# Few Oracle DB Keywords to know.

## 1. Buffer

- **Description**: A buffer in Oracle DB refers to a memory area in the buffer cache that stores data blocks read from disk. When a query requests data, Oracle first checks the buffer cache before fetching data from disk.
- **Example**: If a table scan is performed, the data blocks are read into the buffer cache for subsequent operations.

## 2. Cache

- **Description**: Cache refers to memory structures used by Oracle to store frequently accessed data and objects to reduce disk I/O. The most common types are the **buffer cache**, **library cache**, and **result cache**.
- **Example**: The **buffer cache** stores copies of data blocks from the database, while the **library cache** holds parsed SQL statements and execution plans.

## 3. Parsing

- **Description**: Parsing is the process Oracle uses to translate a SQL statement into an execution plan. It includes syntax checks, semantic checks, and the generation of an execution plan.
- **Example**: When you execute SELECT * FROM employees, Oracle parses this statement to check its syntax and generate a plan for execution.

## 4. Hard Parses

- **Description**: A hard parse occurs when Oracle needs to parse a SQL statement for the first time or cannot reuse an existing execution plan. This involves significant CPU work as Oracle must analyze and optimize the query.
- **Example**: A hard parse happens when you run a SQL query with different literal values that Oracle treats as unique statements, such as SELECT * FROM employees WHERE department_id = 10 and SELECT * FROM employees WHERE department_id = 20.

## 5. Soft Parses

- **Description**: A soft parse occurs when Oracle can reuse an existing execution plan from the library cache without fully parsing the SQL statement. This is faster and consumes fewer resources than a hard parse.
- **Example**: If a previously executed query is run again using bind variables, Oracle performs a soft parse and reuses the cached plan.

### 6. Latching

- **Description**: Latching is a mechanism Oracle uses to control access to shared memory structures and prevent data corruption. Latches are lightweight locks that ensure that only one process modifies a particular memory structure at a time.
- **Example**: Latches are used in the shared pool to manage concurrent access to parsed SQL and PL/SQL code.

### 7. Latch Contention

- **Description**: Latch contention occurs when multiple processes compete for a latch. This results in waits, where processes must wait to obtain the latch before accessing a shared resource.
- **Example**: High latch contention is often seen in the shared pool during peak times when many SQL statements are parsed simultaneously, leading to library cache latch waits.

### 8. Waits

- **Description**: Waits refer to the time a session spends waiting for a resource to become available. Oracle tracks wait times to help diagnose performance bottlenecks.
- **Example**: If a query waits for data to be read from disk, it may show a db file sequential read wait event.

### 9. Wait Events

- **Description**: Wait events are specific conditions that a session encounters while waiting for a resource. These events help identify what the database is waiting on, such as I/O operations, latches, or locks.
- **Example**: Common wait events include db file scattered read, log file sync, and library cache lock.

### 10. Locks

- **Description**: Locks are mechanisms to manage concurrent access to database resources, ensuring data consistency and preventing conflicts. Locks can be row-level, table-level, or system-wide.
- **Example**: When a user updates a row in a table, Oracle places a row lock to prevent other sessions from modifying the same row until the transaction is committed or rolled back.

## 11. Logical Reads

- **Description**: Logical reads are operations where Oracle retrieves data from the buffer cache without accessing disk. It refers to reading data that is already in memory.
- **Example**: A query that reads a data block from the buffer cache incurs a logical read, which is faster than a physical read.

## 12. Deadlocks

- **Description**: A deadlock occurs when two or more sessions block each other by holding resources that the other session needs, creating a cycle where no session can proceed.
- **Example**: Session 1 locks `Table A` and waits for `Table B` locked by Session 2, while Session 2 holds `Table B` and waits for `Table A`. This creates a deadlock that Oracle resolves by aborting one of the sessions.

## 13. PGA (Program Global Area)

- **Description**: The PGA is a memory region that contains data and control information for a server process. It is used for operations such as sorting, hash joins, and other session-specific tasks.
- **Example**: When a query requires sorting, the PGA allocates memory for that sort operation. If insufficient, it spills to disk, causing slower performance.

## 14. SGA (System Global Area)

- **Description**: The SGA is a shared memory area that holds data and control information for an Oracle instance. It includes the buffer cache, shared pool, and other memory structures.
- **Example**: The **shared pool** within the SGA contains parsed SQL statements, while the **buffer cache** holds frequently accessed data blocks.

## 15. Redo Log

- **Description**: The redo log consists of files that record changes made to the database, ensuring data can be recovered in case of a failure. Each transaction writes redo entries to the redo log buffer, which is then flushed to disk.
- **Example**: A `COMMIT` operation triggers Oracle to write redo entries from the redo log buffer to the redo log files, ensuring the changes are saved and can be replayed during recovery.

