AALBORG
UNIVERSITY

Database System (SW5)

1. Introduction

Tiantian Liu
Department of Computer Science
Aalborg University
Fall 2025

Agenda

- Course Organization
- Course Outline
- Introduction to DBMS

AALBORG UNIVERSITY

Lecturer

AALBORG UNIVERSITY

- **Tiantian Liu**
- Assistant Professor, Aalborg University
- Expertise:
  - Graph search
  - Spatial database
- Email: liutt@cs.aau.dk
- Office: 2.2.090, in ACM15 Copenhagen

Administrative Information

AALBORG UNIVERSITY

Course website

- Moodle:

  https://www.moodle.aau.dk/course/view.php?id=56751

All course materials on Moodle

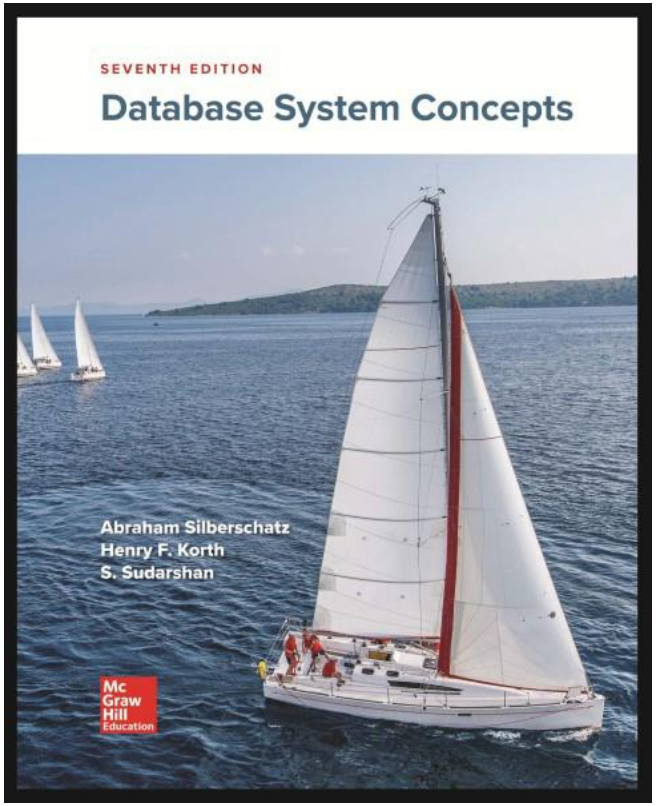
(slides, exercise sheets, . . .)

- News forum

Book

AALBORG UNIVERSITY

- Silberschatz, Korth, Sudarshan
  - 7th edition
  - Relational model and algebra
  - SQL
  - Database Design
  - ER Model
  - Query processing and Optimization
  - Transactions
  - . . .



Course Setup

AALBORG UNIVERSITY

- Slides
  - Slides will be uploaded to the Moodle one day before the lecture
  - Slides override book (if the notations are different from the book, please refer to the slides)
- Exercises
  - After lectures
  - Deepen understanding the material covered in today's lecture
  - Discussions + training for the written exam
  - Mind the notations
  - Solutions handed out after exercise sessions

Course Setup

AALBORG UNIVERSITY

- Exam
  - Written exam
  - 2 hours
  - 100 points

Nature and scope	Written teaching material	Other texts written by others	Own notes	Audio and visual material (offline)	Audio and visual material (online)	Moodle	Offline software	Online software
Unlimited	X		X			X		
Partially allowed (in this case, it must be described exactly how)							PostgreSQL	
Not allowed		X		X	X			X

Exam Aids

Course Setup

AALBORG UNIVERSITY

- Expected Time Usage
  - 1 ECTS corresponds to ca. 27.5 hours
  - 5 ECTS = 137.5 hours
    - Lectures and exercises: ca. 44 hours
    - Home readings: ca. 54 hours
    - Exam preparation: ca. 39.5 hours

