Web Systems Fundamentals and Databases (Grundläggande webbsystem och databaser) DI4020 - 11hp

Lectures 1 and 2

Wagner Ourique de Morais

Previously in the course

- Course Introduciton
  - Objectives, Literature, TLA, ATs, GC
  - Course Plan and Schedule
- Updated Course Description and Week 4 material!
- Select and inform student groups
  - 65 out of 96 students
  - 25 out of 32 groups
  - 9 incomplete (less than 3 students)
  - IMPORTANT
    - On Thursday, Jan 23rd at 13h, students without a student group and student group(s) with less than 2 students will be grouped randomly.
- Lab Exercises Sessions will be set on Friday, Jan 24th during the lecture.

Week	Where	TLA	TLA	Content
w4	Tue, Jan 21 at 11:15	R4318	Course introduction	ILO, TLAs, Literature, Ats, …
w4	Wed, Jan 22 at 15:15	R4318	Lecture 1	Intro to DB and DBMS area
w4	Fri, Jan 24 at 10:15	R4318	Lecture 2	Intro to DB and DBMS area
w5	Mon, Jan 27 at 15:15	R4318	Lecture 3	DB Design
w5	Wed, Jan 29 at 08:15	R4318	Lecture 4	DB Design
w5	Wed, Jan 29 at 10:15	R3140	Lab 1	Intro to DB design (ER and Rel Model)
w6	Tue, Feb 04 at 13:15	R4318	Lecture 5	Rel DB Design
w6	Wed, Feb 05 at 08:15	R3140	Lecture 6	Rel DB programming (SQL)
w6	Thu, Feb 06 at 08:15	R4318	Lab 2	Rel diagram
w7	Mon, Feb 10 at 13:15	R4318	Lecture 7	Rel DB programming (SQL)
w7	Tue, Feb 11 at 13:15	R4129	Lecture 8	Rel DB programming (SQL)
w7	Wed, Feb 12 at 08:15	R3140	Lab 3	DB programming
w8	Mon, Feb 17 at 10:15	R4129	Lecture 9	Intro Web
w8	Mon, Feb 17 at 13:15	R4318	Lecture 10	HTML
w8	Tue, Feb 18 at 13:15	R3140	Lab 4	DB programming
w9	Mon, Feb 24 at 10:15	R4318	Lecture 11	HTML
w9	Mon, Feb 24 at 13:15	R4318	Lecture 12	CSS
w9	Tue, Feb 25 at 13:15	R4318	Project Intro	
w9	Wed, Feb 26 at 10:15	Zoom	Lab 5	Static web - HTML
w10	Mon, Mar 03 at 13:15	R4318	Lecture 13	CSS
w10	Tue, Mar 04 at 11:15	R4318	Lecture 14	DB Review + JavaScript
w10	Wed, Mar 05 at 08:15	Zoom	Lab 6	Static web - HTML + CSS
w11	Mon, Mar 10 at 13:15	R4318	Lecture 15	JavaScript
w11	Tue, Mar 11 at 13:15	R4318	Lecture 16	Dynamic Web - PHP
w11	Wed, Mar 12 at 08:15	D308	Proj. Sup. 1	
w12	Fri, Mar 21 at 09:00		Exam 2202 Mand. Reg. Prelim.	DB Exam 2202
w13	Mon, Mar 24 at 10:15	R4318	Lecture 17	Dynamic Web - PHP
w13	Tue, Mar 25 at 13:15	R4318	Lecture 18	Dynamic Web - PHP
w13	Wed, Mar 26 at 08:15	Zoom	Lab 7	Dynamic Web - PHP + MySQL
w14	Mon, Mar 31 at 10:15	R4129	Lecture 19	Dynamic Web - PHP + MySQL
w14	Tue, Apr 01 at 11:15	R4129	Lecture 20	PHP Validation, State, Security
w14	Wed, Apr 02 at 08:15	Zoom	Lab 8	JavaScript, jQuery and AJAX
w15	Mon, Apr 07 at 08:15	D308	Proj. Sup. 2	
w15	Wed, Apr 09 at 13:15	Zoom	Lab 9	Dynamic Web - PHP + PostgreSQL
w17	Wed, Apr 23 at 08:15	D308	Proj. Sup. 3	
w19	Thu, Maj 08 at 08:15	D308	Proj. Sup. 4	
w21	Mon, Maj 19 at 10:15	R4129	Lecture 21	Web Review
w21	Wed, Maj 21 at 08:15	D308	Prj Seminar	
w22	Fri, Maj 30 at 09:00		Exam 2104 Mand Reg. Prelim.	Web exam 2104
w33	Mon, Aug 11 at 09:00		Re Exam 2202 Mand Reg. Prelim.	Web exam 2202
w33	Fri, Aug 15 at 09:00		Re Exam 2104 Mand Reg. Prelim.	Web exam 2104

