
LUT
University
LAND
OF THE
CURIOUS

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

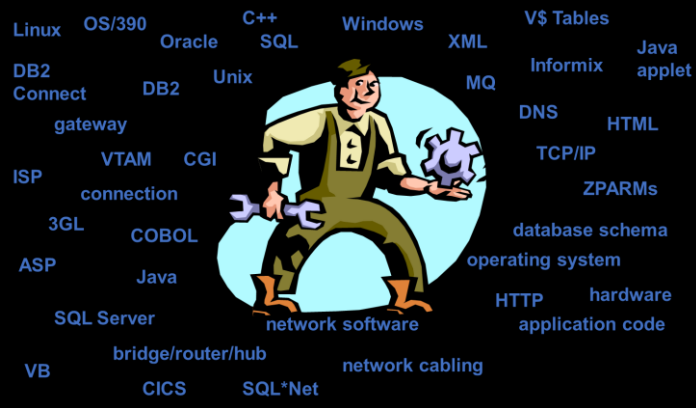
COURSE SUMMARY / RECAP

Lecture

Jiri Musto, D.Sc.

BEING A DATABASE ADMINISTRATOR

- In charge of creating and managing the database environment
  - Specific responsibilities can be delegated to other workers
- Need extensive knowledge on databases as well as other systems/applications that are connected
  - Analysis tools, programming languages, web development, etc.
- Challenges as database administrator
  - Ever-changing technologies, small changes can have a huge impact
  - Technologies used in business vary
  - Required knowledge extends outside of databases



BUILDING THE DATABASE ENVIRONMENT

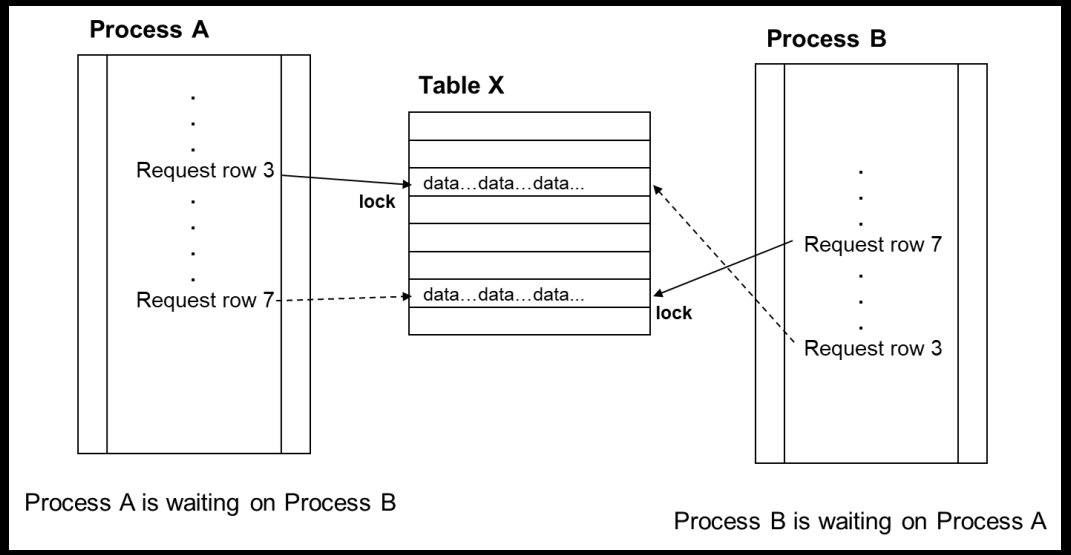
- Choose the hardware and software
  - Multiple factors influence the final decision including business needs and availability of tools & experts
- Different technologies have their own perks
  - Cloud DBMS vs. Local DBMS
- Upgrading the DBMS should be carefully evaluated
- Specify standards and procedures
  - Guidelines
  - Conventions
  - Roles and responsibilities