Course Setup

AALBORG UNIVERSITY

- Acknowledgements
  - Contributors to exams, slides, etc.
    - Yongluan Zhou
    - Jilin Hu
    - Hua Lu
    - Katja Hose
    - Kristian Torp
    - …
  - Teaching material provided by books and their authors
    - Database System Concepts by A. Silberschatz, H. Korth, and S. Sudarshan
    - …

Learning Goals

AALBORG UNIVERSITY

- This is NOT a course that …
  - Trains you to be a database administrator within 3 months
  - Makes you an expert on Oracle, MySQL, DB2, Postgres, or any other commercial/open-source database management system
- This is a course that. . .
  - Teaches you the fundamentals that are common to all such systems
  - Helps you make (better) use of relational database systems

Learning Goals

AALBORG UNIVERSITY

- Understand the relational model and apply relational algebra
- Conceptually design a database (ER model, conceptual design)
- Create and evaluate a database schema that adheres to normal forms (logical design)
- Make use of SQL to create, modify, and query relational databases
- Understand how the data is physically organized and apply an appropriate physical design
- Interpret a query execution plan and assess whether a plan is effective
- Understand the transactional concept and key issues of concurrency control and recovery

Tips from Past Students

AALBORG UNIVERSITY

- Do the exercises after the class, it helps a lot with understanding the topics.
- Attend lectures, do the in-lecture and post-lecture exercises with your group.
- Write the exercises properly down in a notebook, so you don’t lose them and have notes for the exercises for later.

Agenda

- Course Organization
- Course Outline
- Introduction to DBMS

Overview

AALBORG UNIVERSITY

Date	Time	Topic	Location
Sep. 24	10:00 – 11:30	Introduction	0.091
Sep. 24	12:30 – 16:15	Relational Model	0.091
Oct. 1	8:15 – 12:00	SQL 1	3.084b
Oct. 1	12:30 – 16:15	SQL 2	2.0.091
Oct. 8	8:15 – 12:00	ER Model 1	0.091
Oct. 8	12:30 – 16:15	ER Model 2	0.091
Oct. 15	8:15 – 12:00	Relational Database Design	2.0.091
Oct. 15	12:30 – 16:15	Physical Design	2.0.091
Oct. 29	8:15 – 12:00	Query Processing and Optimization	3.084b
Oct. 29	12:30 – 16:15	Transaction and Concurrency Control	2.0.091
Nov. 5	12:30 – 16:15	Recovery and Summary	0.091

Relational Model

AALBORG UNIVERSITY

- Motivation
  - SQL's data model resembles the relational model
  - SQL is partly based on relational algebra
  - Relational algebra helps understanding query plans and query optimization
- Learning Goals
  - Explain the relational model
  - Create non-trivial relational algebra queries
  - Use the various join types in relational algebra queries

SQL

AALBORG UNIVERSITY

- Motivation
  - SQL is the query language
  - SQL is very widely used
  - SQL is well supported by relational database systems
- Learning Goals
  - Explain and use the SQL data model
  - Create non-trivial database tables
  - Modify non-trivial database tables
  - Create non-trivial SQL statements

The Entity-Relationship Model

AALBORG UNIVERSITY

- Motivation
  - ER diagrams are widely used
  - ER model is easy to learn--Much simpler than UML
  - An ER diagram is a good communication tool--Talking the same language
- Learning Goals
  - Create non-trivial ER diagrams
  - Analyze if an ER diagram is good or bad
  - Create and explain the mapping of ER diagrams to relations
  - Use a particular ER notation properly

Relational Database Design

AALBORG UNIVERSITY

- Motivation
  - A good design prevents problems that occur later during usage
  - Changing a database schema before it is in use is much cheaper
  - Redundancy is the source of many problems and should be avoided by meeting certain design criteria
- Learning Goals
  - Understand the concepts of functional dependencies and normal forms
  - Describe the quality of a design by using normal forms
  - Improve a database design by decomposition of relations