HALMSTAD UNIVERSITY

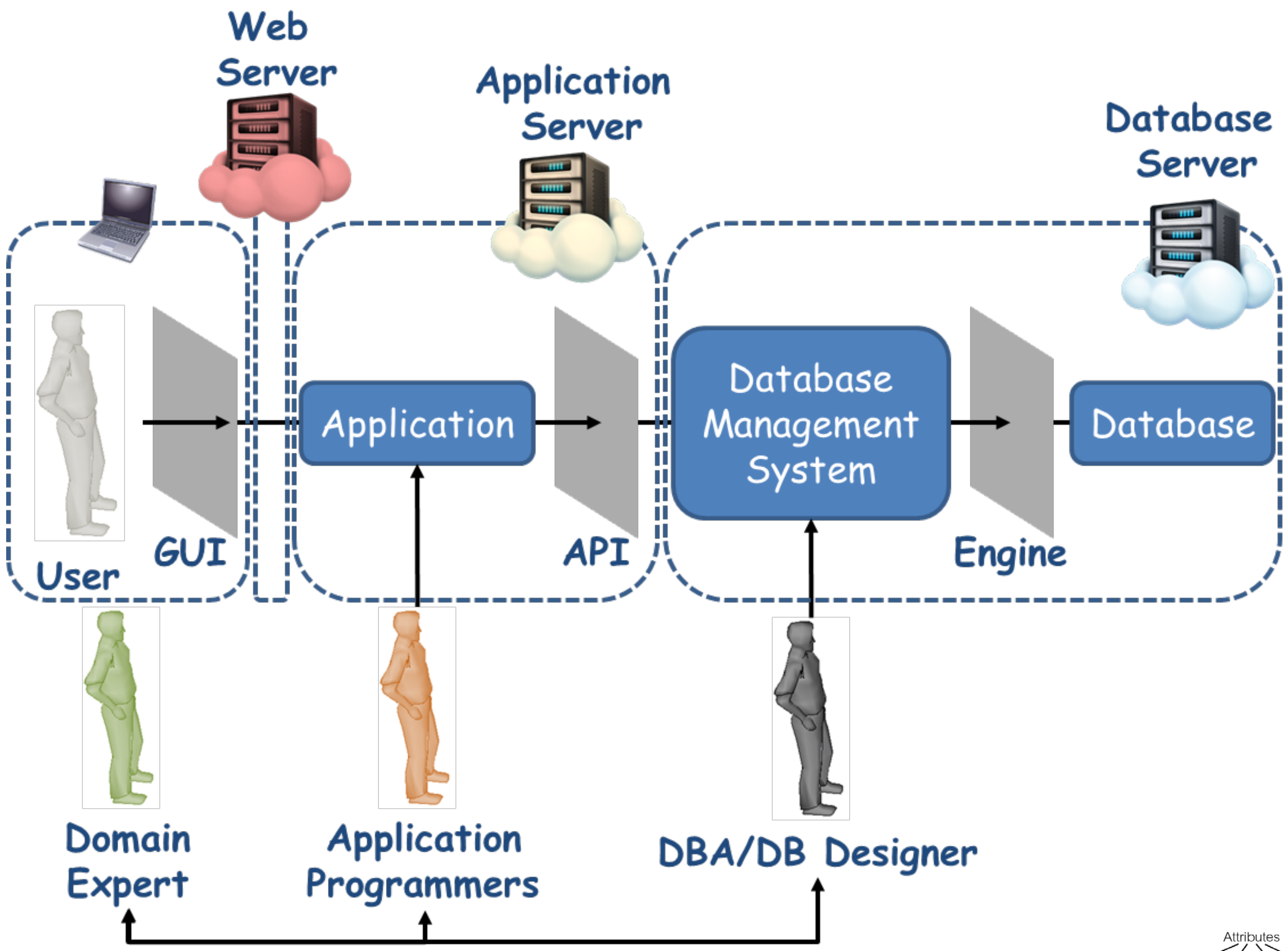
Objectives

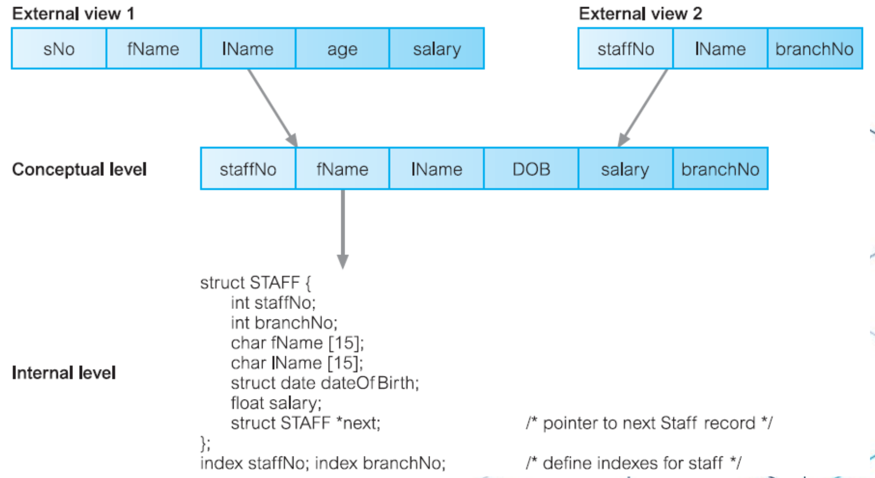
- Introduce and discuss the field of database systems and database design
  - Chapter 1 introduces the field of database management and the advantages offered by databases.
  - Chapter 2 examines the database environment.
  - Chapter 3 examines multi-user DBMS architectures and discusses the different types of middleware that exist in the database field.

…more specifically

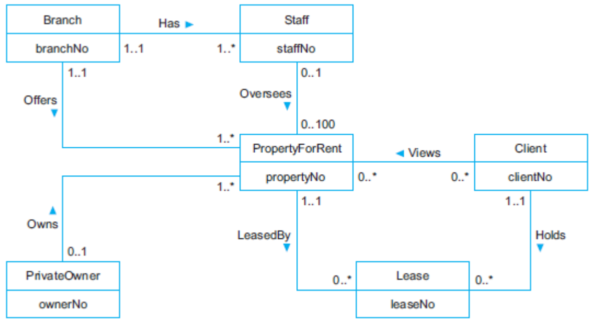
- Present common uses of database systems.
- Introduce the field of database systems.
- Discuss the problems with previous approach (file-based).
- Discuss terminology, e.g., database vs DBMS.
- Brief overview of the history of the development of DBMSs.
- Discuss the purpose, components and functions of a Database Management System (DBMS).
- Discuss the advantages and disadvantages of DBMSs.
- Introduce the concepts of data models and conceptual modelling.
- Discuss the types of language that are used by DBMSs.
- Examine the client-server architecture and differ two-tier, three-tier and n-tier client-server architectures.

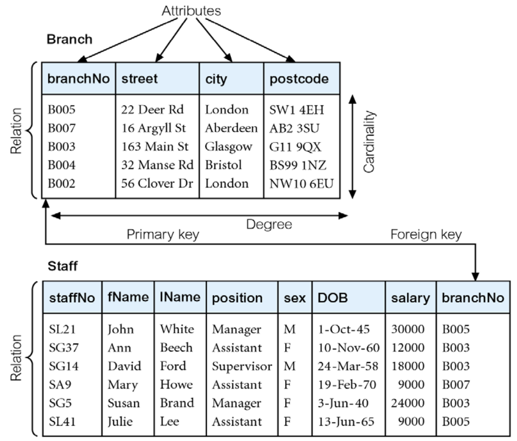
Objectives Illustrated





```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;                /* pointer to next Staff record */
};
index staffNo; index branchNo;         /* define indexes for staff */
```




Attributes
Relation
Cardinality
Degree
Primary key
Foreign key
Relation

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

```
CREATE TABLE Staff(
        staffNo VARCHAR(5) NOT NULL PRIMARY KEY,
        lName VARCHAR(50) NOT NULL,
        salary DECIMAL(7,2));

INSERT INTO Staff VALUES ('SG16', 'Brown', 8300);

SELECT staffNo, lName, salary
FROM Staff
WHERE salary > 10000;
```

HALMSTAD UNIVERSITY

Learning about databases is important because…

- "…today we exist in a world where **everything that can be recorded is recorded** and correcting or deleting all copies of data is difficult if not impossible." [2013 - Lake & Crowther - Concise Guide to Databases - A Practical Introduction]

- "...there is not one aspect of modern business that has avoided the need to **collect, collate, organize and report upon data**..." [Hill (2013). Preface. In Lake et al, Concise Guide to Databases: A Practical Introduction]

- "...**data management** became the key enabling technology for businesses of all sizes worldwide, leading to today's $55B DBMS market and tens of millions of operational databases..." [2015 - Bailis et al - Readings in Database Systems]

- **So, there is a need for correct, secure, efficient and effective storage, management, access and manipulation of data.**

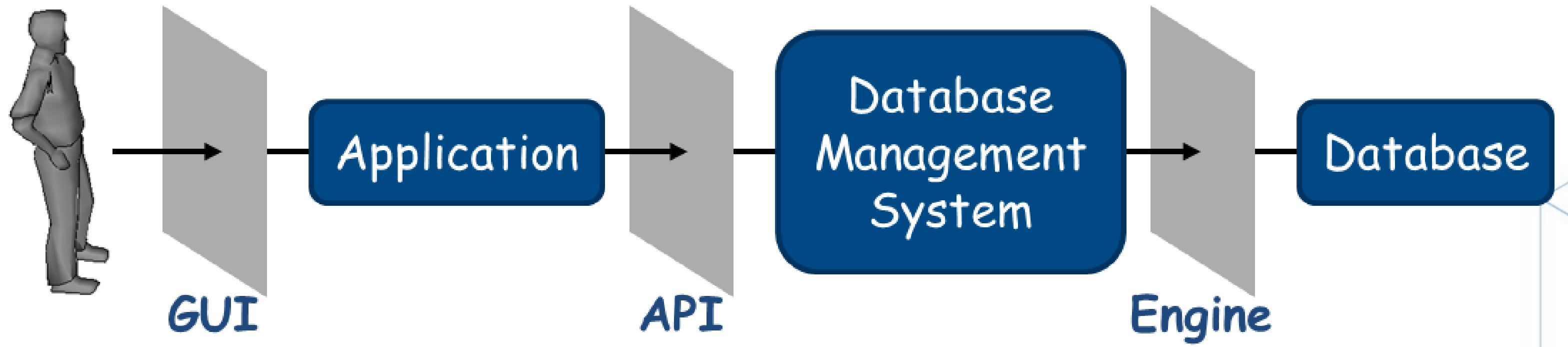
HALMSTAD UNIVERSITY

Common uses of database systems

- "...databases support your daily activities, such as securing your banking and credit card transactions...." [2015 - Bailis et al - Readings in Database Systems]
  - Banking
  - Driving need for the development of DBMS
  - Purchases from the supermarket
  - Booking a holiday at the travel agents
  - Using the local library
  - Renting a video
  - Using the Internet
  - Studying at university
  - Research
  - Biological data (e.g. genome data)
  - Sensor data
  - Geographical data

Chapter 1

Introduction to Databases



HALMSTAD UNIVERSITY

Data

- Raw facts that can be processed to produce information, which in turn enables good decision making and safe operations.
- A single data value does not tell much as a set of data values!
- The value of data as an organizational asset is widely recognized.
- How to manage data?

