do?

Data manipulation language (DML) statements query or manipulate data in existing schema objects. They enable you to:

1. Retrieve data from one or more tables or views (SELECT)
2. Add new rows of data into a table or view (INSERT)
3. Change column values in existing rows of a table or view (UPDATE)
4. Remove rows from tables or views (DELETE)
5. See the execution plan for a SQL statement (EXPLAIN PLAN)
6. Lock a table or view, temporarily limiting other users' access (LOCK TABLE)

Q. What is a DDL and what do they do?

Data definition language (DDL) statements define, alter the structure of, and drop schema objects. DDL statements enable you to:

1. Create, alter, and drop schema objects and other database structures, including the database itself and database users (CREATE, ALTER, DROP)
2. Change the names of schema objects (RENAME)
3. Delete all the data in schema objects without removing the objects' structure (TRUNCATE)
4. Gather statistics about schema objects, validate object structure, and list chained rows within objects (ANALYZE)
5. Grant and revoke privileges and roles (GRANT, REVOKE)
6. Turn auditing options on and off (AUDIT, NOAUDIT)
7. Add a comment to the data dictionary (COMMENT)

Q. What are shared sql's?

Oracle automatically notices when applications send identical SQL statements to the database. The SQL area used to process the first occurrence of the statement is shared—that is, used for processing subsequent occurrences of that same statement. Therefore, only one shared SQL area exists for a unique statement. Since shared SQL areas are shared memory areas, any Oracle process can use a shared SQL area. The sharing of SQL areas reduces memory usage on the database server, thereby increasing system throughput.

Q. What are triggers?

Oracle allows to define procedures called triggers that execute implicitly when an INSERT, UPDATE, or DELETE statement is issued against the associated table or, in some cases, against a view, or when database system actions occur. These procedures can be written in PL/SQL or Java and stored in the database, or they can be written as C callouts.

Q. What is Cost-based Optimization?

Using the cost-based approach, the optimizer determines which execution plan is most efficient by considering available access paths and factoring in information based on statistics for the schema objects (tables or indexes) accessed by the SQL statement.

Q. What is Rule-Based Optimization?

Using the rule-based approach, the optimizer chooses an execution plan based on the access paths available and the ranks of these access paths.

Q. What is meant by degree of parallelism?

The number of parallel execution servers associated with a single operation is known as the degree of parallelism.

Q. What is meant by data consistency?

Data consistency means that each user sees a consistent view of the data, including visible changes made by the user's own transactions and transactions of other users.

Q. What are Locks?

Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource—either user objects such as tables and rows or system objects not visible to users, such as shared data structures in memory and data dictionary rows.

Q. What are the locking modes used in Oracle?

Oracle uses two modes of locking in a multiuser database:

Exclusive lock mode: Prevents the associated resource from being shared. This lock mode is obtained to modify data. The first transaction to lock a resource exclusively is the only transaction that can alter the resource until the exclusive lock is released.

Share lock mode: Allows the associated resource to be shared, depending on the operations involved. Multiple users reading data can share the data, holding share locks to prevent concurrent access by a writer (who needs an exclusive lock). Several transactions can acquire share locks on the same resource.

Q. What is a deadlock?

A deadlock can occur when two or more users are waiting for data locked by each other.

Q. How can you avoid deadlocks?

Multitable deadlocks can usually be avoided if transactions accessing the same tables lock those tables in the same order, either through implicit or explicit locks.

For example, all application developers might follow the rule that when both a master and detail table are updated, the master table is locked first and then the detail table. If such rules are properly designed and then followed in all applications, deadlocks are very unlikely to occur.

Q. What is redo log?

The redo log, present for every Oracle database, records all changes made in an Oracle database. The redo log of a database consists of at least two redo log files that are separate from the datafiles (which actually store a database's data). As part of database recovery from an instance or media failure, Oracle applies the appropriate changes in the database's redo log to the datafiles, which updates database data to the instant that the failure occurred.

A database's redo log can consist of two parts: the online redo log and the archived redo log.

Q. What are Rollback Segments?

Rollback segments are used for a number of functions in the operation of an Oracle database. In general, the rollback segments of a database store the old values of data changed by ongoing transactions for uncommitted transactions.

Among other things, the information in a rollback segment is used during database recovery to undo any uncommitted changes applied from the redo log to the datafiles. Therefore, if database recovery is necessary, then the data is in a consistent state after the rollback segments are used to remove all uncommitted data from the datafiles.

Q. What is SGA?

The System Global Area (SGA) is a shared memory region that contains data and control information for one Oracle instance. An SGA and the Oracle background processes constitute an Oracle instance.

Oracle allocates the system global area when an instance starts and deallocates it when the instance shuts down. Each instance has its own system global area.

Users currently connected to an Oracle server share the data in the system global area. For optimal performance, the entire system global area should be as large as possible (while still fitting in real memory) to store as much data in memory as possible and minimize disk I/O.

The information stored within the system global area is divided into several types of memory structures, including the database buffers, redo log buffer, and the shared pool. These areas have fixed sizes and are created during instance startup.

