DBMS Internals: Exploring the Engine Room

MERiT

This module explores the inner mechanics of modern relational database management systems, using PostgreSQL and MySQL as primary examples.

You'll learn how databases store data, manage memory, track metadata, and more. By the end, you'll apply these concepts in live environments.

Understanding Storage Approaches

MERiT

Storage Type	Primary Use	Read Efficiency	Update Efficiency
Row-Store	OLTP (transactions)	Entire row fast	Very efficient
Column-Store	OLAP (analytics)	Single column fast	Less efficient

Row storage places complete records together. Column storage groups all values from one field.

PostgreSQL and MySQL are primarily row-oriented. Column-stores excel at analytics workloads.

Data Blocks: The Physical Foundation

PostgreSQL

Uses 8KB blocks by default

Contains rows, block ID, pointers

MySQL InnoDB

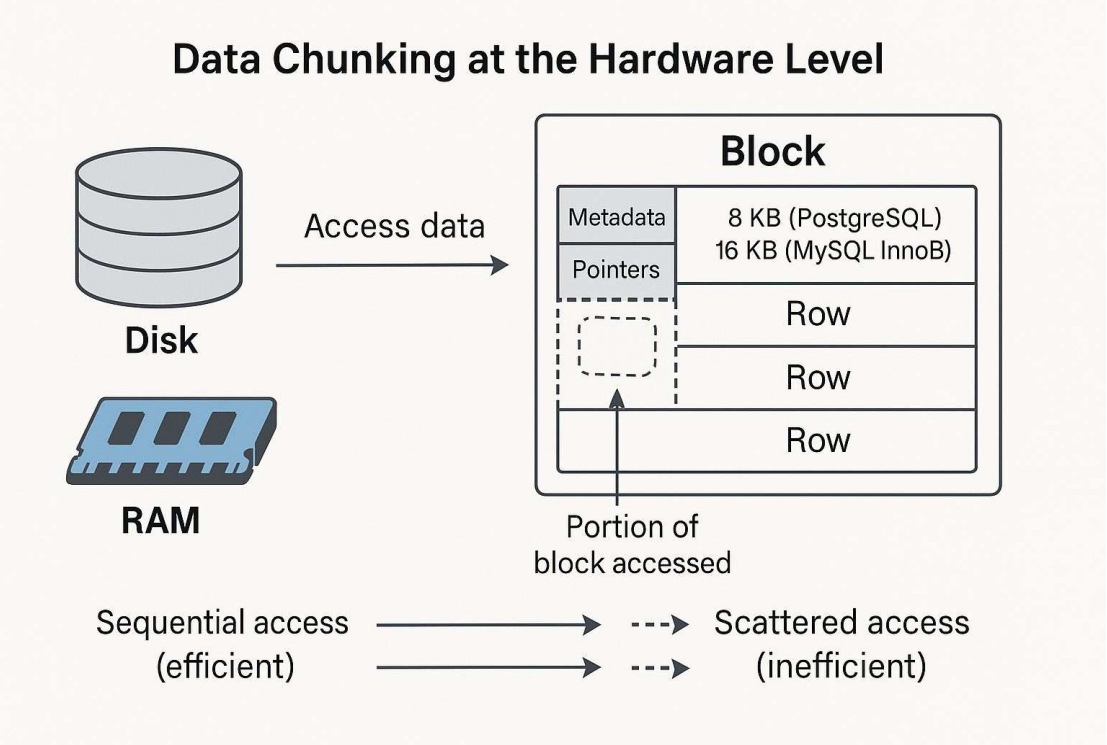
Often uses 16KB pages

Includes checksums and metadata

Access Pattern

DBMS always reads whole blocks

Never accesses individual bytes



Blocks **bundle** data for storage efficiency. Each contains multiple rows and essential metadata.

