Introduction

This geospatial fact sheet is the second in a two-part series outlining important aspects of geodatabase development and use. The first part, Geodatabase 101, discusses geodatabase fundamentals along with design and development considerations, while this part addresses geodatabase management, maintenance, and troubleshooting.

In this paper, geodatabase management refers to tasks typical of what a Database Administrator (DBA) or Systems Administrator handles. Geodatabase maintenance refers to tasks typical of what an ArcGIS Geodatabase Administrator would handle. We acknowledge that your organizational structure and context plays an important role in who or whom performs these tasks for your organization. We explore what that means throughout this fact sheet.

Geodatabase Management

Good geodatabase management starts in the design and development phase of deploying a geodatabase. There are several different relational database management systems (RDBMS) that support geodatabase functionality, and they all have their own requirements for proper management. The spatial software you use on top of the RDBMS may introduce other considerations. However, there are key things that need to be considered across all RDBMSs:

- **Deployment Strategy**
  - Cloud vs. on premise
    - Infrastructure as a service vs. database as a service options
    - IT infrastructure and staffing considerations for on premise management
    - Cost and level of effort to manage

- **Data Management**
  - Master Data Warehouse style setup or separate databases for different purposes
  - Automating data updates/migration via SQL stored procedures or GIS-based tools

- **User access and permissions**
  - Instance level logins with permissions at the database, table, or row level
  - Use of database users vs. Active Directory users
  - Using roles to assign permissions over individual level permissions
  - Automating user and permission management at the RDBMS level
  - Application access vs. individual user access

- **Testing and reporting**
  - Setup repeating database “stress” tests to ensure proper function
  - Automating reporting of trends and database metrics
  - Setting up alerts when specific behaviors are triggered
Proper geodatabase management involves proper database setup and configuration. Understanding these considerations and how they apply in your situation will ease the level of effort that will need to be expended weekly, daily, or even hourly managing your geodatabases. In addition, ensuring that key metrics are actively monitored and alerts are set up via triggers will ensure you are notified as needed. This section is not intended to be a full review of relational database management but rather a starting point to get you on the right path. If any of these concepts are new to you or you are unsure of what is needed in your situation, further research may be required.

**Geodatabase Maintenance**

This section covers maintenance items related to general relational database maintenance.

As was the case in the geodatabase management section, topics in this section differ slightly between different RDBMSs and the spatial software you use. Below are some key things to consider regarding general relational database maintenance.

- **Database Backups**
  - Backup both system and user databases
  - Full vs. incremental (transaction) backups on user databases
  - How often will depend on the frequency of editing but, generally, you would want at least a daily backup of your databases?
  - Automate backups via RDBMS tools or SQL script
- **Rebuild Indexes**
  - As mentioned in Geodatabase 101, indexing in a relational database involves creating a data structure to improve the speed at which data is retrieved from the database
  - How often indexes should be rebuilt for each table depends on the editing traffic
  - Automated is usually preferred for ease of management, but can also be done manually?
- **Update Query Optimization Statistics**
  - RDBMSs may utilize query plans to determine how best to run the queries they receive
  - To optimize these query plans, statistics are used so the software can choose the best plan to run at the time
  - This is generally only needed if you want to update statistics more often than the default RDBMS configuration

It should be noted this is not a comprehensive list, and whether and how often you do each of these tasks is extremely situation dependent.

The spatial software you use on top of the RDBMS may also have required or recommended maintenance in addition to the above tasks. Other RDBMS functionality, such as versioning to manage multi-user edits, introduces other maintenance considerations. Review software documentation to ensure you are meeting the required and recommended maintenance tasks for the options and use cases you pursue.
Geodatabase Troubleshooting

Like the geodatabase maintenance section, geodatabase troubleshooting can have different workflows specific to general RDMS troubleshooting and spatial software troubleshooting. In architectures where client spatial software sits on top of an RDBMS, the problem may be with the client software, the underlying RDBMS, or the connections between the two.

For general RDBMS troubleshooting, the best tools for identifying issues are database tracing functions. Database traces allow users to view specific actions happening behind the scenes when a request is sent to the database. Each RDBMS has their own tools for tracing and their setup is slightly different. However, below are some general things you may pay attention to when troubleshooting an issue.

- **SQL statements**
  - What are the actual SQL statements being sent from the client to the database?
- **Errors**
  - If troubleshooting an error message in the client, it may be helpful to show error messages in the trace.
- **Query plan**
  - This is especially important for troubleshooting performance issues. What is the query plan that is being executed by the database?
- **Batch requests / stored procedures**
  - This is helpful to look at if the action being taken on the client side is a result of a stored procedure being run in the database.

Each RDBMS is slightly different in what you will be able to trace. Your decision on what to include in a trace will also be dependent on the issue that is happening (an error vs. a performance issue). You need to ensure you fully understand the problem before setting up a database trace. What are the inputs on the side? What is the end result supposed to be? How does the result I am seeing differ from what I expect to see? These are all very important troubleshooting questions to answer before beginning.

Review the documentation for the spatial software you are using on top of the RDBMS. The options each system provides vary, but software usually includes tools and options to troubleshoot issues with the client software and connections to the RDBMS.

**Resources**

<table>
<thead>
<tr>
<th>Esri</th>
<th><a href="#">Database Maintenance Strategies</a></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="#">An overview of traditional versioning</a></td>
</tr>
<tr>
<td></td>
<td><a href="#">How To: Diagnose ArcSDE connection and performance issues using SDE Intercept</a></td>
</tr>
<tr>
<td>Wikipedia</td>
<td><a href="#">Relational Database</a></td>
</tr>
</tbody>
</table>

---

**About URISA**

URISA is an multi-disciplinary geospatial organization that provides professional education and training, a vibrant and connected community, advocacy for geospatial challenges and issues, and essential resources. URISA fosters excellence in GIS and engages geospatial professionals throughout their careers. For more information, visit [www.urisa.org](http://www.urisa.org).