Q. What is PCTFREE?

The PCTFREE parameter sets the minimum percentage of a data block to be reserved as free space for possible updates to rows that already exist in that block.

Q. What is PCTUSED?

The PCTUSED parameter sets the minimum percentage of a block that can be used for row data plus overhead before new rows will be added to the block. After a data block is filled to the limit determined by PCTFREE, Oracle considers the block unavailable for the insertion of new rows until the percentage of that block falls below the parameter PCTUSED. Until this value is achieved, Oracle uses the free space of the data block only for updates to rows already contained in the data block.

Notes:

Nulls are stored in the database if they fall between columns with data values. In these cases they require one byte to store the length of the column (zero).

Trailing nulls in a row require no storage because a new row header signals that the remaining columns in the previous row are null. For example, if the last three columns of a table are null, no information is stored for those columns. In tables with many columns, the columns more likely to contain nulls should be defined last to conserve disk space.

Two rows can both contain all nulls without violating a unique index.

NULL values in indexes are considered to be distinct except when all the non-NULL values in two or more rows of an index are identical, in which case the rows are considered to be identical. Therefore, UNIQUE indexes prevent rows containing NULL values from being treated as identical.

Bitmap indexes include rows that have NULL values, unlike most other types of indexes. Indexing of nulls can be useful for some types of SQL statements, such as queries with the aggregate function COUNT.

Bitmap indexes on partitioned tables must be local indexes.

PL/SQL is Oracle's procedural language extension to SQL. *PL/SQL* combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF ... THEN, WHILE, and LOOP.

When designing a database application, a developer should consider the advantages of using stored *PL/SQL*:

- Because *PL/SQL* code can be stored centrally in a database, network traffic between applications and the database is reduced, so application and system performance increases.

- Data access can be controlled by stored *PL/SQL* code. In this case, the users of *PL/SQL* can access data only as intended by the application developer (unless

another access route is granted).

• PL/SQL blocks can be sent by an application to a database, executing complex operations without excessive network traffic.

Even when PL/SQL is not stored in the database, applications can send blocks of PL/SQL to the database rather than individual SQL statements, thereby again reducing network traffic.

The following sections describe the different program units that can be defined and stored centrally in a database.

Committing and Rolling Back Transactions

The changes made by the SQL statements that constitute a transaction can be either committed or rolled back. After a transaction is committed or rolled back, the next transaction begins with the next SQL statement.

Committing a transaction makes permanent the changes resulting from all SQL statements in the transaction. The changes made by the SQL statements of a transaction become visible to other user sessions' transactions that start only after the transaction is committed.

Rolling back a transaction retracts any of the changes resulting from the SQL statements in the transaction. After a transaction is rolled back, the affected data is left unchanged as if the SQL statements in the transaction were never executed.

Introduction to the Data Dictionary

One of the most important parts of an Oracle database is its *data dictionary*, which is a **read-only** set of tables that provides information about its associated database. A data dictionary contains:

• The definitions of all schema objects in the database (tables, views, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, and so on)

• How much space has been allocated for, and is currently used by, the schema objects

• Default values for columns

• Integrity constraint information

• The names of Oracle users

• Privileges and roles each user has been granted

• Auditing information, such as who has accessed or updated various schema objects

• Other general database information

The data dictionary is structured in tables and views, just like other database data.

All the data dictionary tables and views for a given database are stored in that database's SYSTEM tablespace.

Not only is the data dictionary central to every Oracle database, it is an important tool for all users, from end users to application designers and database administrators. To access the data dictionary, you use SQL statements. Because the data dictionary is read-only, you can issue only queries (SELECT statements) against the tables and views of the data dictionary.

Q. What is the function of DUMMY table?

The table named DUAL is a small table in the data dictionary that Oracle and user written programs can reference to guarantee a known result. This table has one column called DUMMY and one row containing the value "X".

Databases, tablespaces, and datafiles are closely related, but they have important differences:

Databases and tablespaces: An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all of the database's data.

Tablespaces and datafiles: Each table in an Oracle database consists of one or more files called datafiles, which are physical structures that conform with the operating system in which Oracle is running.

databases and datafiles:

A database's data is collectively stored in the datafiles that constitute each tablespace of the database. For example, the simplest Oracle database would have one tablespace and one datafile. Another database might have three tablespaces, each consisting of two datafiles (for a total of six datafiles).

Nulls

A *null* is the absence of a value in a column of a row. Nulls indicate missing, unknown, or inapplicable data. A null should not be used to imply any other value, such as zero. A column allows nulls unless a NOT NULL or PRIMARY KEY integrity constraint has been defined for the column, in which case no row can be inserted without a value for that column.

Nulls are stored in the database if they fall between columns with data values. In these cases they require one byte to store the length of the column (zero).

Trailing nulls in a row require no storage because a new row header signals that the remaining columns in the previous row are null. For example, if the last three columns of a table are null, no information is stored for those columns. In tables with many columns, the columns more likely to contain nulls should be defined last to conserve disk space.

Most comparisons between nulls and other values are by definition neither true nor false, but unknown. To identify nulls in SQL, use the IS NULL predicate. Use the SQL function NVL to convert nulls to non-null values.