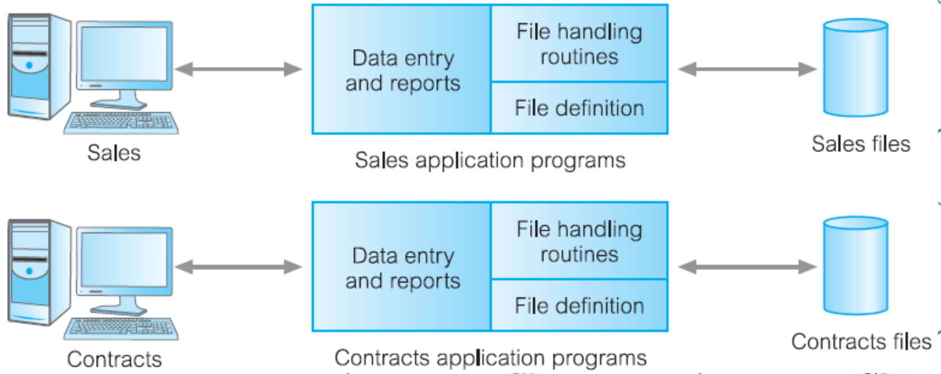
File-Based Systems

```
197911222030, de Morais, Wagner
198011152010, Hakerod, Jesper
198112146545, Ali, Hazem
```

```
197911222030de Morais                        Wagner
198011152010Hakerod                          Jesper
198112146545Ali                              Hazem
```

- Store data into files
  - CSV and spreadsheets
- Each file is a sequence of records
- Each program defines and manages its own data.
  - Parse the files each time they want to read/update records.

```
struct studentRecord{
    char studentId[12];
    char lastName[25];
    char firstName[25];
}; /* 62 bytes*
...
studentRecord sRec[20];
...fread(sRec, 62, 20, file);
/*read 20 student records*/...
```



Conolly & Begg (2015), "Database Systems - A practical approach to design, implementation and management", 2015

Limitations of File-Based Systems

```
struct studentRecord{
    char studentId[12];
    char lastName[25];
    char firstName[25];
}; /* 62 bytes*
...

studentRecord sRec[20];
...
fread(sRec, 62, 20, file); /*read 20 student records*/
...
```

- Separation and isolation of data
  - Each program maintains its own set of data.
  - Users of one program may be unaware of potentially useful data held by other programs.
- Duplication of data
  - Same data is held by different programs.
  - Wasted space and potentially different values and/or different formats for the same item.
- Data dependence
  - File structure is defined in the program code.
- Incompatible file formats
  - Programs are written in different languages, and so cannot easily access each other's files.
- Fixed Queries/Proliferation of application programs
  - Programs are written to satisfy particular functions.
  - Any new requirement needs a new program.

Another approach

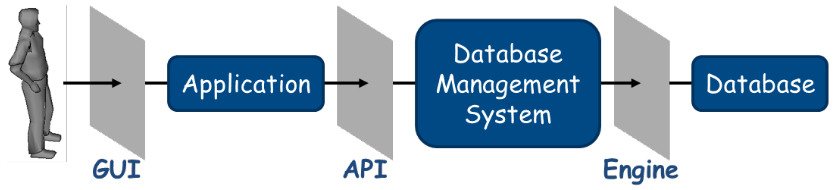
```
struct studentRecord{
    char studentId[12];
    char lastName[25];
    char firstName[25];
}; /* 62 bytes*
...

studentRecord sRec[20];
...
fread(sRec, 62, 20, file); /*read 20 student records*/
...
```

- Why?
  - Definition of the data is embedded in the application programs, rather than being stored separately and independently.
  - No control over the access and manipulation of data beyond that imposed by the application programs.
- How?
  - Database Management System (DBMS)

The Database Approach

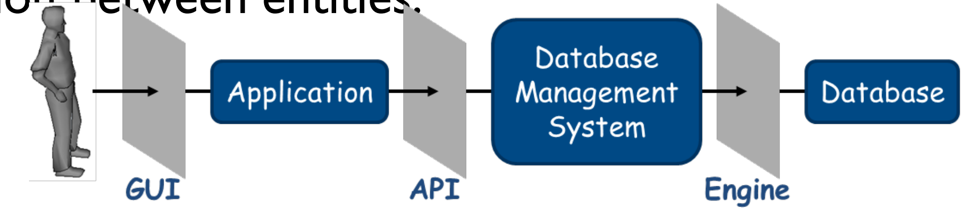
- **Database** is a collection of data used to describe describes real-world objects.
  - Blackboard's database contains information about entities (students and courses) and relationships (student registration in a course)
- **Database Management System (DBMS)** is a specialized software used to manage databases.
  - Ex.: PostgreSQL, Orable, MySQL, SQL Server, …
- **Database application** programs is a software that interacts with the database by issuing requests to the DBMS.
- **Database system** = data + DBMS + application programs



https://www.oracle.com/database/what-is-database/

Database

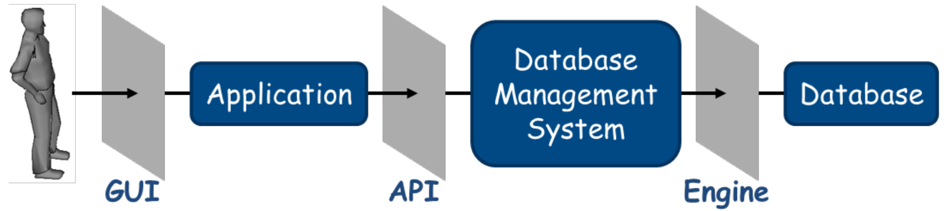
- Data are usually stored in a database.
- Database is a collection of logically related data (including description).
  - "Self-describing collection of integrated records"
- System catalog (data dictionary or metadata) provides description of data to enable program–data independence.
- Databases store data and the relationships among the data.
- Logically related data comprises entities, attributes, and relationships of an organization's information.
- (Relational) Databases store data in tables (or "relations").
  - SCHEMA: structure of the database.
  - ENTITY: distinct object that is to be represented in the database.
  - ATTRIBUTE: property that describes some aspect of the object that we wish to record.
  - RELATIONSHIP: association between entities.



HALMSTAD UNIVERSITY

Database Management System

- DBMS is A software system that enables users to define, create, maintain, and control access to the database.
  - Data Definition Language (DDL).
    - Permits specification of data types, structures and any data constraints.
    - All specifications are stored in the database.
  - Data manipulation Languade(DML).
    - General enquiry facility (query language) of the data.
  - Controlled Access
    - Security system, which prevents unauthorized users accessing the database;
    - Integrity system, which maintains the consistency of stored data;
    - Concurrency control system, which allows shared access of the database;
    - Recovery control system to restore a database to a previous consistent state after HW or SW failure;
    - User-accessible catalog, which contains descriptions of the data in the database.



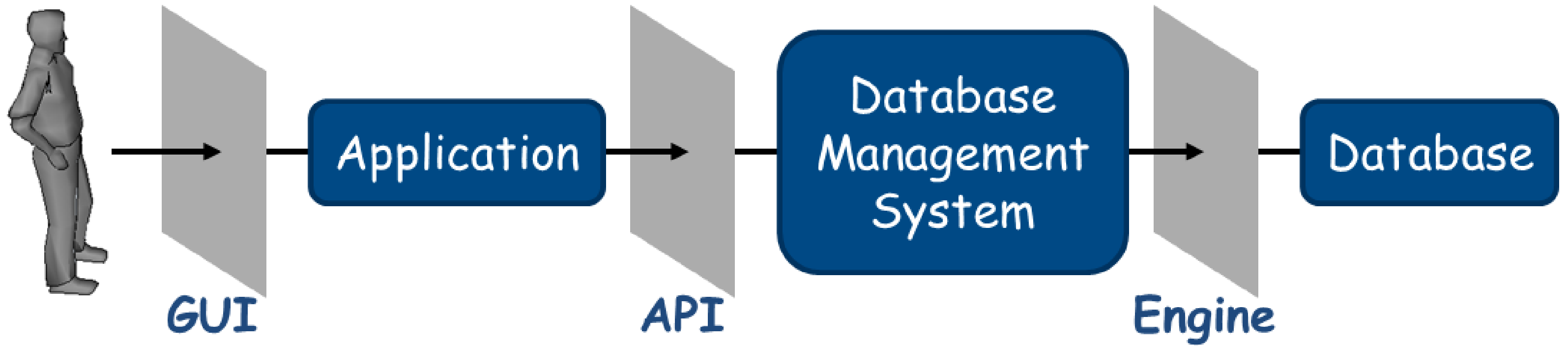
HALMSTAD UNIVERSITY

System Catalog

- Repository of information (metadata) describing the data in the database.
- One of the fundamental components of DBMS.
- Typically stores:
  - names, types, and sizes of data items;
  - constraints on the data;
  - names of authorized users;
  - data items accessible by a user and the type of access;
  - usage statistics.

Database Application program

- A computer program that interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS.