DATABASE APPLICATION DESIGN

- To design an application that connects to the database, you need to know
  - How data is stored
  - How to code and embed SQL statements
  - How to avoid potential database processing problems
- Applications connecting to the database can be different
  - Desktop applications
  - Web applications
  - Different programming languages

DATABASE TRANSACTIONS AND LOCKS

- Locks are used to enable concurrent usage
  - Users, applications
- Different types of locks & different level of locking granularity
  - **Type**: Shared, exclusive, update, intent
  - **Granularity**: Row, table, database
  - Granularity can be escalated during the process
- Locks inevitably lead to users accessing the same data
  - May result in deadlocks or timeouts



DATABASE CHANGE MANAGEMENT

- Multiple types of changes can lead to database change
  - Physical environment
  - Organization
  - Network
  - Application & System
  - Data
- Use additional tools to help manage changes
  - Track changes over time
  - Analyze impact of changes
  - Automate when possible

CHALLENGES OF DATABASE CHANGE

- Some changes are not supported by the ALTER command or DBMS in general
  - Rearranging / removing columns, changing primary / foreign keys,
- Some changes are easier than others
  - Adding a column at the end of a table vs. Adding a column to the middle
- Changes in database cause applications and logical designs to change and vice versa
  - Rollback changes if one part fails

DATA AVAILABILITY

- Availability of 24/7 is a dream
  - Something that is strived for but never achieved
- Database needs to be routinely maintained
  - Smaller tasks may not cause downtime
  - Larger tasks (such as updates) will cause downtime
- Cost of assuring availability vs. Cost of downtime
- Multiple issues can cause availability problems
  - Failures, hardware / software problems, outages
- Availability can be improved with design choices
  - Cluster, partitioning, distribution, RAID

PERFORMANCE MANAGEMENT

- Performance is affected by:
  1. Workload
  2. throughput
  3. Resources
  4. optimization
  5. Contention
- Problem in performance may be in:
  1. Connecting application
  2. Database
  3. System / subsystem
  4. Environment

PERFORMANCE MANAGEMENT

- Management steps
  - Monitor
  - Analyze
  - Fix
- Proactive is better than reactive
- Implementing changes is cheaper in the early steps of development
- Tuning performance
  - Tune one component at a time
  - Do not over tune
  - Accept reality

SYSTEM PERFORMANCE

- DBMS have different requirements and methods of configuration
- Disk storage and I/O
  - I/O operations are the largest bottleneck
    - Caches may help
    - Caches consume memory
  - Disk vs. Cloud, HDD vs. SSD
- Memory management
  - All operations require memory
    - Connections, open databases / objects, locks, caches
  - Balance the cost of memory with return of investment

DATABASE PERFORMANCE

- Optimize database by
  - Efficient structure
  - Efficient SQL queries
  - Defining indices
- SQL tweaking and system tuning are not a solution to overcome a poorly designed database
- Parallelism, partitioning and distribution may help

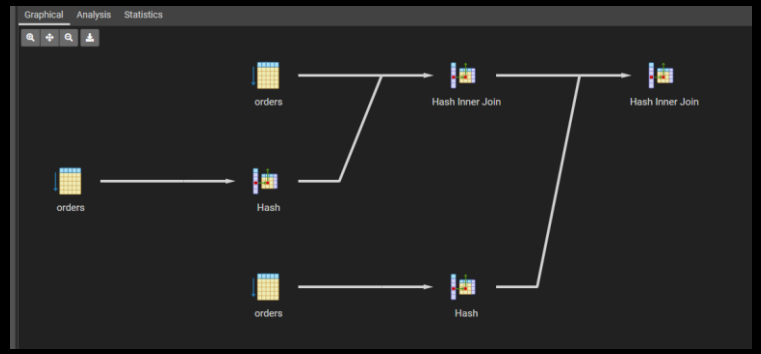
id	value	Loc
1	100	FIN
2	100	SWE

id	value	Loc
1	100	FIN

id	value	Loc
2	100	SWE

APPLICATION PERFORMANCE

- Most performance problems come from poorly coded SQL or programs
  - Either code was bad to begin with or changes have impacted the performance
- DBMS tries to optimize SQL queries
  - Optimizers are not perfect
- Test the database properly
  - Test database vs. Production database
- Use query analyzer to find out problems in your SQL queries