Buffer Pools: The Memory Layer

Request Data

Query needs specific rows

Check Buffer

Look for data in RAM first

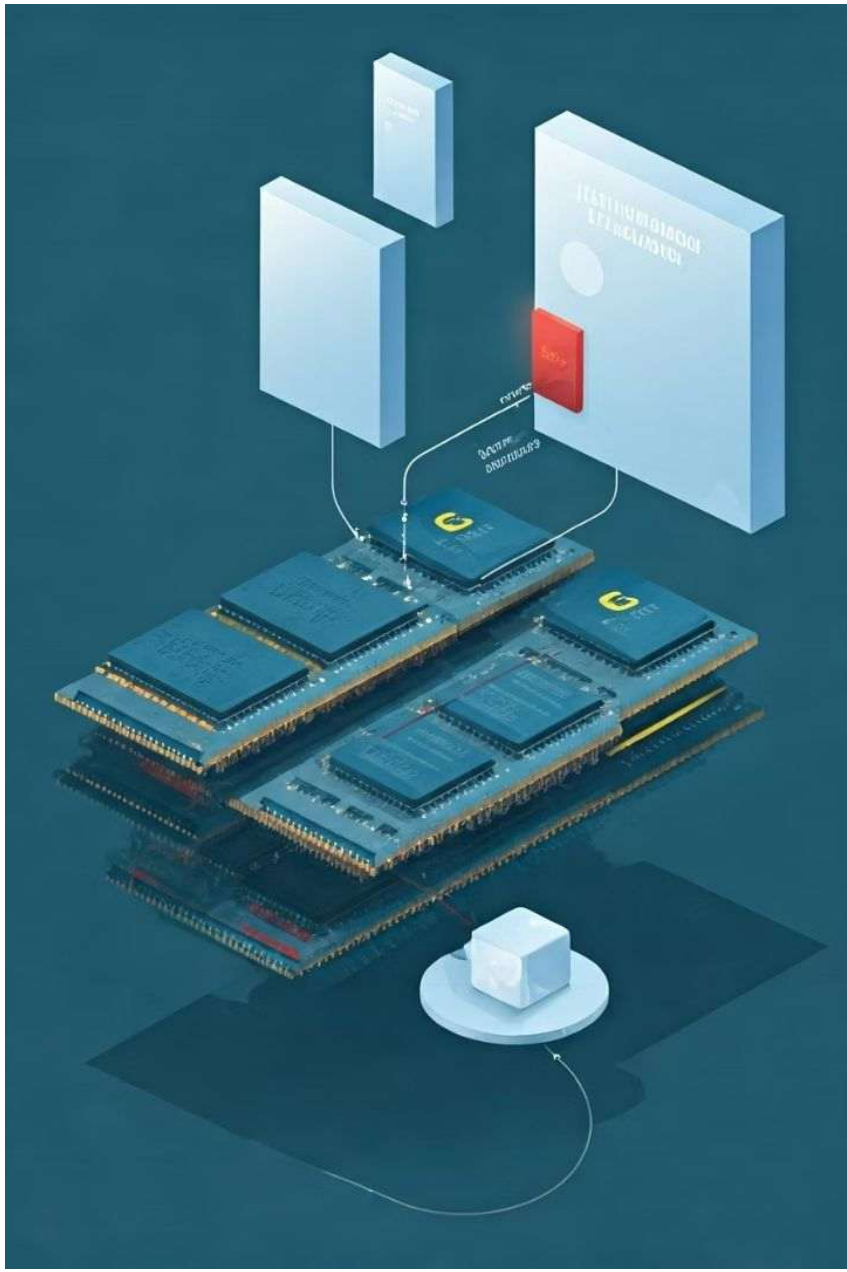
Serve or Load

Return cached data or read from disk

Update Buffer

Cache new blocks, evict old ones

Buffer pools dramatically boost performance by keeping hot data in RAM. They reduce slow disk I/O operations.