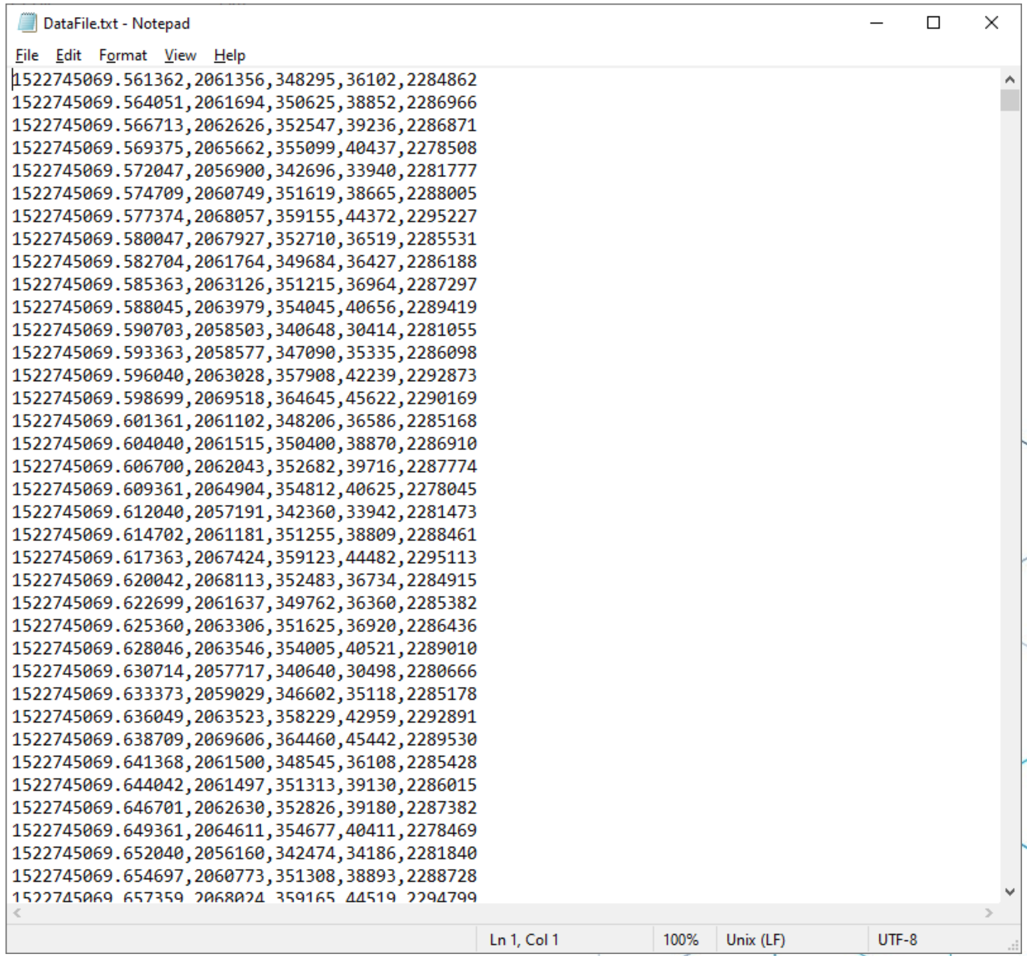


CSV File size: 65,7 MB (68 940 948 bytes)

Small exercise

- File with the following format:
  1. Datetime in epoch
  2. Amount of cash in store 1
  3. Amount of cash in store 2
  4. Amount of cash in store 3
  5. Amount of cash in store 4

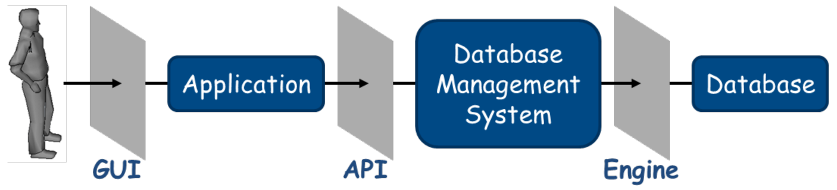
- Task
  - Design algorithms to:
    - Count the number of records (lines)
    - Count how many times store 4 had more than $2649629



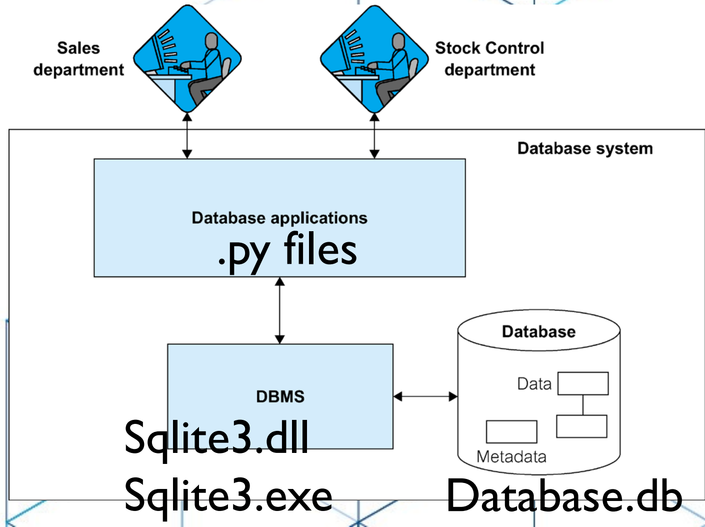
```
1522745069.561362,2061356,348295,36102,2284862
1522745069.564051,2061694,350625,38852,2286966
1522745069.566713,2062626,352547,39236,2286871
1522745069.569375,2065662,355099,40437,2278508
1522745069.572047,2056900,342696,33940,2281777
1522745069.574709,2060749,351619,38665,2288005
1522745069.577374,2068057,359155,44372,2295227
1522745069.580047,2067927,352710,36519,2285531
1522745069.582704,2061764,349684,36427,2286188
1522745069.585363,2063126,351215,36964,2287297
1522745069.588045,2063979,354045,40656,2289419
1522745069.590703,2058503,340648,30414,2281055
1522745069.593363,2058577,347090,35335,2286098
1522745069.596040,2063028,357908,42239,2292873
1522745069.598699,2069518,364645,45622,2290169
1522745069.601361,2061102,348206,36586,2285168
1522745069.604040,2061515,350400,38870,2286910
1522745069.606700,2062043,352682,39716,2287774
1522745069.609361,2064904,354812,40625,2278045
1522745069.612040,2057191,342360,33942,2281473
1522745069.614702,2061181,351255,38809,2288461
1522745069.617363,2067424,359123,44482,2295113
1522745069.620042,2068113,352483,36734,2284915
1522745069.622699,2061637,349762,36360,2285382
1522745069.625360,2063306,351625,36920,2286436
1522745069.628046,2063546,354005,40521,2289010
1522745069.630714,2057717,340640,30498,2280666
1522745069.633373,2059029,346602,35118,2285178
1522745069.636049,2063523,358229,42959,2292891
1522745069.638709,2069606,364460,45442,2289530
1522745069.641368,2061500,348545,36108,2285428
1522745069.644042,2061497,351313,39130,2286015
1522745069.646701,2062630,352826,39180,2287382
1522745069.649361,2064611,354677,40411,2278469
1522745069.652040,2056160,342474,34186,2281840
1522745069.654697,2060773,351308,38893,2288728
1522745069.657359,2068024,359165,44519,2294799
```

The Database Approach

- **Database** is a collection of data used to describe describes real-world objects.
  - Blackboard's database contains information about entities (students and courses) and relationships (student registration in a course)
- **Database Management System (DBMS)** is a specialized software used to manage databases.
  - Ex.: PostgreSQL, Orable, MySQL, SQL Server, …
- **Database application** programs is a software that interacts with the database by issuing requests to the DBMS.
- **Database system** = data + DBMS + application programs