Nulls are not indexed, except when the cluster key column value is null or the index is a bitmap index.

What are different types of locks?

Q. Master table and Child table performances and comparisons in Oracle?

Q. What are the different types of Cursors? Explain. (Honeywell)

Q. What are the different types of Deletes?

Q. Can a View be updated?

Interview Questions from Honeywell

1. What is pragma?
2. Can you write commit in triggers?
3. Can you call user defined functions in select statements
4. Can you call insert/update/delete in select statements. If yes how? If no what is the other way?
5. After update how do you know, how many records got updated
6. Select statement does not retrieve any records. What exception is raised?

Interview Questions from Shreesoft

1. How many columns can a PLSQL table have

Interview Questions from mascot

1. What is Load balancing & what u have used to do this? (SQL Loader)
2. What r Routers?

PL/SQL

1. What are different types of joins?
2. Difference between Packages and Procedures
3. Difference between Function and Procedures
4. How many types of triggers are there? When do you use Triggers
5. Can you write DDL statements in Triggers? (No)
6. What is Hint?
7. How do you tune a SQL query?

Interview Questions from KPIT Infotech, Pune

1. Package body
2. What is molar query?
3. What is row level security

General:

Why ORACLE is the best database for Datawarehousing

For data loading in Oracle, what are conventional loading and direct-path loading ?

7. If you use oracle SQL*Loader, how do you transform data with it during loading ?
Example.

Three ways SQL*Loader could load data, what are those three types ?

What are the contents of "bad files" and "discard files" when using SQL*Loader ?

How do you use commit frequencies ? how does it affect loading performance ?

What are the other factors of the database on which the loading performance depend ?

* WHAT IS PARALLELISM ?

* WHAT IS A PARALLEL QUERY ?

* WHAT ARE DIFFERENT WAYS OF LOADING DATA TO DATAWAREHOUSE USING ORACLE?

* WHAT IS TABLE PARTITIONING? HOW IT IS USEFUL TO WAREHOUSE DATABASE?

* WHAT ARE DIFFERENT TYPES OF PARTITIONING IN ORACLE?

* WHAT IS A MATERIALIZED VIEW? HOW IT IS DIFFERENT FROM NORMAL AND INLINE VIEWS?

* WHAT IS INDEXING? WHAT ARE DIFFERENT TYPES OF INDEXES SUPPORTED BY ORACLE?

- * WHAT ARE DIFFERENT STORAGE OPTIONS SUPPORTED BY ORACLE?
- * WHAT IS QUERY OPTIMIZER? WHAT ARE DIFFERENT TYPES OF OPTIMIZERS SUPPORTED BY ORACLE?
- * EXPLAIN ROLLUP,CUBE,RANK AND DENSE_RANK FUNCTIONS OF ORACLE 8i.

The advantages of using bitmap indexes are greatest for low cardinality columns: that is, columns in which the number of distinct values is small compared to the number of rows in the table. A gender column, which only has two distinct values (male and female), is ideal for a bitmap index. However, data warehouse administrators will also choose to build bitmap indexes on columns with much higher cardinalities.

Local vs global: A B-tree index on a partitioned table can be local or global. Global indexes must be fully rebuilt after a direct load, which can be very costly when loading a relatively small number of rows into a large table. For this reason, it is strongly recommended that indexes on partitioned tables should be defined as local indexes unless there is a well-justified performance requirement for a global index. Bitmap indexes on partitioned tables are always local.

Why Constraints are Useful in a Data Warehouse

Constraints provide a mechanism for ensuring that data conforms to guidelines specified by the database administrator. The most common types of constraints include unique constraints (ensuring that a given column is unique), not-null constraints, and foreign-key constraints (which ensure that two keys share a primary key-foreign key relationship).

Materialized Views for Data Warehouses

In data warehouses, materialized views can be used to precompute and store aggregated data such as the sum of sales. Materialized views in these environments are typically referred to as summaries, because they store summarized data. They can also be used to precompute joins with or without aggregations. A materialized view eliminates the overhead associated with expensive joins or aggregations for a large or important class of queries.

The Need for Materialized Views

Materialized views are used in data warehouses to increase the speed of queries on very large databases. Queries to large databases often involve joins between tables or aggregations such as SUM, or both. These operations are very expensive in terms of time and processing power.

How does MV's work?

The query optimizer can use materialized views by automatically recognizing when an existing materialized view can and should be used to satisfy a request. It then transparently rewrites the request to use the materialized view. Queries are then directed to the materialized view and not to the underlying detail tables. In general, rewriting queries to use materialized views rather than detail tables results in a significant performance gain.

If a materialized view is to be used by query rewrite, it must be stored in the same database as its fact or detail tables. A materialized view can be partitioned, and you can define a materialized view on a partitioned table and one or more indexes on the materialized view.

The types of materialized views are:

- Materialized Views with Joins and Aggregates
- Single-Table Aggregate Materialized Views
- Materialized Views Containing Only Joins

Some Useful system tables:

user_tab_partitions
user_tab_columns