DATA INTEGRITY

- Structural integrity (database) and semantic integrity (data)
  - Structural faults vs. meaning of data
- Managing integrity issues is essential
  - For structural integrity
    - Use DBMS utility programs
    - Use external tools
  - For semantic integrity
    - Use integrity and data constraints
    - Use triggers, functions, and procedures
  - Referential integrity is part of semantic integrity (managing relationship between data)
    - Primary / foreign keys, triggers

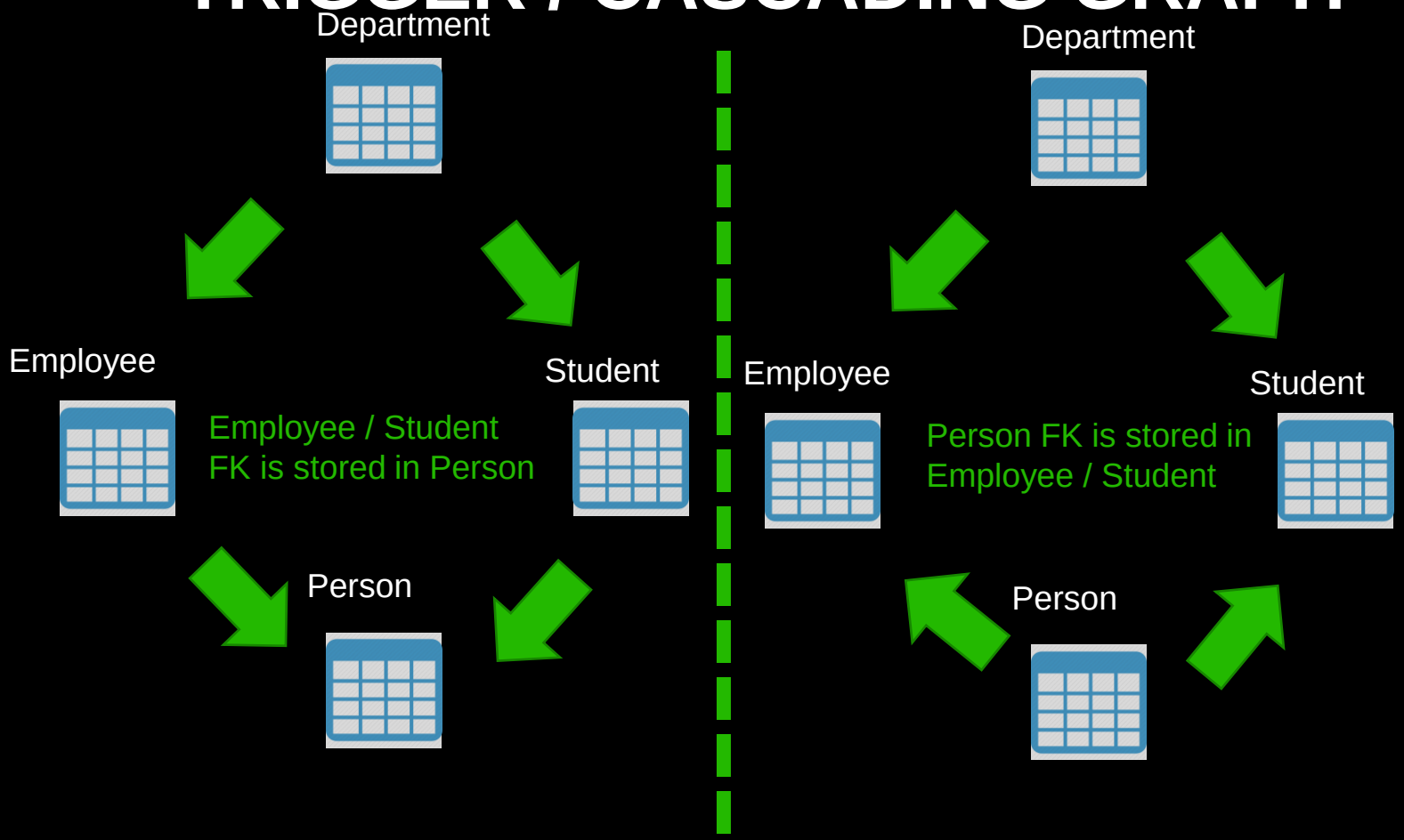
```
CREATE DOMAIN contact_name AS
   VARCHAR NOT NULL CHECK (value !~ '\s');
```

```
CHECK (relationship IN ('married', 'single', 'unknown')
```