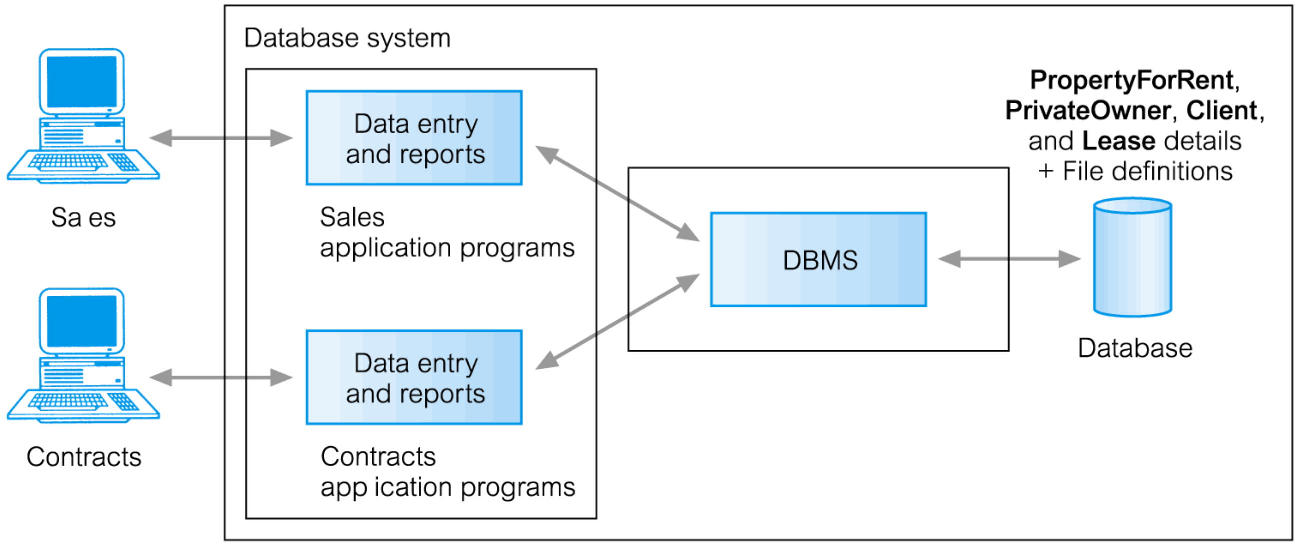


https://www.oracle.com/database/what-is-database/



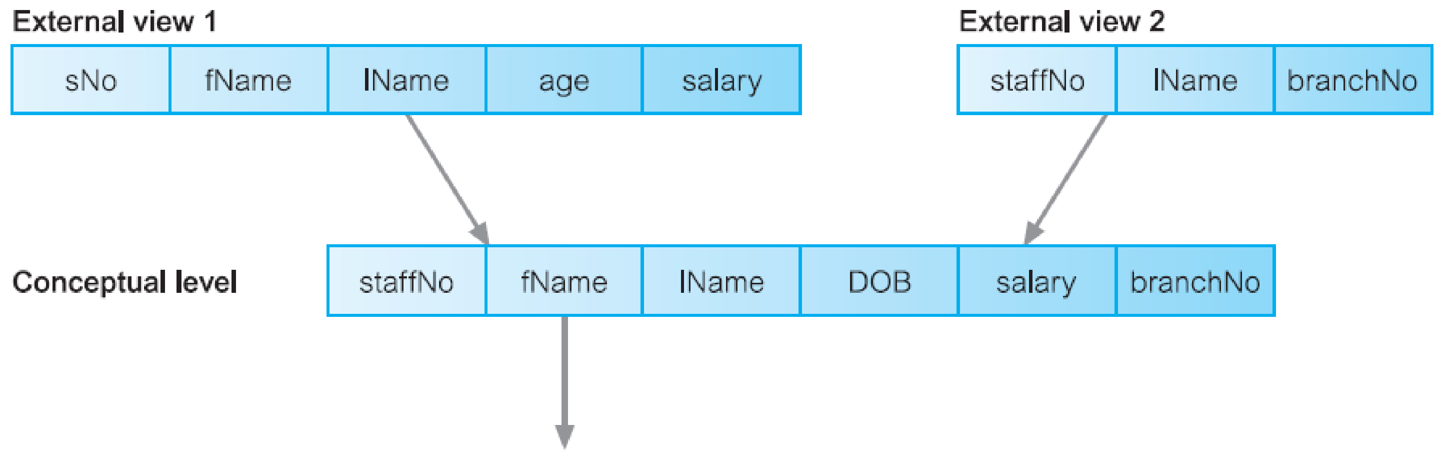
Database Views

- Allows each user to have his or her own view of the database.
- A view is essentially some subset of the database.
- **Benefits**
  - Reduce complexity
  - Provide a level of security
  - Provide a mechanism to customize the appearance of the database
  - Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed


Database system
Data entry and reports
Sales application programs
Data entry and reports
Contracts app ication programs
DBMS
PropertyForRent, PrivateOwner, Client, and Lease details + File definitions
Database
Sa es
Contracts

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Later in the course…



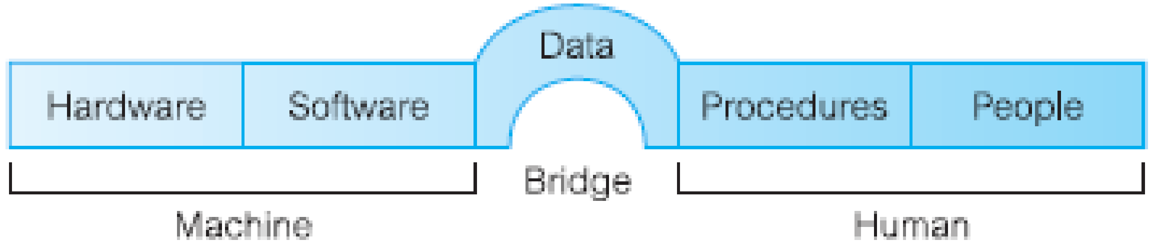
```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;                    /* pointer to next Staff record */
};
index staffNo; index branchNo;               /* define indexes for staff */
```

```
CREATE VIEW staff_view_1 AS
  SELECT
    staffNo AS sNo,
    fName,
    lName,
    YEAR(CURDATE()) - YEAR(DoB) AS age,
    salary
  FROM staff;
```

```
CREATE VIEW staff_view_2 AS
  SELECT
    staffNo,
    lName,
    branchNo
  FROM staff;
```

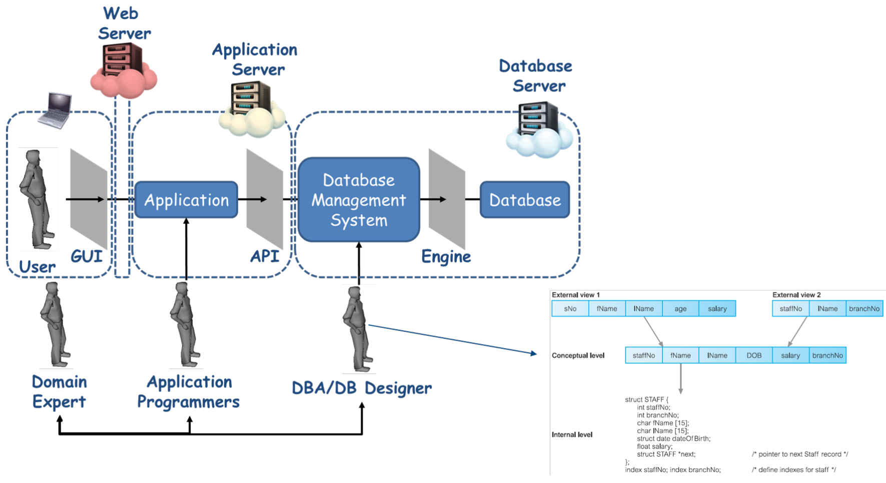
Components of the DBMS environment

- Hardware
  - A PC or a network of computers. Back- and front-ends
- Software
  - DBMS, OS, network software (if necessary) and also the application programs.
- Data
  - Used by the organization and a description of this data called the schema.
- Procedures
  - Instructions and rules that should be applied to the design and use of the database and DBMS.
    - Database Backup , recovery, balance load, …
- People
  - Database Administrator (DBA) and Database Designers (Logical and Physical)
  - Application Programmers
  - End Users (naive and sophisticated)
  - Researcher
  - …



HALMSTAD

Roles in the Database Environment



```
struct STAFF {
    int staffNo;
    int branchNo;
    char fName [15];
    char lName [15];
    struct date dateOfBirth;
    float salary;
    struct STAFF *next;          /* pointer to next Staff record */
};
index staffNo; index branchNo;   /* define indexes for staff */
```

- Data Administrator (DA)
  - Establish data policies, data standards, and data governance practices.
  - May not be directly involved in day-to-day database administration tasks.
- Database Administrator (DBA)
  - Manage database installation, configuration, and day-to-day operations.
  - Handle tasks like user access control, database performance tuning, and data backup and recovery.
- Database Designers (Logical and Physical)
  - Logical: data modeling, defining entities, attributes, and relationships, without concern for the physical implementation details.
  - Physical: translate the logical design into tables, indexes, and storage structures that optimize database performance.
- Application Programmers
  - write code to retrieve, manipulate, and update data in the database
- End Users (naive and sophisticated)
  - Naive: use applications to input, retrieve, and display data without needing to know how the database functions.
  - Sophisticated: can use query languages and tools to extract and manipulate data for more advanced purposes.

HALMSTAD UNIVERSITY

Functions of a DBMS

- Handle data Storage, Retrieval, and Update
- Provide efficient data access and administration
- Provide a convenient user interface
  - To define and manipulate data
- Guarantee integrity constraints
- Provide crash recovery
- Handle authentication and authorization
- Provides data independence
- Provides utility services
- Handle transactions and concurrent access

HALMSTAD

Functions of a DBMS - Handle transactions and concurrent access

- **Transactions** ensure data consistency and integrity.
  - A transaction is a sequence of one or more SQL operations (e.g., data retrieval, insertion, update, deletion) that are treated as a single, indivisible unit of work.
  - Follow the principles of ACID
    - Atomicity
      - An operation or sequence of operations is atomic if it is either performed to completion or not performed at all.
    - Consistency
      - Ensures that the database always makes sense from the application's point of view.
    - Isolation
      - Ensures that a transaction's intermediate states cannot be observed from outside of the transaction.
    - Durability
      - Ensures that a change committed to the database is never lost, even in case of power failure.
- **Concurrent access** refers to multiple users or processes attempting to access and modify the same data in a database simultaneously.
  - Concurrency control mechanisms are employed to manage simultaneous access, preventing data inconsistencies and conflicts.
  - Techniques such as locking, timestamps, and multi-version concurrency control are used to coordinate and control access to data.

