

# What is a DBA?

Mullins chapter 1

Bjarte Wang-Kileng

HVL

Februray 9, 2026



**Western Norway  
University of  
Applied Sciences**

# Outline

- 1 Database Administration
- 2 DBA Tasks
- 3 DBA types
- 4 Other DBA issues

# Outline

1 Database Administration

2 DBA Tasks

3 DBA types

4 Other DBA issues

# Databases, DBMS and DBA

- ▶ A database is an organised store of structured data where the data is accessible by named data elements.
- ▶ A DBMS (Database Management System) is software that enables end users and application programs to share and manage data.
- ▶ A DBA (Database Administrator) is a person responsible for the databases, DBMS and application programs using them.

# Examples of DBMSs

- ▶ Oracle
- ▶ Sybase
- ▶ MySQL
- ▶ MariaDB
- ▶ IBM DB2
- ▶ Microsoft SQL Server
- ▶ Sybase
- ▶ PostgreSQL
- ▶ EnterpriseDB
- ▶ InterBase
- ▶ Berkeley DB
- ▶ CA IDMS
- ▶ IBM Information Management System (IMS)
- ▶ ADABAS D
- ▶ MaxDB
- ▶ Microsoft Access
- ▶ Paradox
- ▶ FileMaker
- ▶ ADABAS
- ▶ Apache CouchDB
- ▶ Cassandra
- ▶ MongoDB
- ▶ Infinispan
- ▶ Riak
- ▶ ObjectStore
- ▶ Versant Object Database

# Administrator roles with database responsibilities

- ▶ Data Administrator (DA).
- ▶ Database Administrator (DBA).
- ▶ System Administrator (SA).

# Data Administrator (DA)

- ▶ Most involved in the early stages of the application development life cycle.
- ▶ Identifying and cataloguing the data required by the users.
- ▶ Produce conceptual and logical data models.
- ▶ Describing the data (meta data, ie. what is the meaning of the data, domain of data, data types, etc.).
- ▶ Setting data policies for the organization.
  - Privacy policies – Norway, EU and others have legislation concerning data privacy
- ▶ Setting standards for control and usage of data.
- ▶ Identifying data owners and users.

# Database Administrator (DBA)

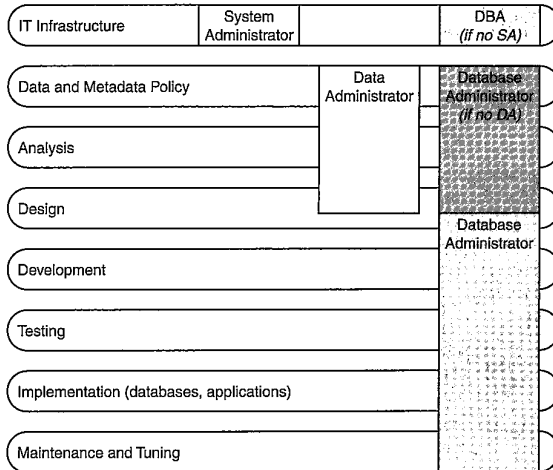
- ▶ Transform logical data models into an efficient physical database designs.
- ▶ Ongoing support of the databases.
- ▶ Management of the applications that access the databases.



# System Administrator (SA)

- ▶ Responsible for the upkeep, configuration, and reliable operation of the computer system.
- ▶ Installation, setup and support of the DBMS.
- ▶ Setup and configure the computer system and DBMS and to make the DBMS operate effectively.

# Data-, Database- and System Administrators



**Figure 1-5** DA, DBA, and SA responsibilities

# Outline

1 Database Administration

2 DBA Tasks

3 DBA types

4 Other DBA issues

# DBA Tasks

- ▶ Performance monitoring and tuning.
- ▶ Availability.
- ▶ Database security and authorisation.
- ▶ Backup and recovery.
- ▶ Data integrity.
- ▶ Maintenance of test and production.
- ▶ DBMS release migration.

# Database design<sup>1</sup>

- ▶ Understand relational theory and the technical details of the specific DBMSs.
- ▶ If no DA – Conceptual and logical data modelling.
- ▶ Transform the logical data model into a physical DBMS implementation.
- ▶ Database design takes a small portion of DBA time. Most time to administering and tuning.
- ▶ Poor relational design can give poor performance.

---

<sup>1</sup>Covered in depth later in course.

# Performance monitoring and tuning<sup>2</sup>

- ▶ *Workload* – The amount of work requested of the DBMS:
  - Transactions, batch jobs, queries etc.
- ▶ *Throughput* – The capability of the computer hardware and software to process data.
- ▶ *Contention* (Konflikt/kamp/strid) – Workload components are fighting for the same and non sharable resources.
- ▶ *Optimisation* – Generate efficient access path to data:
  - Internal to the DBMS: Cache, indexes, etc.
  - SQL optimisation.

---

<sup>2</sup>Covered in depth later in course.

# Availability<sup>3</sup>

- ▶ Keep the DBMS running.
- ▶ Monitoring and automated alerts.
- ▶ Design the database so it can be maintained with minimal disruptions.
- ▶ Clustering, replication etc.