PROCEDURES, FUNCTIONS AND TRIGGERS (IN POSTGRESQL

- Procedures are user-defined operations that can be manually called
  - Can give arguments to procedures
  - Can use transactions
- Functions are like procedures but also:
  - Can be nested in SQL queries
  - Can return values
  - Cannot use transactions
- Triggers are user-defined operations that are automatically run
  - Tied to database modifications

TRIGGER / CASCADING GRAPH



What if a person graduates and becomes employee?

What if a person is an employee but starts new studies?

Graduating and becoming a doctoral student?

DATABASE SECURITY

- Data breaches can be accidental or intentional
- Database security can be enhanced with proper administration
  - User authentication and authorization, encryption
- DBMS offer user privilige management
  - Grant, revoke, with grant option
  - Roles, groups
  - Specify database, table, column and row access rights
  - Specify ability to use DBMS commands or executing programs / procedures
- Label based access control can be used to further enhance security

REGULATORY COMPLIANCE

- There are various standards and laws that need to be followed when managing data
  - GDPR, HIPAA, GLB, BASEL, PCI-DSS
  - Depends on where, how and what data is collected / used

- Organizations have their own regulations that need to be followed as well
  - Legal compliance > organization compliance
  - International laws > national laws > foreign laws (in general)

DATABASE BACKUP

- Prepare for problems by building a backup plan
  - Instance, application, transaction, media failures
  - How, when, what to backup
- Different backup types
  - Image copies vs. Logical copies
  - Full vs. Incremental backup
  - Hot vs. Cold backup

DATABASE RECOVERY

- Create a recovery plan, test it regularly, and keep it up-to-date
- Determine recovery options based on the failure, backups and recovery needs
- Identify failure → Analyze situation → What to recover → Locate backups → Restore
- Different types of recovery
  - Recovery to current
  - Point-in-time (partial) recovery
  - Transaction recovery
- Alternatives to backup & recovery
  - Standby databases, replication, and disk mirroring

DISASTER PLAN

- Disaster plan is an extension of recovery plan
  - Specifies incidents that can be classified as ”disasters”
  - Specifies recovery protocol in the case of a disaster
  - Dictates priorities for connecting appliucations
    - Priorities influence what data needs to be recovered first
- Disasters affect all enterprise operations, not just the database
- Disaster plan needs to be in writing and distributed to all key personnel
- Plan should be tested when larger changes happen

BIG DATA MANAGEMENT

- Big data is characterized by multiple V's and comes in different formats
  - Volume, variety, velocity
  - Structured, semistructured, unstructed
- Used in recognizing patterns, analyzing users/products/services as well as machine learning
- Often require a NoSQL database
- Major challenges using big data are:
  - Large amount of data
  - Problems with quality
  - Data integration and preparation
  - System scaling


LUT
University