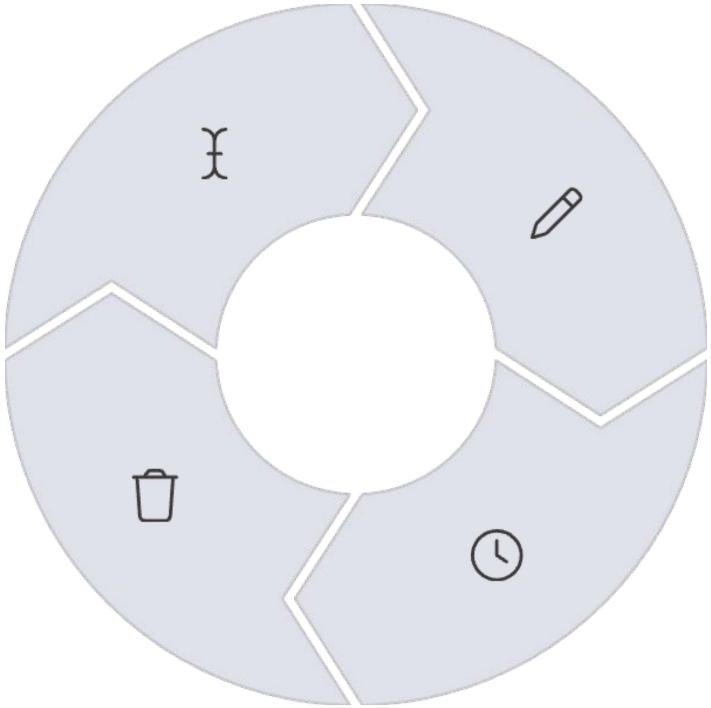
Buffer Management Strategies

Modify Data

Changes create "dirty" pages in buffer

Write Log

WAL/REDO log ensures crash recovery



Eviction

LRU algorithm removes least-used blocks

Checkpoint

Flush dirty pages to disk periodically

Smart buffer management balances performance with data safety. It uses algorithms to decide what stays in RAM.