Physical Design

AALBORG UNIVERSITY

- Motivation
  - Physical design directly impacts the performance of database operations.
  - Faster databases lead to better user experiences, lower operational costs, and the ability to handle larger datasets effectively.
- Learning Goals
  - Understand how tables are stored in files
  - Understand basic indexing techniques
  - Understand the effects of storage and indexes on the performance of basic SQL queries

Query Execution and Optimization

AALBORG UNIVERSITY

- Motivation
  - Understanding the basics of query processing and query optimization is the foundation of database tuning
- Learning Goals
  - Understand how selection statements are executed
  - Understand the basic join algorithms
  - Understand the basics of heuristic (logical) query optimization
  - Understand the basics of physical query optimization

Transactions and Concurrency Control

AALBORG UNIVERSITY

- Motivation
  - Transaction boundaries are an important part of system design
  - Users think in transactions
  - Simple to handle multiple users
- Learning Goals
  - Understanding the transaction concept
  - Understanding serializability
  - Understand and use lock-based concurrency control
  - Understand and use two-phase locking

Recovery System

AALBORG UNIVERSITY

- Motivation
  - We want to preserve consistency and availability even in the case of failures.
- Learning Goals
  - Understanding basic logging algorithms
  - Understanding the importance of atomicity and durability

Agenda

- Course Organization
- Course Outline
- Introduction to DBMS

Database Management Systems

AALBORG UNIVERSITY

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both convenient and efficient to use
- Database systems are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple users and applications, often at the same time

Database Application Example

AALBORG UNIVERSITY

- Enterprise Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.

Source: Plopdo

Database Application Example

AALBORG UNIVERSITY

- Banking and finance
  - Customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data

Source: loyensloeff

Database Application Example

AALBORG UNIVERSITY

- Universities
  - Student information
  - Faculty details
  - Courses
  - Departments
  - Enrollments
  - Grades
  - …

Problems with Data

AALBORG UNIVERSITY

- Student (id, first name, last name, birthdate, department, ECTS)
- Student timPedersen = new Student(1235, “tim", “pedersen", “32.4.2004", “computer science", 15);
- Student tomJensen = new Student(1235, “tom", “jensen", “11.3.2017", “Dept. of Computer Science", -20);

Is there a potential problem?

Problems with Data

AALBORG UNIVERSITY

- Student (id, first name, last name, birthdate, department, ECTS)
- Student timPedersen = new Student(1235, “tim", “pedersen", “32.4.2004", “computer science", 15);
- Student tomJensen = new Student(1235, “tom", “jensen", “11.3.2017", “Dept. of Computer Science", -20);
- Potential Problems
  - Both students have the same ID 1235
  - 32.4.2004 no valid date
  - 11.3.2017 rather young for a student
  - Is "computer science" and "Dept. of Computer Science" the same?
  - Is "-20" a valid number of ECTS?

Problems with Data

AALBORG UNIVERSITY

- Student timPedersen = new Student(1235, “tim", “pedersen", “32.4.2004", “computer science", 15);
- Student tomJensen = new Student(1235, “tom", “jensen", “11.3.2017", “Dept. of Computer Science", -20);

