Relational Database Administration Notes

Chuks Okoli

22 December, 2023

Week 1

Overview of Database Management Tasks

Typical day for a database administrator:

- Checking the state of the database. Database should be running with no errors.
- Check that scheduled backups have completed and resolve issues.
- Responding to support tickets.
- Meeting with developers and other stakeholders.
- Monitoring database activity.
- Determine appropriate server resources needed for planning a new database.

Database Management Lifecycle

The database lifecycle is shown in **Fig. 1**.

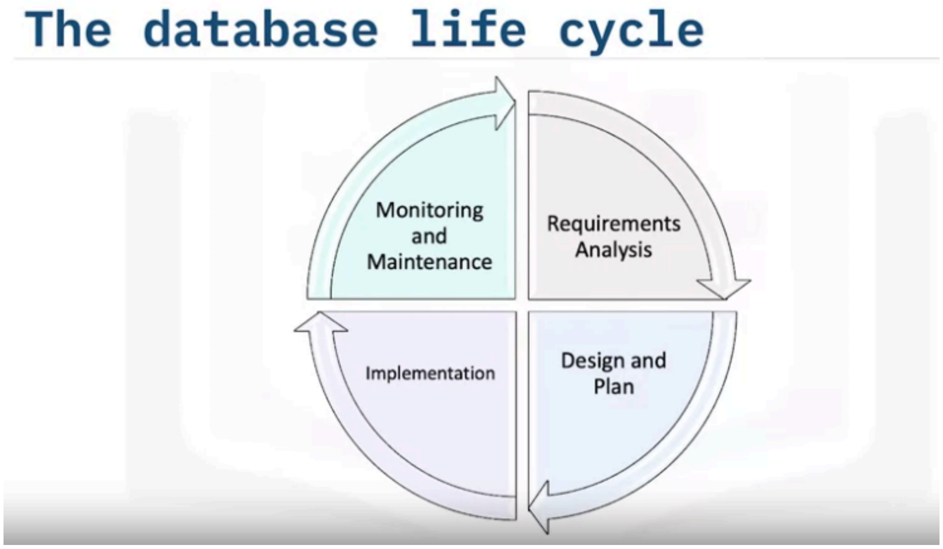

Figure 1: The database lifecycle

Requirement analysis

- Understand purpose and scope of the database
  - Establish what data is involved
  - Talk to the users and producers of the data, and develop samples of how users will use the data such as reports and dashboards

- Work with stakeholders such as Developers, Data Engineers, DBAs, Technology Managers, End-Users and determine requirements

Design and Plan

- Develop a plan for implementing the database by working with database objects such as instances, databases, tables, and indexes.
  - Develop database model showing which instances contains which databases and tables, how tables relate to each other, how users access data and so on.
  - Design Entity Relationship Diagrams or ERDs
- Determine appropriate server resources like storage space, memory, processing power, and log file size.
- Plan for how database objects are physically stored.

Implementation

- Create and configure database objects like instances, databases, tables, views, and indexes.
- Configuring database security, granting access for database users, groups, and roles
- Automates repeated database tasks such as backups, restores, and deployments to improve efficiency
- Import data from other databases, export data based on a query from a different source, or migrate projects from one environment to another, such as moving a project from the Application Development environment to the Production environment.

Monitoring and Maintenance

- Looks after the daily operations of the database
- Monitor the system for long-running queries and help end-users optimize them to run faster and not overuse system resources
- Review report and monitor activity, identify expensive queries, resource waits
- Apply upgrades and security patches to database software.
- Recommend and implement emerging database technologies
- Automate deployments and routine tasks such as backups whenever possible to keep processes working efficiently.
- Reviews logs and alerts, looking for failed logins and data access attempts to identify potential threats and vulnerabilities.
- Maintains database user and application permissions – revoking access to users and groups who should no longer have access and adding new users and roles as required to perform their jobs.

Server Objects and Hierarchy

Database Objects

- An instance is a logical boundary for a database or set of databases where you organize database objects and set configuration parameters.
- Common database objects are items that exist within the database such as tables, constraints, indexes, keys, views, aliases, triggers, events, and log files.
- Different RDBMSs use different names for their system objects. Most use the terms system schema, system tables, catalog, or directory.
- Database storage is managed through logical database objects and physical storage.

Week 2

Backup and Restore Databases

Backup and restore refer to the process of backing up data for protection purposes-restoring it after data loss from an unplanned shutdown, accidental deletion, or data corruption.

Backup and Restore scenarios

- Saving a copy of data for protection

- Recovering from data loss
  - After unplanned shutdown
  - Accidental deletion
  - Data corruption
- Move to a different database system
- Share data with business partners
- Use a copy of the data, e.g., in development or test environment

Physical vs. logical backups

- Logical backup
  - Contains DDL and DML commands to recreate database
  - Can reclaim wasted space
  - Slow and may impact performance
  - Granular
  - Backup/restore, import/export, dump & load utilities
- Physical backup
  - Copy of physical files, including logs, and configuration
  - Smaller and quicker
  - Less Granular
  - Can only restore to similar RDBMS
  - Common for specialized storage and cloud

What to backup

- Database
- Schema
- Tables
- Subset of data
- Other objects

Key considerations

- Check that your backup is valid
- Check that your restore plan works
- Ensure that your back up files are secure

Types of Backup

- Full backups - complete copy of all of the data in the object or objects that you are backing up
- Point-in-time backups - uses logged transactions to restore to an earlier point in time
- Differential backups - a copy of any data that has changed since the last full backup was taken
- Incremental backup - a copy of any data that has changed since the last backup of any type was taken

Backup Policies
Hot backup or online backup - are performed while the data is in use, while in Cold backup, the data is offline. Many decisions have to be made about backup policies such as:

- Physical or logical
- Full, differential, or incremental
- Hot or cold
- Compression
- Encryption
- Frequency:
  - Is data regularly changing or being added?
  - Is the existing table large?
- Schedule:
  - Is the data accessed equally across the 24-hour day?
  - Is it accessed at weekends?