### 16. Library Cache

- **Description**: The library cache is a part of the shared pool within the SGA that stores parsed SQL statements, PL/SQL code, and execution plans. This allows Oracle to reuse execution plans and reduce the need for hard parsing.
- **Example**: When a SQL statement is executed, Oracle checks the library cache to see if a parsed version already exists. If found, it performs a soft parse; if not, it hard parses the statement and stores it in the cache.

### 17. Shared Pool

- **Description**: The shared pool is a component of the SGA that holds various memory structures, including the library cache and the data dictionary cache. It plays a crucial role in SQL parsing and execution.
- **Example**: The shared pool helps manage the parsed SQL statements and metadata. If it's too small, Oracle may encounter library cache contention, leading to performance issues.

### 18. Data Dictionary Cache

- **Description**: The data dictionary cache, also known as the row cache, is part of the shared pool that stores metadata information about database objects (e.g., tables, columns, users). This helps reduce the need to repeatedly access the physical data dictionary.
- **Example**: When a query references a table, Oracle checks the data dictionary cache for metadata to validate the query. If the cache doesn't contain the required information, it fetches it from disk, which can slow performance.

### 19. Control File

- **Description**: Control files are crucial database files that store information about the database structure, including datafile locations, redo log locations, and other metadata essential for database operation.
- **Example**: During instance startup, Oracle reads the control file to identify the database structure and ensure it is in sync. If the control file is damaged or missing, the database will not start.

### 20. Checkpoint

- **Description**: A checkpoint is an event that writes all modified database buffers in the buffer cache to datafiles, ensuring that data is consistent up to that point. It helps in reducing recovery time after a failure.

- **Example**: Oracle performs checkpoints during certain intervals or events, such as when the redo log switches. This flushes dirty buffers to disk, making recovery faster and reducing the number of redo log entries needed for recovery.

## 21. Latch Free Wait Event

- **Description**: The latch free wait event indicates that a process is waiting for a latch that is currently held by another process. This can be due to contention in accessing shared resources.
- **Example**: High latch free wait events often occur when there are many concurrent queries trying to access the same memory structure, such as the shared pool or redo log buffer.

## 22. Row Lock (TX) Waits

- **Description**: Row lock waits occur when a session tries to modify a row that is locked by another session. This prevents data corruption and ensures data consistency.
- **Example**: If Session 1 updates a row in the orders table but hasn't committed the change, Session 2 trying to update the same row will experience a TX - row lock contention wait event until Session 1 commits or rolls back the transaction.

## 23. Deadlock Detection

- **Description**: Oracle automatically detects deadlocks when two or more sessions block each other in a cyclic wait condition. The database resolves the deadlock by aborting one of the sessions and rolling back its transaction.
- **Example**: When a deadlock occurs, Oracle writes details to the alert log and produces a trace file with information about the sessions and SQL involved. This helps diagnose and prevent future deadlocks.

## 24. Redo Log Buffer

- **Description**: The redo log buffer is a part of the SGA where redo entries are temporarily stored before being written to the redo log files on disk. It ensures data changes are logged and recoverable.
- **Example**: If the redo log buffer is undersized or heavily used, you may see log buffer space wait events, indicating that processes are waiting for the buffer to flush before they can write more redo entries.

### 25. Log File Sync Wait Event

- **Description**: The log file sync wait event occurs when a session waits for a commit to complete. This wait event measures the time it takes for redo entries to be written from the redo log buffer to the redo log files.
- **Example**: If you frequently see high log file sync waits, it indicates slow redo log writes. This can be mitigated by placing redo logs on faster storage or optimizing the redo log buffer size.

### 26. Log Writer (LGWR) Process

- **Description**: The LGWR process is responsible for writing redo log entries from the redo log buffer to the redo log files on disk. It is crucial for ensuring that changes are safely recorded and available for recovery.
- **Example**: The LGWR process writes to disk when a transaction commits, when the redo log buffer reaches a certain threshold, or at specific intervals set by the database.

### 27. Consistent Gets

- **Description**: Consistent gets are logical reads of data blocks in the buffer cache that ensure a consistent view of data for a query, even while other sessions might be modifying the data.
- **Example**: When a SELECT statement is executed, Oracle retrieves the relevant data blocks from the buffer cache, performing consistent gets to provide a consistent snapshot.

### 28. Direct Path Reads

- **Description**: Direct path reads bypass the buffer cache and read data directly from disk to the PGA. This method is often used for large, full table scans or direct read operations like CREATE TABLE AS SELECT.
- **Example**: When a large SELECT query reads data too big to benefit from caching, Oracle may choose direct path reads for efficiency, reflected as direct path read wait events.