…, 10000 more students are added

Problems with Data

AALBORG UNIVERSITY

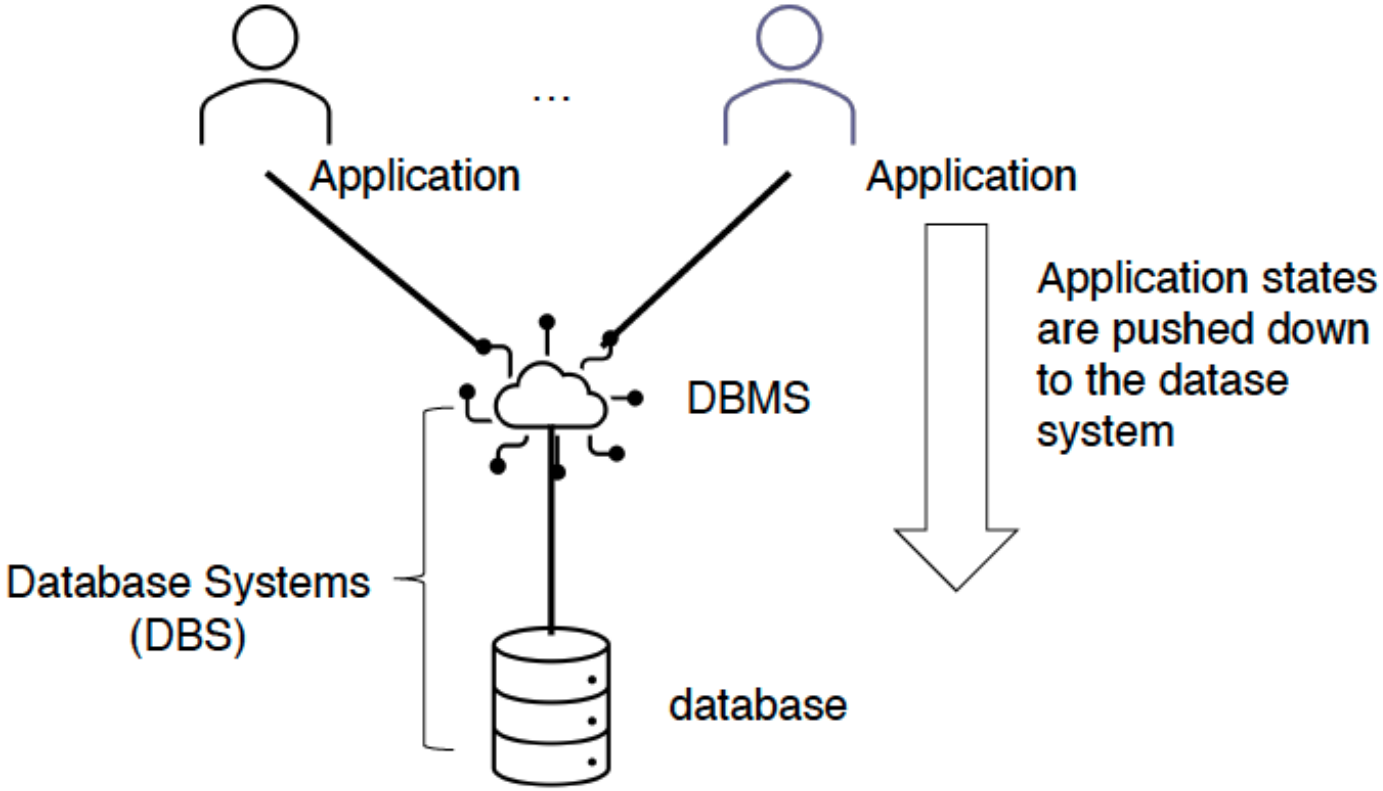
- Student timPedersen = new Student(1235, “tim", “pedersen", “32.4.2004", “computer science", 15);
- Student tomJensen = new Student(1235, “tom", “jensen", “11.3.2017", “Dept. of Computer Science", -20);

…, 10000 more students are added

- Potential Problems
  - Not enough main memory for all the data
  - When data is updated in main memory, the hard disk also needs to be updated.
  - Efficient query processing (search, sorting,. . .)
  - Robust sophisticated query language
  - Access rights
  - Storing additional information

Database Systems

AALBORG UNIVERSITY



University Database Example

AALBORG UNIVERSITY

- Data consists of information about:
  - Students
  - Instructors
  - Courses
- Application program examples:
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts

Purpose of Database Systems

AALBORG UNIVERSITY

- In early days, data-intensive applications were built directly on top of file systems, which leads to:
  - Data redundancy and inconsistency
    - Data is stored in multiple file formats resulting in duplication of information in different files
  - Data isolation
    - Multiple files and formats
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task, e.g.,
      - Joining information about students and course registrations
      - Finding students that failed a particular course
  - Integrity problems
    - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

Purpose of Database Systems

AALBORG UNIVERSITY

- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Ex: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- Security problems
  - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

View of Data

AALBORG UNIVERSITY

- A database system is a collection of interrelated data and a set of programs that allow users to access and modify these data.
- A major purpose of a database system is to provide users with an abstract view of the data.
  - Data models
    - A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
  - Data abstraction
    - Hide the complexity of data structures to represent data in the database from users through several levels of data abstraction.