HALMSTAD UNIVERSITY

Advantages and Disadvantages of DBMSs

+ Control of data redundancy
+ Data consistency
+ Sharing of data
+ Improved data integrity
+ Improved security
+ Enforcement of standards
+ Economy of scale
+ Balance of conflicting requirements
+ Improved data accessibility and responsiveness
+ Increased productivity
+ Improved maintenance through data independence
+ Increased concurrency
+ Improved backup and recovery

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- Cost of conversion
- Performance
- Higher impact of a failure

HALMSTAD UNIVERSITY

Comments or Questions?

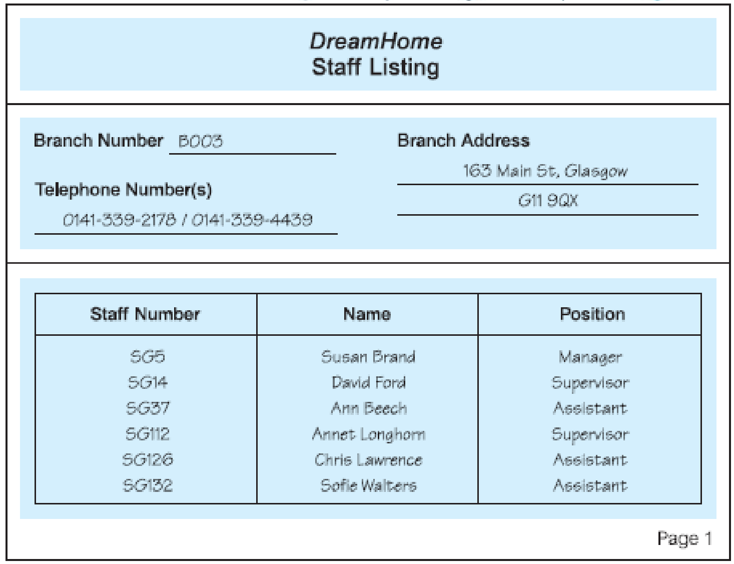
History of Database Systems

- Predatabase
  - All data were stored in separate files. Data integration was difficult. Data Storage was expensive and limited.
- First-generation (aka navigational era - 1960s)
  - Hierarchical and Network. Structured data in tree-like or graph-like structures.
- Second generation (aka relational era -1970s–1990s)
  - Relational. Represented data as tables with rows and columns.
- Third generation (aka "one-size-fits-all" era - 2000s')
  - Object-Relational and Object-Oriented. Integrated object-oriented features with the relational model, allowing for complex data structures and richer data modeling.
  - Too much work for the perceived benefit.
- Fourth generation (aka Big Data era – 2000 - present)
  - NoSQL (2000-2010)
    - "Not only" Structured Query Language - Non-relational data structures
    - Types include document-oriented, key-value, column-family, and graph databases.
  - Distributed Databases and Big Data (2010s-Present)
    - To manage vast volumes of data across multiple nodes and clusters.
  - NewSQL and In-Memory Databases (2010s-Present)
    - Combine the scalability of NoSQL with the transactional capabilities of traditional RDBMS.
    - In-memory databases have gained popularity for high-speed data processing.
  - Blockchain and Decentralized Databases (2010s-Present)
    - decentralized databases, providing tamper-resistant and transparent record-keeping.

HALMSTAD UNIVERSITY

The DreamHome Case Study

- DreamHome specializes in property management, taking an intermediate role between owners who wish to rent out their furnished property and clients of who require to rent furnished property for a fixed period.
- The first branch office of DreamHome was opened in 1992 in Glasgow in the UK. Since then, the Company has grown steadily and now has several offices in London, Aberdeen and Bristol.
- Each branch has an appropriate number and type of staff including a Manager, Supervisors, and Assistants.



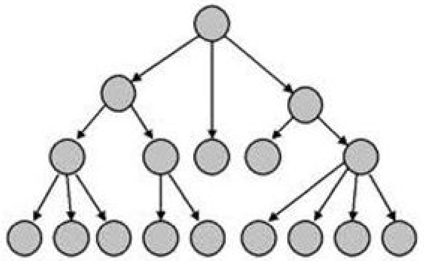
DreamHome
Staff Listing

Branch Number	B003
Telephone Number(s)	0141-339-2178 / 0141-339-4439
Branch Address	163 Main St, Glasgow G11 9QX

Staff Number	Name	Position
SG5	Susan Brand	Manager
SG14	David Ford	Supervisor
SG37	Ann Beech	Assistant
SG112	Annet Longhorn	Supervisor
SG126	Chris Lawrence	Assistant
SG132	Sofie Walters	Assistant

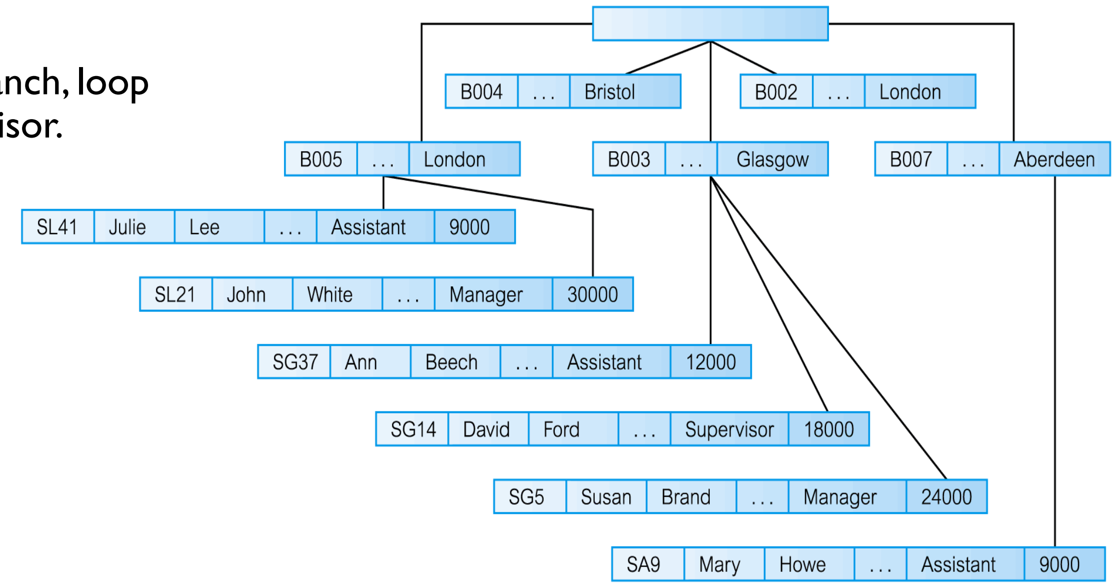
Page 1

Hierarchical Databases



- Hierarchical databases are organized in a tree-like structure.
  - Parent-children relationship node.
  - Each child record has only one parent. A parent can have multiple children.
  - One analogy is the file system in an operating system like Windows.
- Problems
  - Hard to write queries
    - Find all supervisors
      - Write loops to navigate among the branches and, for each branch, loop over each staff checking if the position field is equal to supervisor.
  - Data redundancy
  - Data independence

- Example: IBM's IMS

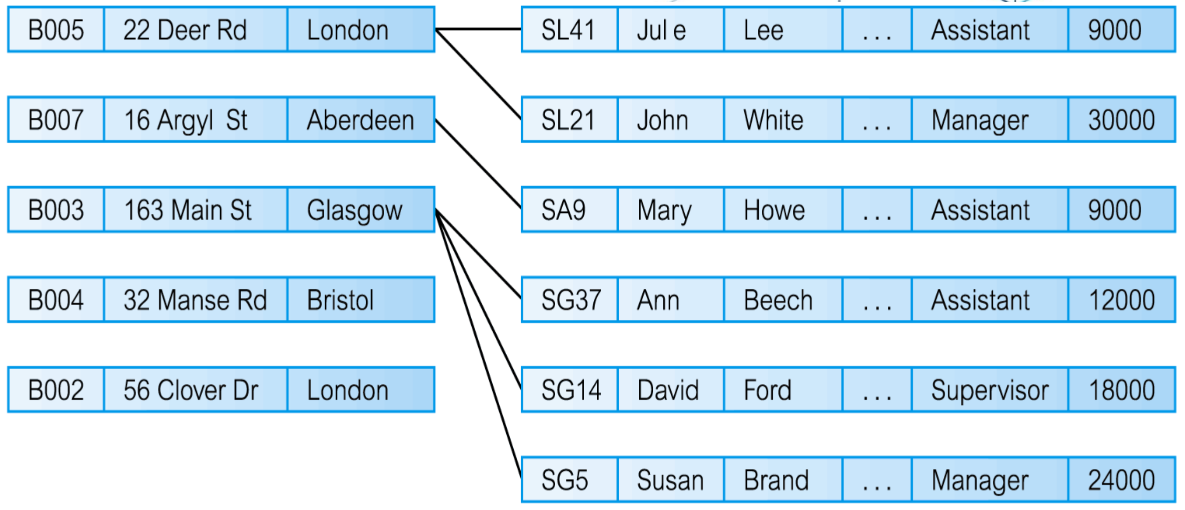


B004 ... Bristol
B002 ... London
B005 ... London
B003 ... Glasgow
B007 ... Aberdeen
SL41 Julie Lee ... Assistant 9000
SL21 John White ... Manager 30000
SG37 Ann Beech ... Assistant 12000
SG14 David Ford ... Supervisor 18000
SG5 Susan Brand ... Manager 24000
SA9 Mary Howe ... Assistant 9000