System Catalogs: The Metadata Layer

Schema Information

Tables, columns, data types, constraints

Access Controls

Users, roles, privileges, security policies

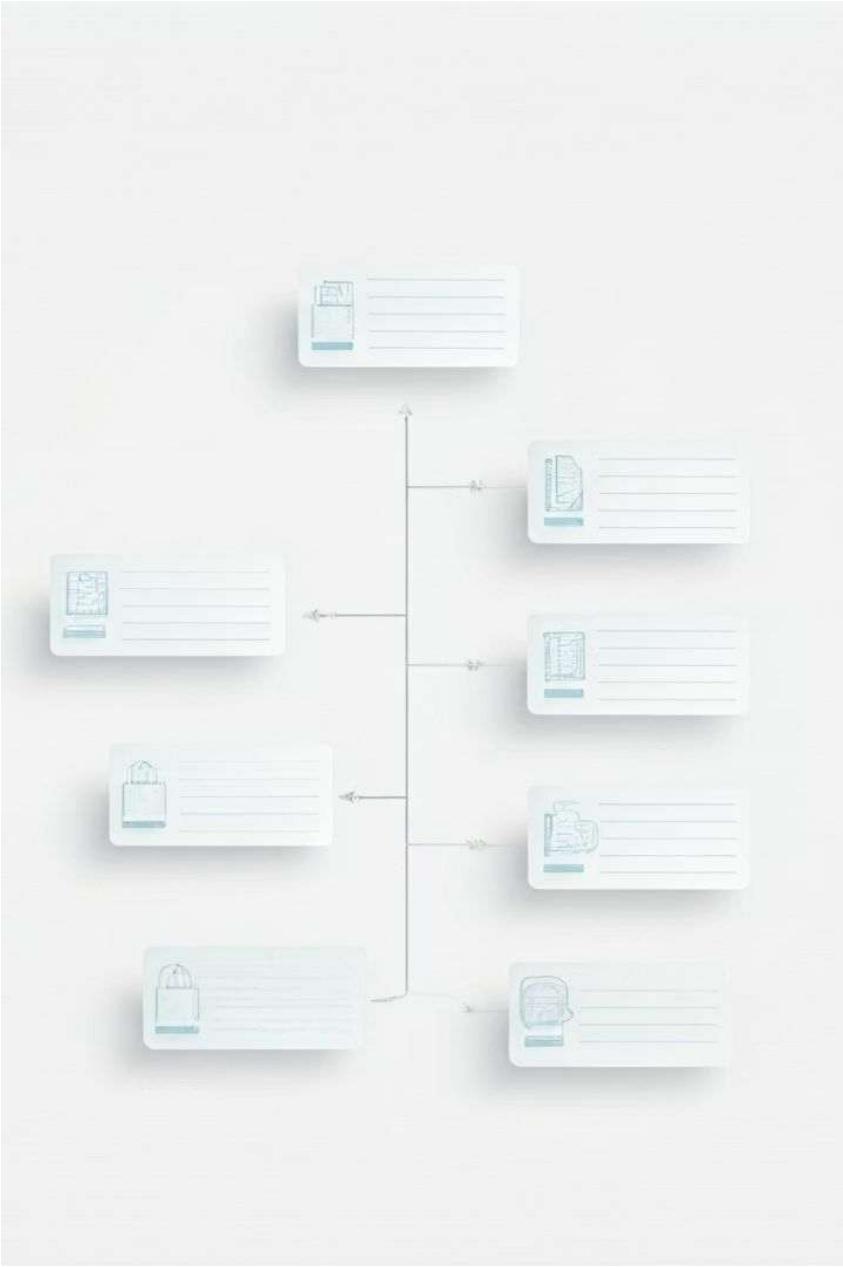
Statistics

Row counts, index usage, query patterns

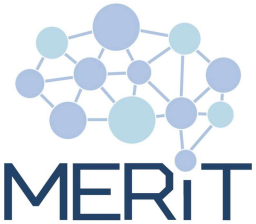
Configuration

Parameter settings, runtime variables

System catalogs store crucial metadata about your database. They let you query the database about itself.