Data Models

AALBORG UNIVERSITY

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-structured data model (XML)
- Graph data model
- Other older models:
  - Network model
  - Hierarchical model

Relational Models

AALBORG UNIVERSITY

- All the data is stored in various tables.
- Example of tabular data in the relational model

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

A Sample Relational Database

AALBORG UNIVERSITY

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Levels of Abstraction

AALBORG UNIVERSITY

- **Physical level**: describes how a record (e.g., instructor) is stored.
- **Logical level**: describes data stored in database, and the relationships among the data.

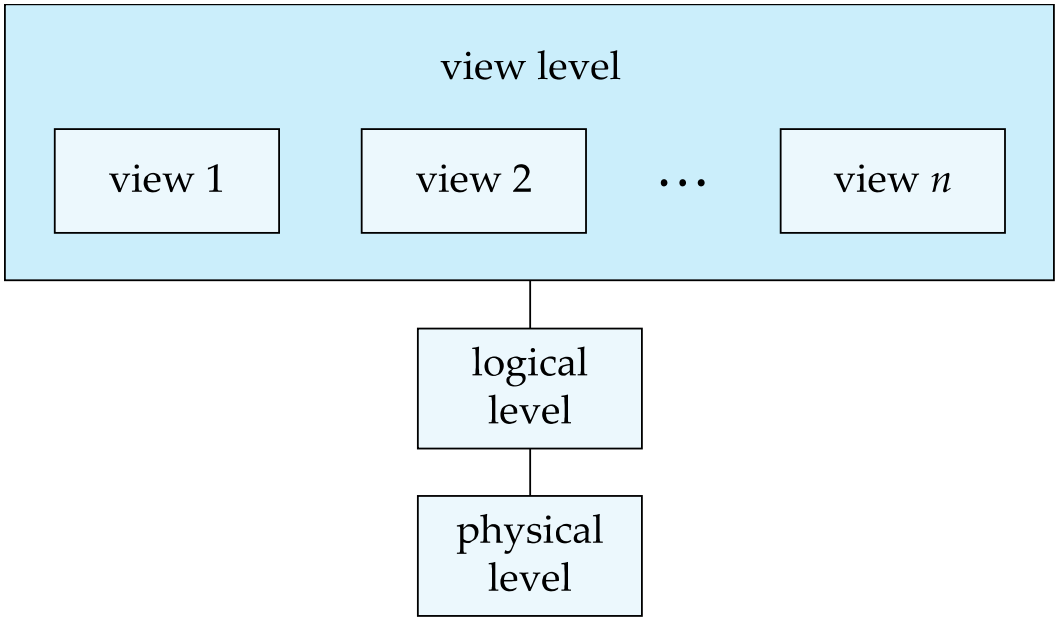
```
type instructor = record
    ID: string;
    name: string;
    dept_name: string;
    salary: integer;
        end;
```

- **View level**: application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Levels of Abstraction

AALBORG UNIVERSITY

- An architecture for a database system



Instances and Schemas

AALBORG UNIVERSITY

- **Logical Schema** – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
    - Analogous to type information of a variable in a program
- **Physical schema** – the overall physical structure of the database
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable

Data Definition Language (DDL)

AALBORG UNIVERSITY

- Specification notation for defining the database schema

  Example: **create table** *instructor (*

  ID **char**(5)*,*

  name **varchar**(20),

  dept_name **varchar**(20),

  salary **numeric**(8,2));

- DDL compiler generates a set of table templates stored in a data dictionary
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

Data Manipulation Language (DML)

AALBORG UNIVERSITY

- Language for accessing and updating the data organized by the appropriate data model
  - DML also known as query language
- There are basically two types of data-manipulation language
  - Procedural DML -- require a user to specify what data are needed and how to get those data.
  - Declarative DML -- require a user to specify what data are needed without specifying how to get those data.