### 29. Direct Path Writes

- **Description**: Direct path writes are used to write large volumes of data directly from the PGA to disk, bypassing the buffer cache. This is common in bulk operations, such as INSERT APPEND.
- **Example**: INSERT /*+ APPEND */ INTO large_table SELECT * FROM another_table will write data directly to disk, avoiding cache overhead and speeding up large data loads.

## 30. Sorts in PGA vs. Temporary Tablespace

- **Description**: Sort operations can be performed in memory (PGA) or written to the temporary tablespace if the PGA runs out of space. Sorting in the PGA is faster, while disk-based sorts are slower and more I/O intensive.
- **Example**: When running a ORDER BY or GROUP BY, if the operation requires more memory than allocated in the PGA, it will spill to the temporary tablespace, impacting performance.

## 31. Automatic Workload Repository (AWR)

- **Description**: AWR is a built-in Oracle repository that collects, processes, and maintains performance statistics for the database. It captures data such as wait events, session activity, and SQL performance metrics at regular intervals.
- **Example**: AWR reports can be generated to identify performance bottlenecks and understand resource utilization patterns using DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT and querying DBA_HIST_* views.

## 32. Automatic Database Diagnostic Monitor (ADDM)

- **Description**: ADDM analyzes AWR data to automatically identify and diagnose performance issues in the database. It provides recommendations to improve performance based on the data captured in AWR snapshots.
- **Example**: After running a workload, you can view ADDM findings through Oracle Enterprise Manager or by querying the DBMS_ADVISOR package to get insights into performance problems and suggested resolutions.

## 33. Flashback Technology

- **Description**: Flashback technology allows Oracle databases to rewind data to a previous state without requiring a full database restore. It helps recover from logical errors such as accidental data deletions.
- **Example**: If a table is accidentally truncated, the FLASHBACK TABLE command can be used to restore it to a point in time before the truncation, as long as the UNDO_RETENTION is sufficient.

## 34. Data Guard

- **Description**: Data Guard is an Oracle feature for disaster recovery and high availability. It maintains standby databases as replicas of the primary database and can switch roles in case of a failure.

- **Example**: A physical standby database continuously applies redo logs from the primary database to stay synchronized, providing a backup that can be activated during a failover.

## 35. Oracle RAC (Real Application Clusters)

- **Description**: RAC allows multiple servers (nodes) to run a single database instance, enabling clustering for high availability and load balancing. It helps distribute workload across nodes and improves database resilience.
- **Example**: In a RAC environment, client connections can access any of the nodes in the cluster. If one node fails, the remaining nodes continue to serve requests, minimizing downtime.

## 36. ASM (Automatic Storage Management)

- **Description**: ASM is Oracle's file system and volume manager that simplifies storage management by automatically distributing database files across all available disk devices to optimize performance and reliability.
- **Example**: ASM can rebalance data across disks when a new disk is added or removed, without impacting database performance, which is useful for scaling storage dynamically.

## 37. Block Corruption

- **Description**: Block corruption occurs when a database block becomes unreadable or inconsistent due to disk errors, software bugs, or hardware failures. It can lead to data retrieval issues or even database crashes.
- **Example**: The DB_BLOCK_CHECKING parameter can be set to validate blocks during reads and writes, and the DBMS_REPAIR package can be used to detect and fix block corruption.

## 38. Row-Level Locking

- **Description**: Oracle uses row-level locking to allow multiple users to modify different rows of the same table simultaneously, improving concurrency and performance. It locks only the rows being modified, unlike table-level locking.
- **Example**: When UPDATE or DELETE commands are executed, only the affected rows are locked, allowing other rows to be accessed or modified by different transactions.

### 39. Snapshot Too Old (ORA-01555)

- **Description**: The ORA-01555 error, known as the “snapshot too old” error, occurs when a long-running query tries to access an older version of data that has been overwritten in the undo tablespace due to insufficient UNDO_RETENTION.
- **Example**: To prevent this, ensure the UNDO_RETENTION parameter is set appropriately for long-running transactions and monitor the size of the undo tablespace using V$UNDOSTAT.

### 40. Materialized Views

- **Description**: Materialized views are database objects that store the results of a query physically and can be refreshed periodically to keep them up-to-date. They are used to improve query performance by avoiding repeated complex joins or aggregations.
- **Example**: A materialized view created for a SELECT query with REFRESH FAST enables it to be updated incrementally whenever the base tables are updated, reducing the load on the system during refreshes.