HALMSTAD UNIVERSITY

Network Databases

- Network databases are organized as a graph structure.
  - Are hierarchical databases but a network node can have a relationship with multiple entities, i.e. it can have more than one parent.
  - Problems
    - Hard to write queries
    - Find all supervisors
      - » Write loops to navigate among the branches and, for each branch, loop over each staff checking if the position field is equal to supervisor.
    - Data redundancy
    - Data independence
- Example: IDS



B005	22 Deer Rd	London
B007	16 Argyl St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

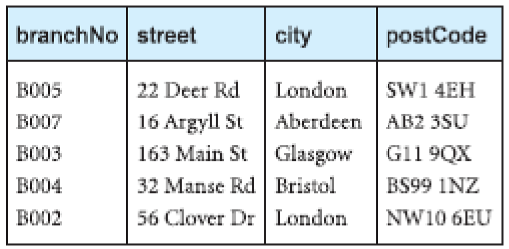
SL41	Jul e	Lee	. . .	Assistant	9000
SL21	John	White	. . .	Manager	30000
SA9	Mary	Howe	. . .	Assistant	9000
SG37	Ann	Beech	. . .	Assistant	12000
SG14	David	Ford	. . .	Supervisor	18000
SG5	Susan	Brand	. . .	Manager	24000

HALMSTAD UNIVERSITY

Relational Database

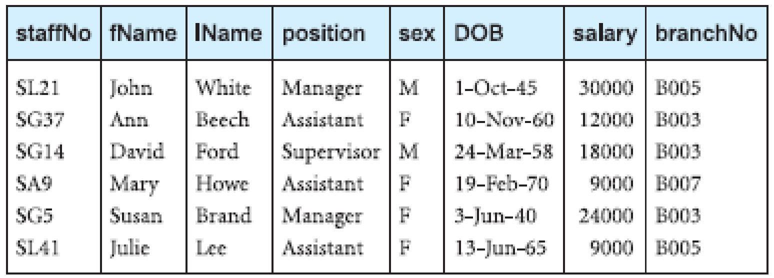
- Proposed by Ted Codd at IBM Research in the 70's
  - To cope with the problem of rewriting IMS and Codasyl programs when database's schema or layout changed.
  - Application programs must not be affected by modifications to the internal data representation.
  - Decouple the logical and physical structure of the database.
    - Use abstract objects called relations to represent the data for a single entity
  - Access data through high-level declarative language
    - What you want instead of how you want
- Problem
  - Easy to write queries
  - Not all data fit naturally into a tabular structure

Branch



branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff



staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

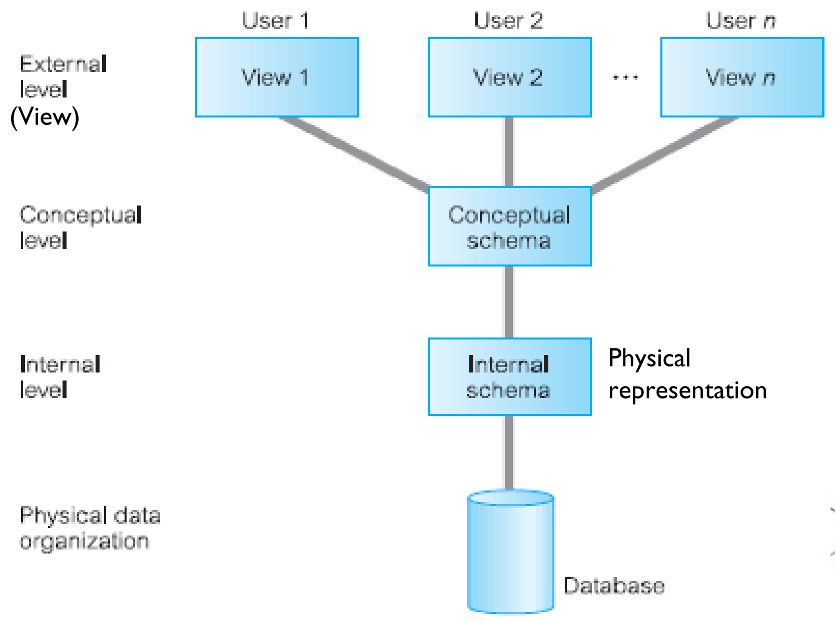
Comments or Questions?

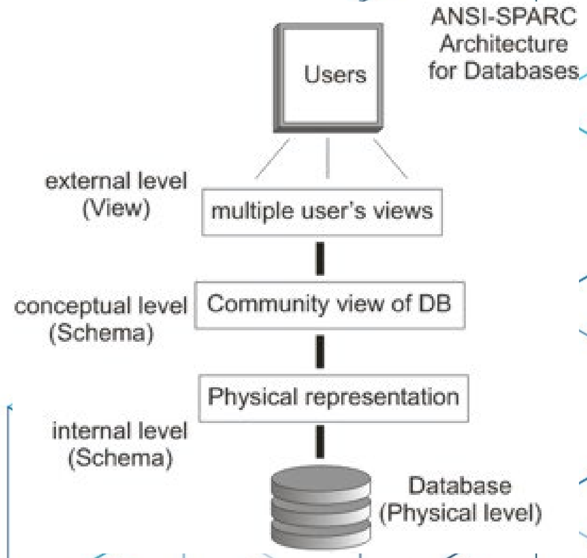
Chapter 2

Database Environment

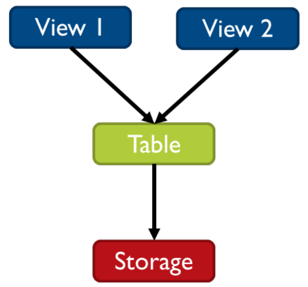
Three-Level ANSI-SPARC Architecture

- Degrees of data abstraction
  - External, conceptual, and internal.
- Aim to separate the users' view
  - All users should be able to access same data.
  - A user's view is immune to changes made in other views.
  - Users should not need to know physical database storage details.
  - DBA should be able to change conceptual structure of database without affecting all users.
  - DBA should be able to change database storage structures without affecting the users' views.
  - Internal structure of database should be unaffected by changes to physical aspects of storage.