SQL Query Language

AALBORG UNIVERSITY

- SQL query language is nonprocedural. A query takes as input several tables (possibly only one) and always returns a single table.
- Example to find all instructors in Comp. Sci. dept

```
select name
from instructor
where dept_name = 'Comp. Sci.';
```

- To be able to compute complex functions SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Database Design

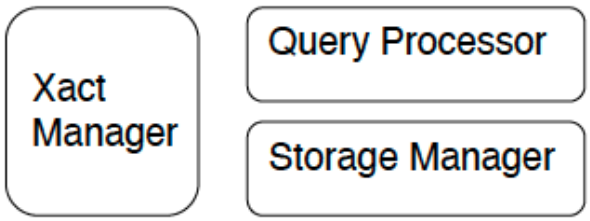
AALBORG UNIVERSITY

- The process of designing the general structure of the database:
  - Logical Design – Deciding on the database schema. Database design requires that we find a “good” collection of relation schemas.
    - Business decision – What attributes should we record in the database?
    - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
  - Physical Design – Deciding on the physical layout of the database

Database Engine

AALBORG UNIVERSITY

- A database system is partitioned into modules that deal with each of the responsibilities of the overall system
- The functional components of a database system can be divided into
  - The storage manager
  - The query processor component
  - The transaction management component



History of Database Systems

AALBORG UNIVERSITY

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provided only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - Won the ACM Turing Award for this work

History of Database Systems

AALBORG UNIVERSITY

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - Wisconsin, IBM, Teradata
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History of Database Systems

AALBORG UNIVERSITY

- 2000s:
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon,
    - “NoSQL” systems.
  - Big data analysis: beyond SQL
    - Map reduce and friends
- 2010s:
  - SQL reloaded
    - SQL front end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases

Popular Database Systems

AALBORG UNIVERSITY

- The big players
  - Oracle
  - Microsoft SQL server
  - …
- The “smaller” players
  - PostgreSQL
  - MySQL
  - …

