LUT University

LAND OF THE CURIOUS

TABLE OF CONTENTS

- Application design
  - What you need to know
  - Infrastructure
  - Transactions and locks
- Database change management
  - Need for change
  - Difficulties
  - Change types
- Data availability

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

APPLICATION DESIGN

Lecture

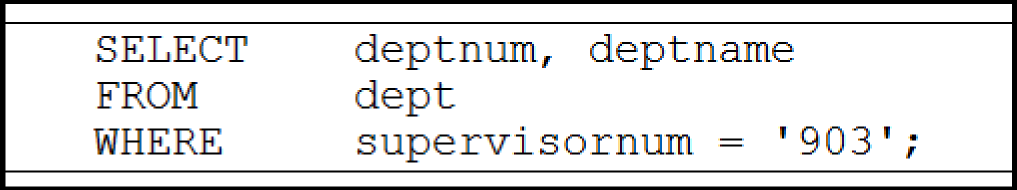
Jiri Musto, D.Sc.

DB APPLICATION DEVELOPMENT AND SQL

- To design an application that relies on databases, you need to understand:
- How data is stored
- How to code SQL statements
- How to embed SQL statements into host programming language
- How to optimize database access
- Programming methods to avoid potential database processing problems

SQL IN SHORT

- Structured Query Language, used for accessing relational databases
- English-like syntax
- SQL operations are performed on table(s) and result in another table



```
SELECT    deptnum, deptname
FROM      dept
WHERE     supervisornum = '903';
```

APPLICATION INFRASTRUCTURE

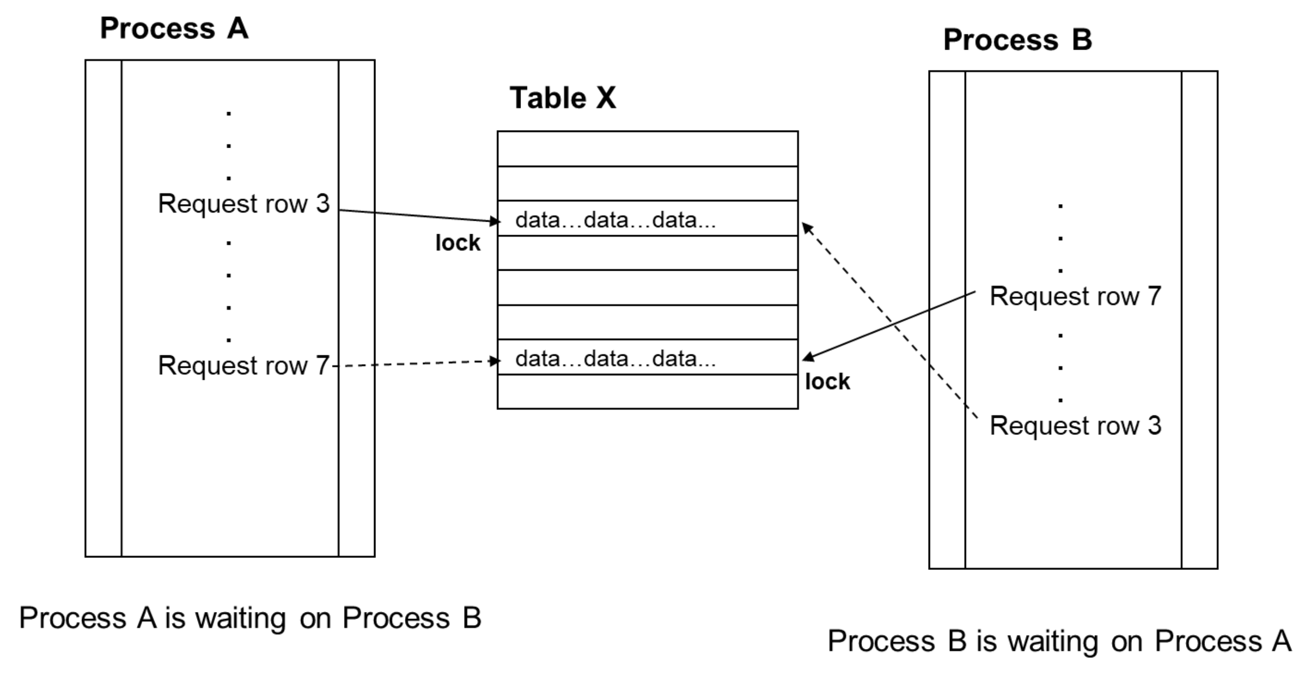
- The combined hardware and software environment
- Vary from organization to organization
- Different programming languages can be used to connect to database
  - They may have their own libraries, adapters, addons, plugins etc.
  - Each application may require a different interface tool to connect to the database
  - Same database can be used for multiple applications