### 41. Temporary Tablespace

- **Description**: The temporary tablespace is used for operations like sorting, hash joins, and large ORDER BY clauses that exceed the memory allocated in the PGA. It stores intermediate results temporarily.
- **Example**: When a query with ORDER BY needs to sort more data than the PGA can handle, it spills to the temporary tablespace. Monitoring V$TEMPSEG_USAGE can help identify large temporary space usage.

### 42. Consistency and Read Consistency

- **Description**: Oracle ensures read consistency by providing a snapshot of the data at the start of a query. It uses undo data to reconstruct changes made by other transactions that haven’t been committed yet.
- **Example**: If a long-running SELECT statement is reading rows while other transactions update those rows, Oracle ensures that the query sees the data as it was when the query began, using undo records to maintain consistency.

### 43. Redo Apply and Media Recovery

- **Description**: Redo apply is a mechanism that applies redo log records to a database to ensure changes are reflected in datafiles. Media recovery uses redo logs to recover from data loss or corruption.

- **Example**: If a datafile is lost or damaged, media recovery can restore it using the last backup and apply redo logs to bring the database up to the last committed transaction.

### 44. System Change Number (SCN)

- **Description**: The SCN is a unique number that Oracle assigns to each committed transaction. It serves as a timestamp and helps maintain data consistency and recovery points.
- **Example**: The SCN is used during recovery and flashback operations to identify the exact point in time to which the database needs to be restored.

### 45. Undo Data and Rollback

- **Description**: Undo data stores the before-image of data changes made by transactions, allowing changes to be rolled back and supporting read consistency. The undo tablespace is where this data is stored.
- **Example**: When a ROLLBACK command is issued, Oracle uses undo data to revert the affected rows to their previous state. The V$UNDOSTAT view provides statistics on undo space usage and retention.

### 46. Index B-Tree Structure

- **Description**: Oracle uses B-tree structures for indexes to ensure efficient searching and retrieval. Each index entry points to a row in a table, allowing for faster data access compared to a full table scan.
- **Example**: When a query uses a WHERE clause with an indexed column, Oracle can traverse the B-tree to find the relevant rows without scanning the entire table.

### 47. Histograms

- **Description**: Histograms provide detailed information about the distribution of values in a column. They help the optimizer make better decisions about which execution plan to use.
- **Example**: If a column has skewed data, creating a histogram allows the optimizer to understand the distribution and choose between full table scans and index lookups for better performance.

### 48. Bind Variables

- **Description**: Bind variables are placeholders in SQL statements that are replaced with actual values at runtime. They help reduce hard parsing and improve SQL plan reuse.

- **Example**: Instead of writing SELECT * FROM employees WHERE department_id = 10 and SELECT * FROM employees WHERE department_id = 20, using SELECT * FROM employees WHERE department_id = :dept_id ensures the same execution plan is reused.

## 49. Data Pump

- **Description**: Oracle Data Pump is a utility for fast data and metadata export/import. It allows for high-speed transfer of data between databases and supports parallel execution to improve performance.
- **Example**: To export a schema, you can use expdp with options like DIRECTORY, DUMPFILE, and SCHEMAS to specify the schema and output location. This is useful for data migration or backup purposes.

## 50. SQL Plan Baseline

- **Description**: SQL Plan Baselines are a feature that allows the database to maintain and use known, good execution plans. This ensures stability in query performance by preventing the optimizer from choosing suboptimal plans.
- **Example**: If a query has an execution plan that is efficient, Oracle can capture and store that plan as a baseline. This plan will be reused in future executions, even if new statistics or changes might otherwise lead the optimizer to choose a different plan.

## 51. Adaptive Execution Plans

- **Description**: Adaptive execution plans are dynamic plans that can change during the execution of a query based on the actual statistics collected at runtime. This helps Oracle adjust the execution strategy to optimize performance.
- **Example**: If a nested loop join initially appears optimal but actual runtime data shows a hash join would be better, Oracle can switch to the hash join during the execution phase.

## 52. Flash Recovery Area (FRA)

- **Description**: The Flash Recovery Area is a location on disk where Oracle stores recovery-related files such as backups, archived redo logs, and flashback logs. It simplifies recovery operations and automates space management for recovery files.
- **Example**: Configuring the FRA using parameters like DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE ensures that backup files and redo logs are managed in one location for quick access during recovery.

### 53. Row-Level Security (RLS)

- **Description**: RLS is an Oracle feature that provides fine-grained access control to data by applying policies that restrict data access at the row level. It allows different users to see different subsets of data based on security policies.
- **Example**: Using Oracle's DBMS_RLS package, you can create a policy that restricts access to a sales table so that each salesperson can only see data relevant to their own sales region.