PostgreSQL System Catalog



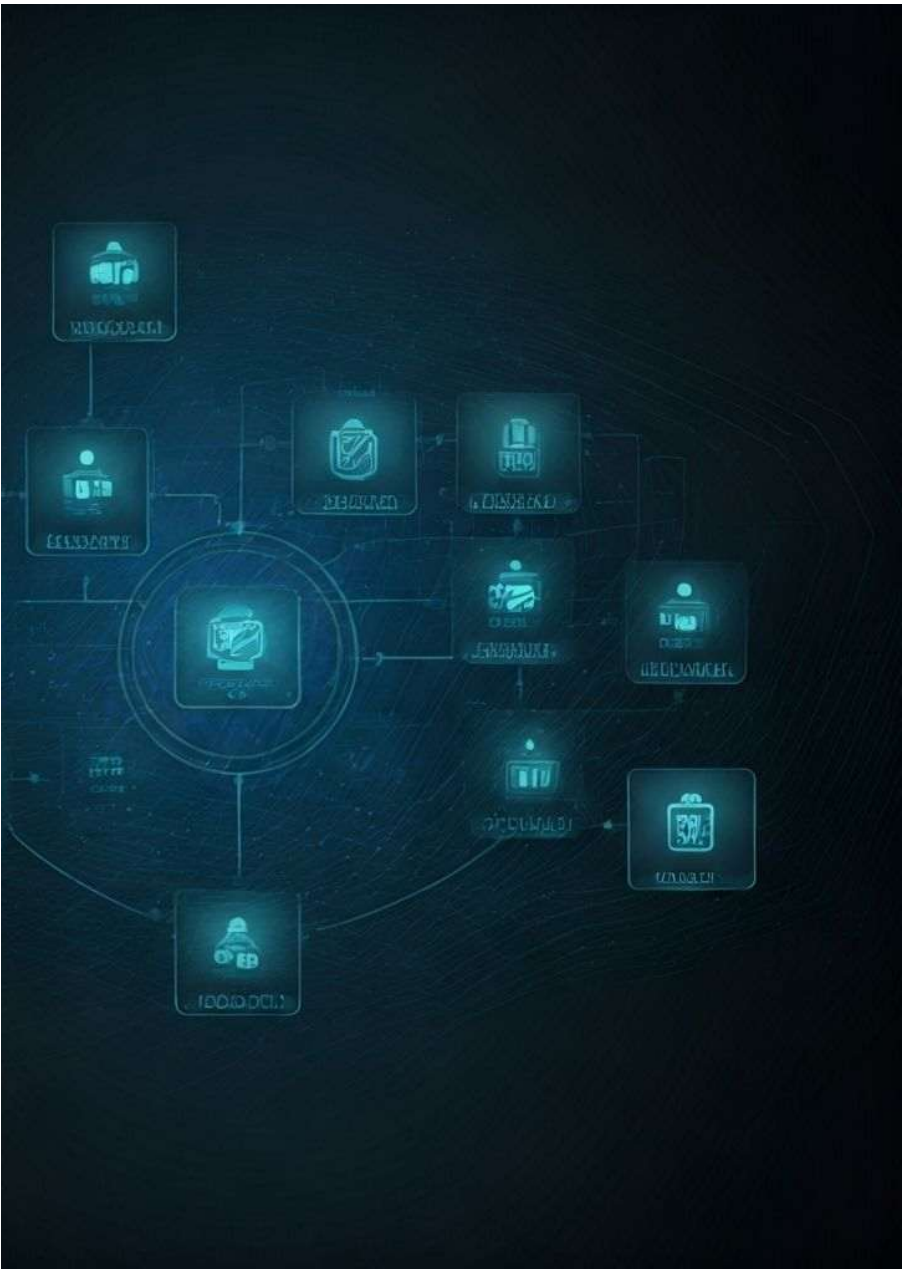
Key Catalog Views

- pg_tables: lists all tables
- pg_class: all relations
- pg_attribute: column definitions
- pg_index: index definitions
- pg_constraint: constraints

Sample Query

```
SELECT tablename FROM pg_catalog.pg_tables
WHERE schemaname='public';
```

The pg_catalog schema contains over 400 tables and views. It exposes all database internals.



MySQL System Catalog

MERiT

tables

Lists all tables and views in the database. Core component for schema discovery.

columns

Contains metadata about column properties. Essential for type inspection.

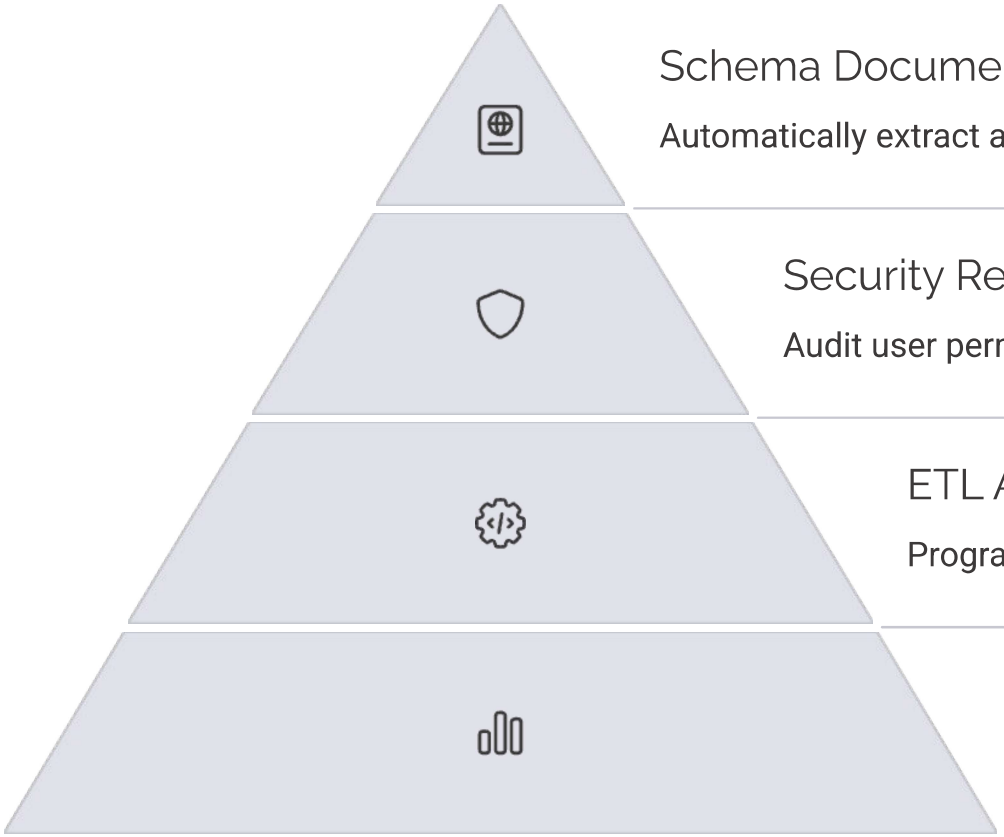
key_column_usage

Reveals columns used in constraints and keys. Critical for dependency analysis.

statistics

Provides index information. Vital for performance tuning and optimization.

Real-World Applications



Schema Documentation

Automatically extract and report on database structures

Security Reviews

Audit user permissions across database objects

ETL Automation

Programmatically identify and migrate specific structures

Monitoring

Detect schema drift and changes over time