TRANSACTIONS AND LOCKING

- DBMS uses locking to enable concurrent users
- Different levels of locking granularity
  - Row, table, database, etc.
- Different types of locks exist
  - Shared lock
  - Exclusive lock
  - Update lock
  - Intent lock
- Locks may restrict access to certain data until the lock is opened

DEADLOCKS AND TIMEOUTS

- Concurrent users inevitably lead to users accessing the same data
- If another user reaches the data first
  - The other users are put on hold
  - Timeout may occur
- If users require the same data at the same time
  - Deadlock occurs
- Minimze lock duration by having data modification close to the end



LOCK ESCALATION

- Increasing lock granularity for the process
  - Row → Table → Database
- Controlled by system parameters
- For example
  - If one process has multiple row locks in a table, it can be escalated to a table lock
  - Can cause concurrency issues

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

DATABASE CHANGE MANAGEMENT

Lecture

Jiri Musto, D.Sc.

NEED FOR CHANGE

- Physical environment changes
  - Employee amount changes, different skill sets
- Organization changes
  - Processes and methodology are changed, faster product / service delivery
- Network changes
  - New geographic locations, growing network
- Application & system changes
  - More data, different types of data, different processes with existing data
- Data changes
  - Type and structure of data changes

CHANGE MANAGEMENT REQUIREMENTS

- Proactivity
- Intelligence
- Planning analysis
- Impact analysis
- Automation
- Standardization
- Availability

DIFFICULTY OF DATABASE CHANGE

- Many types of changes that are not all supported by DBMS
  - DBMS Software
  - Hardware Configuration
  - Logical and Physical Design
  - Applications
  - Physical Database Structures
- Most organizations have at least two copies of each database
  - Single change need to be done on multiple databases

DBMS SOFTWARE AND HARDWARE CONFIGURATION

- Migration to new DBMS versions and releases
  - New features and functions
  - Deprecated functions and features
- DBMS may require hardware upgrades or configuration changes
- Work in conjunction with the system admins responsible for maintaining the hardware
- When hardware changes, DBMS configuration may have to change

LOGICAL AND PHYSICAL DATABASE DESIGN

- When database changes, blueprints have to change as well
- Changes made at logical level
  - Migrate through the physical level and then the database
- Changes made to database
  - Need to be fitted to the models
- Data modelling tools are helpful

APPLICATIONS

- Application changes need to be synchronized with database changes
- If application changes
  - Database needs to me modified with new data types, amount of data etc.
- If database changes
  - Application needs to be changed to have access to the new database
- If changes are rolled back on one component, the other should follow

PHYSICAL DATABASE STRUCTURES

- The most complicated change
- Requires planning
- ... Analyzing
- ... Implementing
- Not every aspect of database object can be changed using ALTER command
  - Some require dropping and re-creating the object
  - Differs between DBMS