### 54. Read Consistency Mechanism

- **Description**: Oracle's read consistency mechanism ensures that a query always returns a consistent view of the data as it was at the beginning of the query, even if changes are made to the data during query execution.
- **Example**: If a SELECT query starts and a UPDATE occurs on the same data partway through, Oracle uses undo data to provide the original state of the data as of the start of the query.

### 55. Direct Path Load

- **Description**: Direct path load bypasses the buffer cache and loads data directly into datafiles. This is used for faster data loads, as it reduces the overhead of writing to the buffer cache.
- **Example**: The SQL*Loader tool can perform direct path loading using the DIRECT=TRUE option, which is useful for bulk data loading operations.

### 56. ASM Disk Group

- **Description**: An ASM Disk Group is a collection of disks managed by Oracle ASM (Automatic Storage Management). ASM automatically stripes data across disks in the group and manages redundancy to improve performance and reliability.
- **Example**: You can create a disk group with the CREATE DISKGROUP command and include disks, specifying redundancy levels like NORMAL or HIGH to protect against disk failures.

### 57. ARCH Process

- **Description**: The ARCH process (Archiver Process) is responsible for copying redo log files to the archive location when the redo log is full or a log switch occurs. This ensures that redo logs are preserved for recovery purposes.
- **Example**: In databases operating in ARCHIVELOG mode, the ARCH process is essential for creating archived redo logs, which are used for point-in-time recovery and Data Guard synchronization.

## 58. Automatic Segment Space Management (ASSM)

- **Description**: ASSM automates the management of space within a segment, such as tables and indexes, by dynamically tracking free space and avoiding the need for manual space management.
- **Example**: Using ASSM with CREATE TABLESPACE commands helps eliminate the use of freelists, leading to better scalability and reduced contention in high-insert environments.

## 59. Resource Manager (DBRM)

- **Description**: The Oracle Resource Manager helps manage CPU, I/O, and other resources among database users and applications. It enables prioritization and resource allocation, ensuring critical workloads receive adequate resources.
- **Example**: Resource plans can be created and managed using the DBMS_RESOURCE_MANAGER package to limit the CPU usage of background and non-critical sessions.

## 60. SQL Trace and TKPROF

- **Description**: SQL Trace is a debugging tool that collects performance data for SQL statements. TKPROF processes the output of SQL Trace and generates a human-readable report detailing the execution of SQL statements.
- **Example**: Enable SQL Trace for a session with ALTER SESSION SET SQL_TRACE = TRUE, then use TKPROF to analyze the trace file and identify slow queries and optimization opportunities.

## 61. Auditing in Oracle

- **Description**: Oracle's auditing feature records database operations performed by users for security and compliance purposes. It helps track changes, data access, and administrative actions.
- **Example**: Enabling standard auditing or Unified Auditing can log activities like SELECT, INSERT, or changes to database objects, which are stored in DBA_AUDIT_TRAIL for review.

## 62. Archivelog Mode

- **Description**: When Oracle operates in ARCHIVELOG mode, redo logs are archived before being overwritten. This allows for complete recovery of the database to any point in time, ensuring data protection against failures.
- **Example**: Switching to ARCHIVELOG mode with ALTER DATABASE ARCHIVELOG ensures that all changes are logged, enabling recovery up to the last committed transaction.

### 63. Dynamic Performance Views (V$ Views)

- **Description**: V$ views (dynamic performance views) are in-memory views that provide real-time performance data about the database, sessions, wait events, and other metrics. They help DBAs monitor and troubleshoot the database.
- **Example**: V$SESSION provides information about current database sessions, while V$SYSTEM_EVENT shows wait events that affect database performance.

### 64. Explain Plan

- **Description**: The EXPLAIN PLAN statement displays the execution plan for a SQL query. It shows the steps Oracle takes to execute a query, such as table scans, index usage, and join methods.
- **Example**: Use EXPLAIN PLAN FOR SELECT * FROM employees; and view the plan with SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY); to understand how the query is processed.

### 65. Clustered Indexes

- **Description**: Oracle does not have clustered indexes like SQL Server, but it supports index-organized tables (IOTs), where the data is stored in the order of the index, providing faster access for specific queries.
- **Example**: Creating an index-organized table with CREATE TABLE employees (id NUMBER PRIMARY KEY, name VARCHAR2(50)) ORGANIZATION INDEX; ensures that the data is physically stored in the order of the id column.