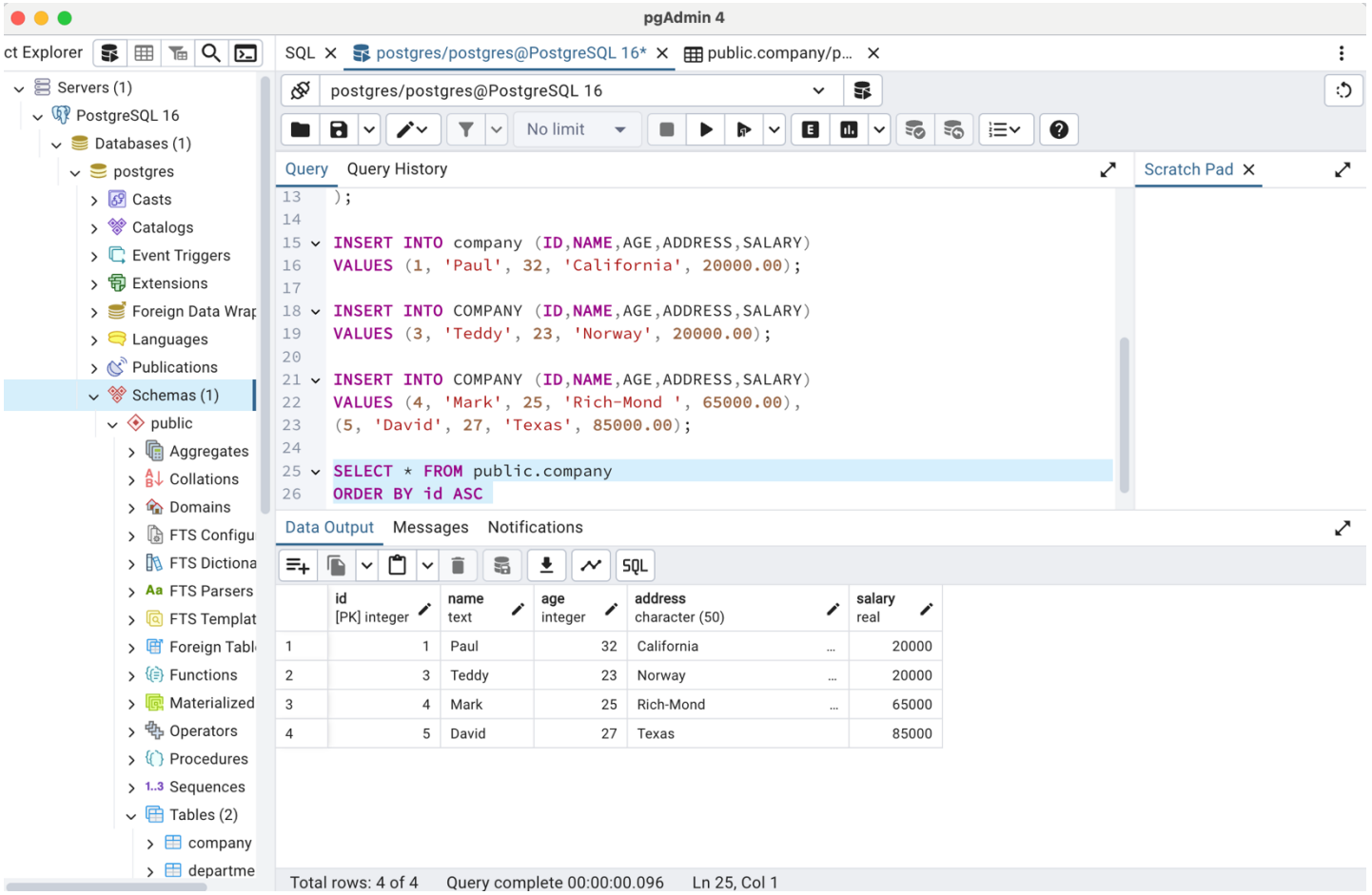
PostgreSQL

AALBORG UNIVERSITY

- The world's most advanced open-source database
- Easy to install
- Good SQL support
- Transactions
- Good Documentation

- Try to install by yourself
  - https://www.enterprisedb.com/downloads/postgres-postgresql-downloads
  - Shttps://www.postgresql.org/docs/current/index.html

PostgreSQL


pgAdmin 4
Servers (1)
PostgreSQL 16
Databases (1)
postgres
Schemas (1)
public
Tables (2)
company
postgres/postgres@PostgreSQL 16
No limit
Query
Query History
Scratch Pad
INSERT INTO company (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (1, 'Paul', 32, 'California', 20000.00);
INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (3, 'Teddy', 23, 'Norway', 20000.00);
INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00),
(5, 'David', 27, 'Texas', 85000.00);
SELECT * FROM public.company
ORDER BY id ASC
Data Output
Messages
Notifications
id [PK] integer
name text
age integer
address character (50)
salary real
1 1 Paul 32 California 20000
2 3 Teddy 23 Norway 20000
3 4 Mark 25 Rich-Mond 65000
4 5 David 27 Texas 85000
Total rows: 4 of 4
Query complete 00:00:00.096
Ln 25, Col 1

Summary

AALBORG UNIVERSITY

- What is the DBMS
- Why do we need a DBMS
- How does DBMS work

AALBORG UNIVERSITY

Next Lecture

- Relational Model (Chap 2)
  - Explain the relational model
  - Create non-trivial relational algebra queries
  - Use the various join types in relational algebra queries

Before Next Wednesday

AALBORG UNIVERSITY

- Install PostgreSQL
- Get familiar with PostgreSQL
- Create two databases (copy, paste, and run the provided code)
  - One for exercise session
  - One for in-class exercise (DDL and SQL code for creating small relations)
- Relevant materials are available under Lecture 3 on Moodle