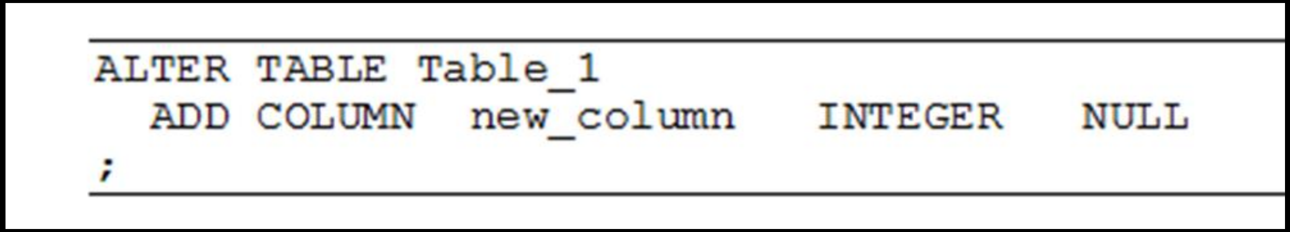
LUT University

LIMITATIONS OF ALTER: NOT LIKELY SUPPORTED

- Moving an object to another database
- Changing the number of table space partition
- Moving a table from one table space to another
- Rearranging the order of columns
- Removing columns from a table
- Changing primary / foreign keys
- Modifying a view
- Etc.

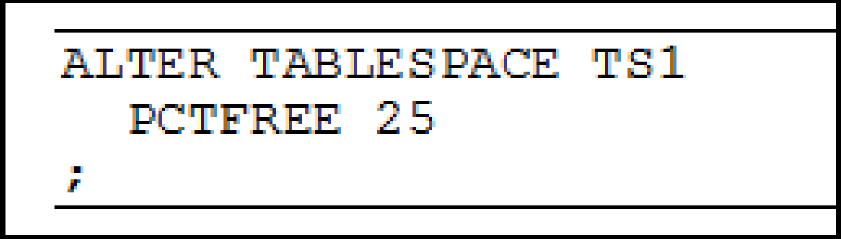
SIMPLE AND MORE DIFFICULT DATABASE CHANGE

- Adding a new column to the end of a table



```
ALTER TABLE Table_1
  ADD COLUMN  new_column    INTEGER    NULL
;
```

- Modifying free space for a table space
  - Requires reorganization after the modification



```
ALTER TABLESPACE TS1
  PCTFREE 25
;
```

ADDING COLUMN TO MIDDLE OF A TABLE

1. Retrieve the definitions of
   1. Current table, and views, indices, triggers using the table
2. Check the referential constraints of the table
3. List all programs accessing the table
4. Unload data from the table and drop it OR use it for step 6 before dropping
5. Re-create the table adding the new column
6. Upload data to the new table
7. Re-create
   1. Constraints, triggers, views, indices, security authorizations
8. Examine all applications if changes are required to function properly

CHANGE MANAGEMENT TOOLS

- Reduce the amount of time required to specify what needs to change
- Simpler method of analyzing the impact of changes
- Simpler method of creating, altering and dropping database objects
- Track changes over time
- Reducing the time to perform changes
- Compare database structures

REQUESTING CHANGES

- Database changes are not done immediately
  - Not reasonable to expect it
- Requests should be reviewed by the database administrator
  - May need to change the requests
- Establish a standardized form for implementing changes
  - Include all necessary information related to each change
- Have a checklist to follow for different types of changes
  - An easy walkthrough to check everything is done correctly and works properly
- Comply with regulations
  - Track who makes what changes

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

DATA AVAILABILITY

Lecture

Jiri Musto, D.Sc.

WHY AVAILABILITY?

- Internet and globalization
  - Prime time is always somewhere
- Customers demand service now
- Competitive advantage
- Need to react to competitors

DEFINING AVAILABILITY

- A given resource can be accessed by consumers
- If database is available, data users can access it
- Availability can be measures
  - Percentage of time system can be used
- Database availability and database performance are often confused
  - Poor performance database can still be accessed but unavailable databases cannot be accessed
- If performance suffers so that users cannot perform their job, database is unavailable

COMPONENTS OF AVAILABILITY

- Manageability—the ability to create and maintain an effective environment that delivers service to users
- Recoverability—the ability to re-establish service in the event of an error or component failure
- Reliability—the ability to deliver service at specified levels for a stated period
- Serviceability—the ability to determine the existence of problems, diagnose their cause(s), and repair the problems.