### 66. Global Temporary Tables (GTT)

- **Description**: Global Temporary Tables are database tables that hold data specific to a session or transaction. Data in these tables is private to the session and is deleted automatically at the end of the session or transaction.
- **Example**: Create a GTT using CREATE GLOBAL TEMPORARY TABLE temp_sales (sale_id NUMBER, amount NUMBER) ON COMMIT DELETE ROWS;. Data inserted during a session is visible only to that session and is removed upon commit or session end.

### 67. SQL*Loader

- **Description**: SQL*Loader is a tool for high-speed data loading into Oracle databases from external files. It supports various loading methods, including conventional and direct path loads.
- **Example**: A control file specifies how to load data, such as LOAD DATA INFILE 'data.txt' INTO TABLE employees FIELDS TERMINATED BY ',' (emp_id, emp_name);. This command loads data from a CSV file into the employees table.

## 68. ASM Rebalance

- **Description**: ASM rebalance is an automatic process in Oracle Automatic Storage Management that redistributes data across disks in a disk group when disks are added or removed. This process ensures that data is spread evenly for optimal performance and space utilization.
- **Example**: When a new disk is added to a disk group, ASM starts a rebalance operation to move extents to the new disk, balancing the I/O load. The REBALANCE operation can be monitored using the V$ASM_OPERATION view.

## 69. Index Organized Tables (IOT)

- **Description**: Index Organized Tables store data in a B-tree index structure, where the table rows are stored in the order of the primary key. This structure provides fast retrieval for primary key-based queries.
- **Example**: Creating an IOT with CREATE TABLE products (product_id NUMBER PRIMARY KEY, name VARCHAR2(100)) ORGANIZATION INDEX; ensures that the product_id column is indexed and the data is stored accordingly, improving access times for queries based on product_id.

## 70. SQL Profiles

- **Description**: SQL Profiles are performance tuning tools that store auxiliary information for the optimizer to use when generating execution plans. This helps the optimizer choose better plans by providing additional context about data distribution and execution statistics.
- **Example**: Use the SQL Tuning Advisor to create a SQL Profile that the optimizer can use to improve the execution plan for a specific query without changing the SQL code itself.

## 71. Parallel Query Execution

- **Description**: Parallel query execution divides the workload of a query among multiple parallel processes, improving the performance of large-scale data retrieval and processing tasks.
- **Example**: When running a SELECT query with the PARALLEL hint, such as SELECT /*+ PARALLEL(4) */ * FROM large_table;, Oracle splits the operation across four parallel threads, reducing query execution time.

## 72. ASM Mirroring

- **Description**: ASM mirroring provides redundancy by storing copies of data across different disks in a disk group. It helps protect against data loss due to disk failure.

- **Example**: When creating a disk group with CREATE DISKGROUP dg_data NORMAL REDUNDANCY DISK '/dev/sd1', '/dev/sd2';, ASM stores two copies of the data for redundancy. This ensures that if one disk fails, the data remains accessible.

## 73. Partition Pruning

- **Description**: Partition pruning is an optimization technique where Oracle accesses only the relevant partitions of a table instead of scanning the entire table. This reduces the I/O workload and improves query performance.
- **Example**: A query like SELECT * FROM sales WHERE sale_date BETWEEN '2023-01-01' AND '2023-01-31'; will only scan the January partition of a partitioned sales table if partition pruning is correctly applied.

## 74. Bitmap Indexes

- **Description**: Bitmap indexes use bitmaps to represent data, making them suitable for columns with a low cardinality (few distinct values). They are efficient for complex WHERE clauses and multi-column filtering.
- **Example**: A bitmap index created on a status column with only a few distinct values (e.g., 'Active', 'Inactive') can improve the performance of queries that filter on this column, such as CREATE BITMAP INDEX idx_status ON employees(status);.

## 75. Check Constraints

- **Description**: Check constraints are rules enforced by the database to ensure that column values meet specific conditions. They maintain data integrity by restricting the data that can be inserted or updated in a column.
- **Example**: A salary column can have a check constraint to ensure it is positive: ALTER TABLE employees ADD CONSTRAINT chk_salary CHECK (salary > 0);. This prevents any entry with a non-positive salary.

## 76. Oracle GoldenGate

- **Description**: Oracle GoldenGate is a data replication tool that enables real-time data integration and movement between databases. It supports heterogeneous environments and is used for high-availability, disaster recovery, and data migration.
- **Example**: GoldenGate can replicate data changes from an Oracle production database to a reporting database to ensure reports run on a near-real-time dataset without impacting the primary system.

## 77. Redo Log Switch

- **Description**: A redo log switch occurs when Oracle fills a redo log file and begins writing to the next available log file. Log switches help in the continuous recording of transactions and facilitate checkpointing.
- **Example**: If the redo log group size is small and transaction volume is high, log switches will occur frequently, potentially causing log file switch (checkpoint incomplete) waits. This can be mitigated by increasing the size of the redo log files.

