Database Administration

Server machine Linux Windows Disk Configuration
Configuring server software Logs Configuring for InnoDB
Optimising for speed and size
Reliability Replication OLTP/OLAP Distribution
Disaster Planning - checks, backups, repairs

If you have a database, small or large, you have a database administrator (DBA). If your organisation is small and you are its one and only software developer, you *are* the DBA. At the other end of the continuum, your organisation may be big enough to support a DBA group and you may be but one cog in it, with sharply defined responsibilities.

A simple, troublesome paradox governs the working life of most DBAs. To the extent a database-driven application succeeds, its performance degrades. The better you do your job, the closer you get to failure.

Why? Because if the application is a success then in all likelihood the number of users is increasing, their expectations are driving more and more change, data is accumulating at increasing rates, and demands on all parts of the system inexorably push the system closer to critical slowdown and failure thresholds.

Leo Gerstner, a former CEO of IBM, said "Inside IBM we talk about ten times more connected people, a hundred times more network speed, a thousand times more devices, and a million times more data" (*http://www.informationweek/816/gerstner.htm*). To contain the paradox of success, you build success and its consequences into all aspects of the system design. So DBA responsibilities extend to all aspects of servers and the databases that reside on them.

Table 17-1: Application Life Cycle

Life cycle stage	***Product***
Requirements analysis	Functional specification, UML diagrams, use cases
Logical design	Logical data design, user interface design, business rules
Database design	Database
Application design	Application model
Application development	Application
Application testing	Deployable application
Deployment	Working application
Maintenance	Mature working application
Application retirement	Data moved to new DBMS

In the life cycle of a MySQL application (Table 17-1), the DBA enters at the *database design* stage, and stays till the application is retired. She has specific responsibilities at each stage:

- *Database design*: Transform the logical data model into a physical database implementation that meets application demands; plan server hardware acquisition and allocation; set up database support for security and maintenance functions.
- *Application design*: Help application designers develop a security scheme, queries, updates, backup and recovery, and replication or mirroring.
- *Application development*: Analyze and manage database modifications; begin to log and analyze the application's interaction with the database.
- *Testing:* Thoroughly analyse and test mockups of DBMS workload, throughput, optimization, contention, backups, disaster planning and recovery.
- *Deployment*: Install the database on production servers, and ensure that the database is responding correctly to the application.
- *Maintenance*: Continue analysis, maintenance and tuning of every aspect of the database; participate in code refactoring; prevent disasters and manage recovery from them.
- *Retirement*: copy the data to the new DBMS, perhaps set up a read-only archive (OLAP) version of the database.

To read the rest of this and other chapters, *buy a copy of the book*

TOC ***Previous*** ***Next***