- Automated backups
- Consider cloud backups

Summary of Backup and Restore Databases

- The types of backups are full, point-in-time, differential, and incremental.
- The difference between physical backups and logical backups, and between hot backups and cold backups.
- Your backup policy should be determined from your recovery needs and your data usage.
- Database transaction logs keep track of all activities that change the database structure and record transactions that insert, update, or delete data in the database.

Security and User Management
Authorization
Authorization are priviledges granted to users, and to groups or roles. Use the `SQL GRANT CONNECT` command to grant connection access to a particular database to a particular user, group, or role. To grant connection access to an individual user rather than a group or role, you simply specify the user name in place of the group name. To grant database or table access:

- To team, use `GRANT CONNECT TO ``salesteam''`
- To a specific person e.g., Joan, use `GRANT CONNET TO 'joan'`
- To grant select priviledges only on a table, use `GRANT SELECT ON mydb.mytable TO 'salesteam'`
- To grant insert priviledges, use `GRANT INSERT ON mydb.mytable TO 'salesteam'`
- To grant update priviledges, use `GRANT UPDATE ON mydb.mytable TO 'salesteam'`
- To grant delete priviledges, use `GRANT DELETE ON mydb.mytable TO 'salesteam'`
- To view a stored procedure, use `GRANT VIEW ON mydb.myproc TO 'salesteam'`
- To execute the stored procedure, use `GRANT EXECUTE ON mydb.myproc TO 'salesteam'`
- To change the definition of a procedure, use `GRANT ALTER ON mydb.myproc TO 'salesteam'`
- To remove granted priviledges, use `REVOKE SELECT ON mydb.mytable TO 'salesteam'`
- To deny any previous grant of a permission, use `DENY SELECT ON mydb.mytable TO 'salesteam'`

Summary of Security and User Management

- Authentication is the process of verifying that the user is who they claim to be. Authorization is the process of giving users permissions or privileges to access the objects and data in the database.
- When securing a database, you need to consider the security of the server and operating system, as well as the database and data.
- A database user is a user account that is allowed to access specified database objects. Groups are logical groupings of users to simplify user management. A database role defines a set of permissions needed to undertake a specific role in the database.
- When implementing role or group membership, use the principle of least privilege.
- Auditing does not directly protect your database but does identify gaps in your security.
- Customer-managed keys provide the data owner with more control over their data stored in the cloud.

Week 3

Monitoring and Optimization

What is database monitoring
One of the most challenging and necessary elements of database management is performance tuning, and one of the critical parts of this process is database monitoring. The term database monitoring refers to the different tasks related to the scrutinization of the day-to-day operational status of your database.

Database monitoring is crucial to maintain the health and performance of your relational database management system, regardless of which vendor's database product you are using. As a database admin, you can then utilize this useful information to perform several database monitoring tasks, including:

- Forecasting your future hardware requirements based on database usage patterns.

- Analyzing the performance of individual applications or database queries.
- Tracking the usage of indexes and tables.
- Determining the root cause of any system performance degradation.
- Optimizing database elements to provide the best possible performance.
- Assessing the impact of any optimization activities.

Reactive monitoring is done after an issue occurs, when you act in direct response to that issue; perhaps by fixing a configuration setting or adding more resources, for example. The most common situations when reactive monitoring occurs is either when your database security has been breached, or the performance of your database reaches critically low levels, or when some other kind of major database incident occurs that greatly impacts your business and therefore needs resolving as soon as possible. In contrast, a proactive monitoring strategy seeks to prevent this reactive panic by identifying issues before they grow into large problems. This is primarily achieved by observing specific metrics from your database and then sending alerts to interested parties if the values of these metrics reach abnormal levels. Proactive monitoring typically utilizes automated processes to perform tasks such as regularly verifying that a database system is online and accessible, verifying that configuration changes do not adversely affect the performance of the database system, and that the database system is operating and performing at acceptable levels. This proactive approach is widely considered to be the better strategy and is preferred by most database admins.

Summary of Security and User Management

- Database performance is measured by using key performance indicators, known as metrics, that enable DBAs to optimize organizations' databases.
- You should monitor at the infrastructure, platform, query, and user levels.
- A database diagnostic log file, also known as an error log or troubleshooting log, contains timestamped messages for various types of informational messages, events, warnings, and error details.
- Database optimization commands include OPTIMIZE TABLE in MySQL, VACUUM, and REINDEX in PostgreSQL, and RUNSTATS and REORG in Db2.
- Query execution plans show details of the steps used to access data when running query statements.

Week 4