## 78. Oracle Enterprise Manager (OEM)

- **Description**: Oracle Enterprise Manager is a web-based tool for managing Oracle databases and applications. It provides a comprehensive set of tools for monitoring, tuning, and managing databases.
- **Example**: OEM can be used to monitor session activity, generate AWR reports, and run the SQL Tuning Advisor to identify performance issues and solutions.

## 79. Cost-Based Optimizer (CBO)

- **Description**: The CBO is Oracle's query optimizer that determines the most efficient way to execute a SQL statement based on statistics and the cost of various execution plans. It compares different query plans and chooses the one with the lowest estimated cost.
- **Example**: The CBO might choose an index scan over a full table scan for a query like SELECT * FROM orders WHERE order_id = 100; if the statistics show that using the index is more efficient.

## 80. Extent Management

- **Description**: Extents are collections of contiguous data blocks that Oracle allocates for database objects. Extent management can be either locally managed (using bitmaps) or dictionary managed (using data dictionary tables).
- **Example**: A tablespace created with EXTENT MANAGEMENT LOCAL uses bitmaps to track free and used extents, which reduces contention and improves space allocation efficiency.

## 81. Flashback Database

- **Description**: Flashback Database allows Oracle to revert the entire database to a previous point in time, enabling recovery from user errors like accidental deletions. It leverages flashback logs stored in the Flash Recovery Area (FRA).
- **Example**: To flashback the database to a specific SCN, use FLASHBACK DATABASE TO SCN 123456;. This operation rewinds the database to its state at the specified SCN.

## 82. DBMS_JOB and DBMS_SCHEDULER

- **Description**: DBMS_JOB and DBMS_SCHEDULER are packages for scheduling and managing jobs within Oracle. DBMS_JOB is simpler, while DBMS_SCHEDULER provides advanced features like job chaining and job windows.
- **Example**: Create a scheduled job with DBMS_SCHEDULER.CREATE_JOB, specifying job details like start time, repeat intervals, and job actions. This automates tasks such as data backups or report generation.

## 83. Oracle Data Guard Switchover and Failover

- **Description**: Switchover and failover are mechanisms in Oracle Data Guard to switch roles between the primary and standby databases. **Switchover** is a planned role reversal, while **failover** is unplanned and triggered by an emergency.
- **Example**: In a planned maintenance scenario, you can perform a switchover to a standby database using ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY;. Failover is used during a primary database failure to promote a standby database to primary.

## 84. Redo Apply and SQL Apply

- **Description**: **Redo Apply** is used in physical standby databases to apply redo logs and maintain data synchronization, while **SQL Apply** is used in logical standby databases to apply SQL statements.
- **Example**: Redo Apply uses redo logs to keep a physical standby up-to-date (ALTER DATABASE RECOVER MANAGED STANDBY DATABASE;), whereas SQL Apply applies changes in a logical format, allowing for data transformations.

## 85. Oracle Streams

- **Description**: Oracle Streams is a data replication and integration feature that captures and propagates changes between databases. It supports complex replication scenarios and data sharing across different Oracle environments.
- **Example**: Oracle Streams can be configured to replicate data changes from a source database to multiple target databases, enabling distributed data consistency and offloading read operations to replicas.

## 86. Cluster Interconnect in Oracle RAC

- **Description**: The **cluster interconnect** is the communication pathway between nodes in an Oracle RAC cluster. It ensures that data and messages are exchanged efficiently for maintaining cache coherency and distributing workload.
- **Example**: If the interconnect is slow or has high latency, you may see gc buffer busy wait events. Optimizing the interconnect can involve upgrading network bandwidth or tuning network settings.

## 87. Table Partitioning

- **Description**: Table partitioning splits large tables into smaller, more manageable pieces called partitions. This improves performance by allowing Oracle to scan only relevant partitions rather than the entire table.
- **Example**: Range partitioning can be implemented with CREATE TABLE sales (sale_id NUMBER, sale_date DATE) PARTITION BY RANGE (sale_date) (PARTITION p1 VALUES LESS THAN (TO_DATE('2023-01-01', 'YYYY-MM-DD'))); to organize data by date ranges.

## 88. Sparse Indexes

- **Description**: Sparse indexes include only certain rows in the index, typically those that meet a specific condition. This can optimize queries that focus on a subset of data.
- **Example**: A partial index on a status column might only include rows where status = 'Active', improving query performance for SELECT * FROM employees WHERE status = 'Active';.