---

<sup>3</sup>Covered in depth later in course.

# Database security and authorisation<sup>4</sup>

- ▶ All levels of DMBS access must be authorised. e.g.:
  - Reading/modifying/creating/dropping of databases/tables/rows/columns/views/procedures.
  - Starting and stopping DBMS.
  - Setting and modifying parameters.
  
- ▶ Can create *Views* to block access to sensitive data.

---

<sup>4</sup>Covered in depth later in course.



# Backup and recovery<sup>5</sup>

- ▶ Failure in hardware, software, human.
- ▶ 80% are human- or software errors.
- ▶ DBA is responsible for recovering data.
- ▶ Hardware failure:
  - Recover to current.
- ▶ Human or software error:
  - Recover to point in time (PIT).
  - Transaction recovery – Remove effects of specific transactions during a specified time frame.

---

<sup>5</sup>Covered in depth later in course.

# Data integrity<sup>6</sup>

- ▶ Physical data integrity – Data types and data domains:
  - Choose appropriate data types for columns.
  - Using constraints.
- ▶ Semantic data integrity – E.g. correct address, phone number, e-mail of person:
  - Must be obtained through application code business logic.
  - Need procedures to make redundant data consistent.
- ▶ Internal data integrity – Index consistency, backup consistency, etc.

---

<sup>6</sup>Covered in depth later in course.

# Outline

1 Database Administration

2 DBA Tasks

3 DBA types

4 Other DBA issues

# DBA types

## DBA types

Some (big) organizations can split the DBA responsibilities into separate jobs.

# System DBA

- ▶ Used only if no SA (System Administrator).
- ▶ Overlaps with the SA-tasks.
- ▶ Installing new DBMS versions and applying maintenance fixes from DBMS vendor.
- ▶ Installing DBA tools and utilities.
- ▶ Setting and tuning system parameters.
- ▶ Tuning of OS, network etc.
- ▶ Ensuring and enabling appropriate storage for the DBMS.
- ▶ Interfacing with other technologies required by database applications.

# Application DBA

- ▶ Database design and support for specific applications.
- ▶ Write and debug complex SQL.
- ▶ Performance tuning.
- ▶ Database change management.
- ▶ Pros:
  - Better service for application.
- ▶ Cons:
  - Isolated from rest of DBA team. Loses sight of the overall data needs and resources.

# Other roles

- ▶ Database Architect – Not common. New design and development only.
- ▶ Database Analyst – No clear definition. Involves tasks from the other roles.
- ▶ Data Modeller – Subset of DA role.
- ▶ Task-Oriented DBA – Focus on a specific DBA task, eg. *Performance Analyst*.
- ▶ Data Warehouse Administrator.

# Outline

- 1 Database Administration
- 2 DBA Tasks
- 3 DBA types
- 4 Other DBA issues



# Multiplatform DBA Issues

- ▶ Multiplatform or single platform DBA?
- ▶ Many similarities between DBMSs.
- ▶ Different features and nuances between DBMS's. Difficult to be an expert on many platforms.
- ▶ Use specialist DBAs for the heavily used DBMSs.
- ▶ Share responsibility for less used DBMSs.
- ▶ Use multiplatform DBA tools.
- ▶ Make guidelines for which DBMS to use in what situations.

# Test and production

- ▶ Two separate environments must be maintained, *test* and *production*.
- ▶ Development and maintenance is first performed in the test environment.
- ▶ The test environment contains a subset of the data in production.
- ▶ The test environment must contain enough data for acceptable application testing.
- ▶ The test environment should be restored before different runs of tests:
  - Need automated procedures.

# Business rules in the DBMS<sup>7</sup>

- ▶ Stored procedures – Program stored in and executed by the DBMS.
- ▶ Triggers – Event driven procedures. Code is triggered automatically e.g. when data is inserted into a table.
- ▶ User defined functions – Like *stored procedures*, but return values.

---

<sup>7</sup>Covered in depth later in course.

# Internet, e-business and DBA

- ▶ 24/7 availability.
- ▶ New technologies (XML, JSON, Java, PHP, Web services).
- ▶ Integrating legacy data with modern web-based applications.
- ▶ Web-based administration.
- ▶ Databases in the Cloud.
- ▶ NoSQL.
- ▶ Unpredictable workload.
- ▶ Synchronization of data to and from laptops and mobile phones.
- ▶ Laptops and phones with data can be lost or stolen.

# NoSQL – Not only SQL

- ▶ Tied to big data.
- ▶ Low cost storage and access to large amount of data.
- ▶ Nonrelational, distributed, flexible and scalable.
- ▶ Data typically accessed one way only. No flexibility in generating ad hoc queries.
- ▶ Common attributes are simple to use, no schema, replication support, “eventually consistent”.
- ▶ Little database administration work.
- ▶ Examples: [MongoDB](#), [Apache CouchDB](#), [Infinispan](#), [Apache Cassandra](#), [Scylla](#).