AVAILABILITY REQUIREMENTS

- The shrinking maintenance window
  - More data, 24/7 availability requirement
  - Avoid downtime during traditional maintenance tasks
- Decision support, business intelligence, and analytics
  - Core business requirements in the modern world
- Data warehousing
  - Critical in many organizations, different performance needs but similar availability needs
- Full-time availability
- Increasing IT complexity

COST OF DOWNTIME

- Lost business during outage
- Cost of catching up
- Legal costs
- Impact of reduced stock value
- Impact to company’s image

- All of these add up to huge amounts of money

HOW MUCH AVAILABILITY IS ENOUGH

- Not all require 100 % availability
- What about 99.999 %?
  - 5 minutes per year
- Build a business case
- Calculate the cost of downtime
  - Varies by industry and application
  - Cost of assuring availability vs. Cost of downtime
- Service level agreements

AVAILABILITY VS DOWNTIME

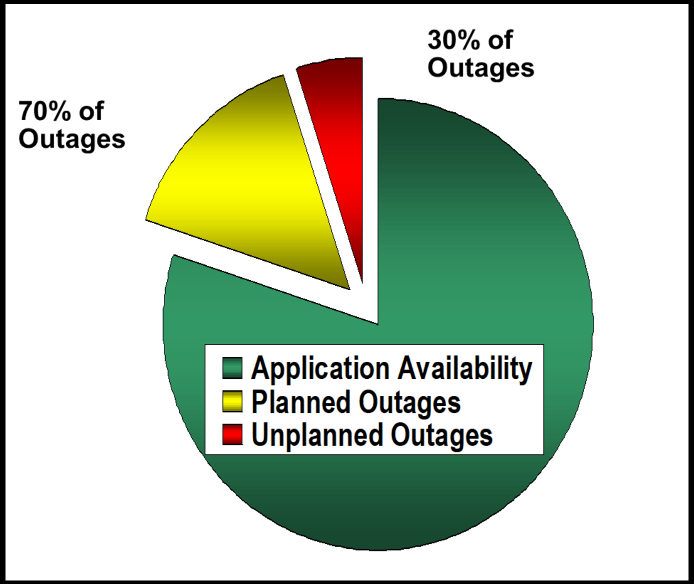
Availability	Approximate downtime per year	
	In minutes	In hours
99.999%	5 minutes	.08 hours
99.99%	53 minutes	.88 hours
99.95%	262 minutes	4.37 hours
99.9%	526 minutes	8.77 hours
99.8%	1,052 minutes	17.5 hours
99.5%	2,628 minutes	43.8 hours
99%	5,256 minutes	87.6 hours
98%	10,512 minutes	175.2 hours (or 7.3 days)

CAUSES OF AVAILABILITY PROBLEMS

Data related	Software related	Hardware related
Corruption of data	Operating system failure	Loss of data center
Loss of data	DBMS failure	Loss of server hardware
Loss of database objects	Application problems	Disk outage
Data replication failures	Performance problems	Network problems
Security and authorization problems		
Recovery issues		

PLANNED VS UNPLANNED OUTAGES

- Unplanned outages - driven by problems
  - Element failure
  - Performance degradation
  - Capacity limitation
  - Application logic error
  - Transaction backout
  - Data corruption
- Planned outages - driven by change
  - Database maintenance
  - Application migrations
  - Configuration upgrade
  - Data propagation



IMPROVING AVAILABILITY

- Perform routine maintenance
- Automate database administration functions
- Exploit high availability features
  - Parallelism
  - Clustering, partitioning
- Exploit hardware technologies
  - For example RAID (Redundant Array of Independent Disks)

LUT
University