## 89. ASM Striped and Mirrored Extents (SAME)

- **Description**: Striped and Mirrored Extents (SAME) in Oracle ASM provide a balance of striping data across disks for performance and mirroring it for redundancy.
- **Example**: By creating a disk group with NORMAL REDUNDANCY, ASM stripes data across disks and mirrors extents, ensuring data redundancy while optimizing read and write performance.

## 90. Oracle Scheduler Chains

- **Description**: Scheduler chains in Oracle allow for the creation of complex job sequences where jobs can run conditionally based on the success or failure of other jobs.
- **Example**: Using DBMS_SCHEDULER, you can create a chain with job dependencies, ensuring that data loading starts only after data extraction is complete and verified.

## 91. Data Compression

- **Description**: Oracle supports various types of data compression to reduce the storage footprint and improve I/O performance. Compression can be applied to tables, indexes, and backup files.
- **Example**: Using ALTER TABLE employees COMPRESS FOR QUERY HIGH; compresses table data to save space and improve query performance by reducing I/O requirements.

### 92. Flashback Versions Query

- **Description**: Flashback Versions Query allows you to view historical versions of rows over a specified time interval. This helps in auditing changes and recovering from accidental data modifications.
- **Example**: A query like SELECT versions_starttime, versions_endtime, salary FROM employees VERSIONS BETWEEN TIMESTAMP TO_TIMESTAMP('2023-10-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS') AND TO_TIMESTAMP('2023-10-31 23:59:59', 'YYYY-MM-DD HH24:MI:SS') WHERE emp_id = 101; shows the salary changes for an employee over time.

### 93. Oracle RAC Cache Fusion

- **Description**: Cache Fusion is an Oracle RAC technology that keeps data consistent across nodes by allowing nodes to share data directly from their memory instead of reading from disk.
- **Example**: When a query in Node A needs a block modified by Node B, Cache Fusion transfers the block directly over the cluster interconnect, ensuring high availability and performance.

### 94. Consistent Gets vs. DB Block Gets

- **Description**: **Consistent Gets** are logical reads of data blocks as they appeared at the start of a query. **DB Block Gets** are direct reads of data blocks as they are currently stored, without ensuring read consistency.
- **Example**: A SELECT query that scans data will use consistent gets to show data as it was when the query started. A SELECT FOR UPDATE might use DB Block Gets to access the current state of data.

### 95. Datafile Autoextend

- **Description**: Datafile autoextend is a feature that allows Oracle datafiles to automatically grow when they run out of space, preventing errors due to lack of space.
- **Example**: ALTER DATABASE DATAFILE '/u01/oradata/users01.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE 2G; ensures that the datafile expands in 10 MB increments up to a maximum size of 2 GB.

### 96. Oracle Restart

- **Description**: Oracle Restart is a feature that ensures that Oracle components (such as databases, listeners, ASM) are automatically restarted in the correct order when a server is restarted.
- **Example**: Oracle Restart can be configured using the srvctl utility to monitor and automatically start the database and related components after a server reboot.

## 97. Private vs. Public Synonyms

- **Description**: A **private synonym** is an alias for a database object created for a specific user and accessible only to that user, while a **public synonym** is available to all users in the database.
- **Example**: CREATE PUBLIC SYNONYM emp_view FOR hr.employees; allows all users to reference hr.employees using emp_view.

## 98. Invisible Indexes

- **Description**: Invisible indexes are not considered by the optimizer by default but can be used for testing the impact of adding or removing an index without actually dropping it.
- **Example**: Create an invisible index with CREATE INDEX inv_idx ON employees(last_name) INVISIBLE; and test performance before deciding to make it visible using ALTER INDEX inv_idx VISIBLE;.

## 99. Database Auditing Modes

- **Description**: Oracle provides different auditing modes such as **standard auditing** and **Unified Auditing**. These modes help track database usage and changes for compliance and security purposes.
- **Example**: Unified Auditing consolidates all audit records into one place and can be managed with DBMS_AUDIT_MGMT. You can set up auditing to capture actions like ALTER, DROP, and SELECT on specific objects.

## 100. Oracle Wallet

- **Description**: Oracle Wallet is a secure external location that stores authentication credentials, allowing secure password storage for database connections.
- **Example**: Oracle Wallet can be configured to eliminate hardcoded passwords in scripts by using mkstore and specifying database connection details securely.

## 101. Flashback Data Archive (FDA)

- **Description**: Flashback Data Archive (FDA) enables long-term storage of historical data changes, allowing for compliance with data retention policies and enabling historical queries.
- **Example**: Using DBMS_FLASHBACK_ARCHIVE to create an archive and associate it with a table ensures that historical versions of data are maintained and can be queried later.