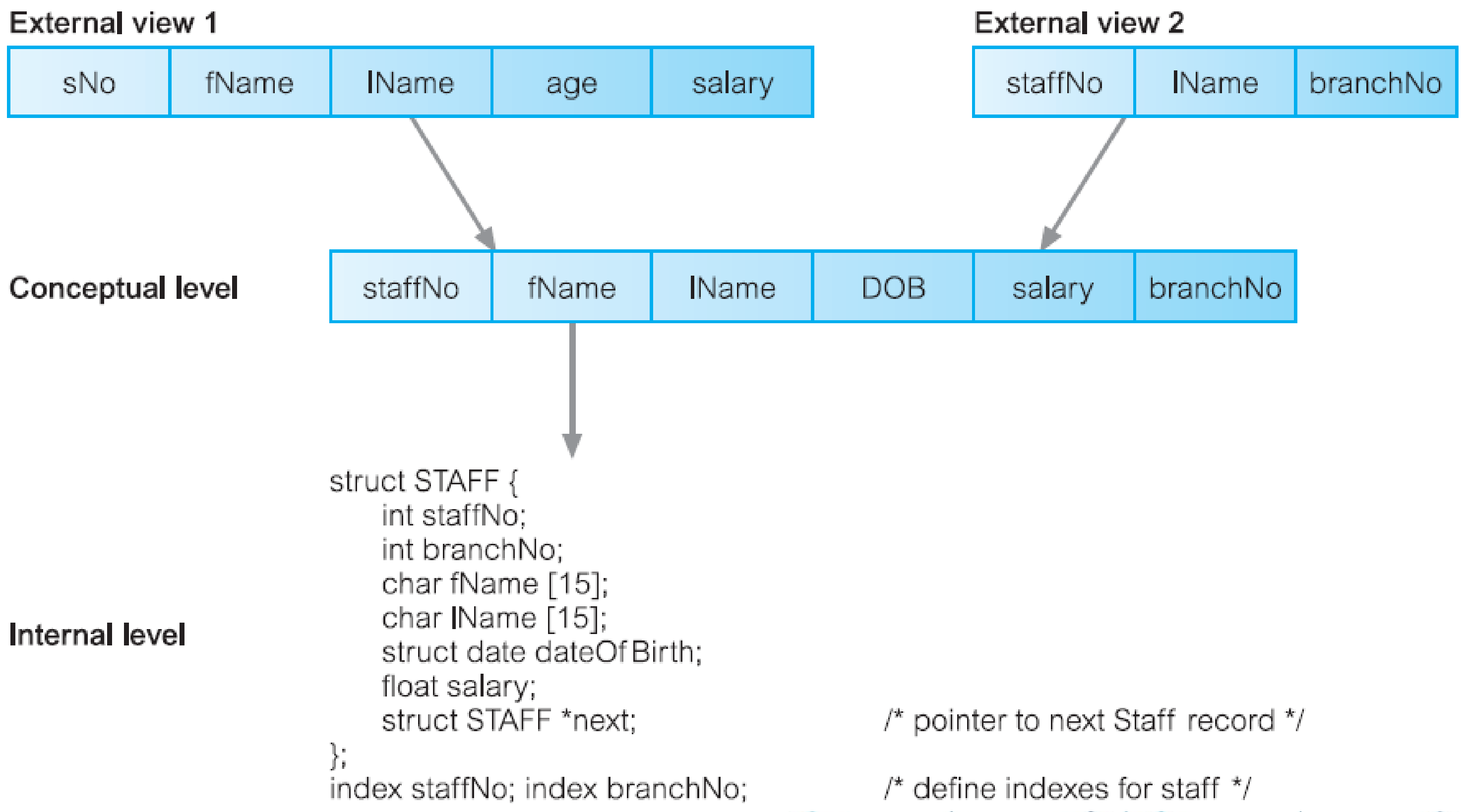




Differences between the three levels

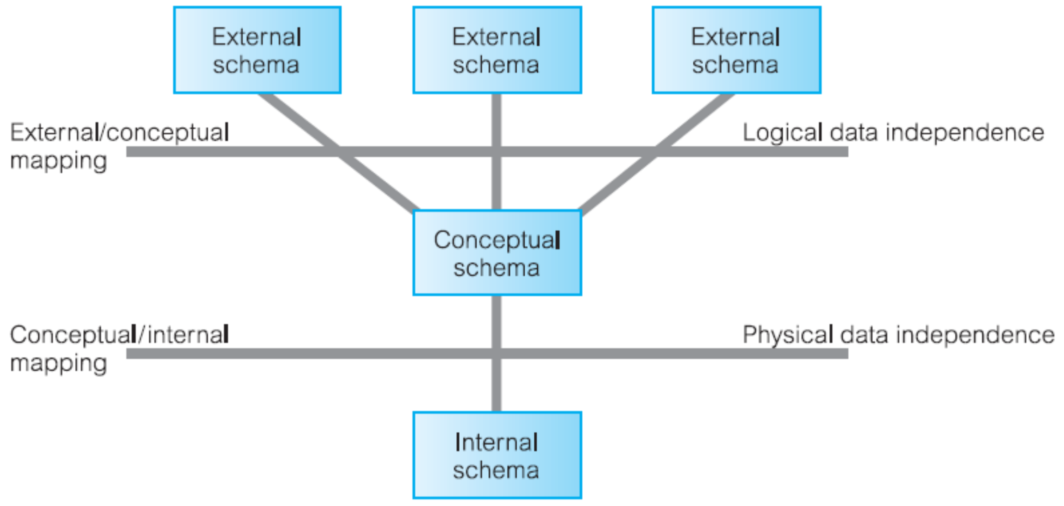


- External Level or View level
  - End users' view of the database.
  - Describes that part of database that is relevant to a particular user.
  - What information is exposed to users, what are they allowed to see…
- Conceptual Level
  - Community view of the database.
  - What tables are there, their attributes and relationships.
  - What constraints should the DBMS enforce…
- Internal Level or Physical level
  - Physical representation of the database on the computer.
  - How data is stored, where it is located, how many bytes, what type of indexes…



```
struct STAFF {
    int staffNo;
    int branchNo;
    char fName [15];
    char lName [15];
    struct date dateOfBirth;
    float salary;
    struct STAFF *next;            /* pointer to next Staff record */
};
index staffNo; index branchNo;     /* define indexes for staff */
```

Data Independence


External schema
External schema
External schema
External/conceptual mapping
Logical data independence
Conceptual schema
Conceptual/internal mapping
Physical data independence
Internal schema

- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema (e.g. addition/removal of entities).
    - Modify table definitions without having change the application's views.
      - Add/drop/rename attributes for a table or rename a table.
  - Should not require changes to external schema or rewrites of application programs.
- Physical Data Independence
  - Refers to immunity of conceptual schema to changes in the internal schema (e.g. using different file organizations, storage structures/devices).
    - Change how/where database objects are represented in the physical storage.
    - Change to 64-bits for integers and move a table to another disk/machine.
  - Should not require change to conceptual or external schemas.

Database Languages

- Data Definition Language (DDL)
  - Allows the DBA or user to describe and name entities, attributes, and relationships required for the application
    - plus any associated integrity and security constraints.
- Data Manipulation Language (DML)
  - Provides basic data manipulation operations on data held in the database.
- Procedural DML
  - Allows user to tell system exactly how to manipulate data.
- Non-Procedural DML
  - Allows user to state what data is needed rather than how it is to be retrieved.
- Fourth Generation Languages (4GLs)
  - Advances commands for data management and data analysis

HALMSTAD UNIVERSITY

Data Models

- Integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization.
- Data Model comprises:
  - – a structural part;
  - – a manipulative part;
  - – possibly a set of integrity rules.

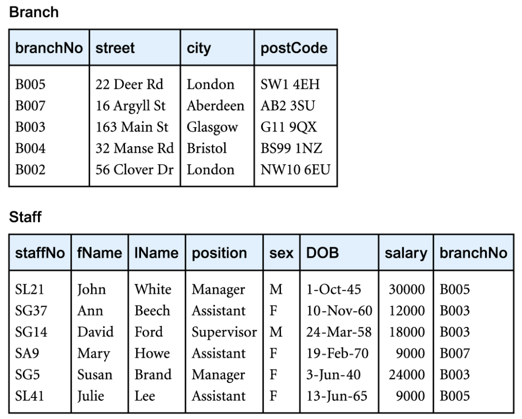
- Purpose
  - – To represent data in an understandable way, ie. to hide storage details and present the users with a conceptual view of the database.
    - Programs refer to the data model constructs rather than data storage details.

Masri, E., & Navathe, S. (2016). Fundamentals of database systems.

Categories of Data Models

- Object-Based Data Models
  - Entity-Relationship
  - Semantic
  - Functional
  - Object-Oriented.
- Record-Based Data Models
  - Relational Data Model (tables)
  - Network Data Model (graphs)
  - Hierarchical Data Model (graphs with single-parent nodes)
- Physical Data Models

Categories of Data Models



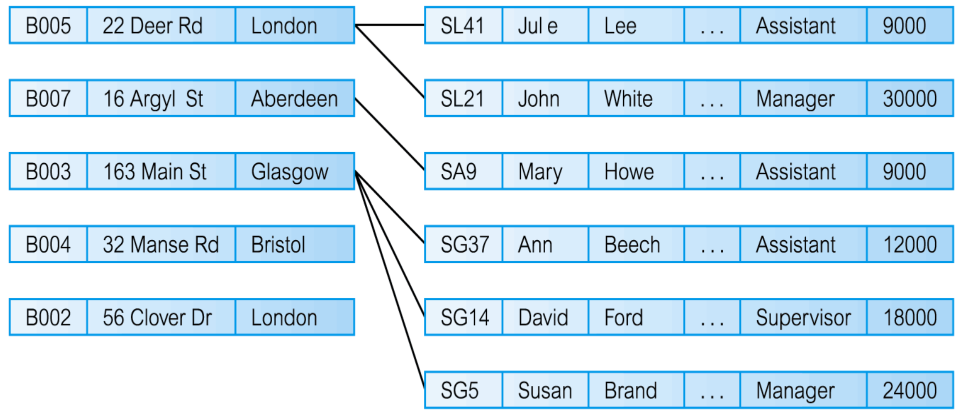
Branch

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

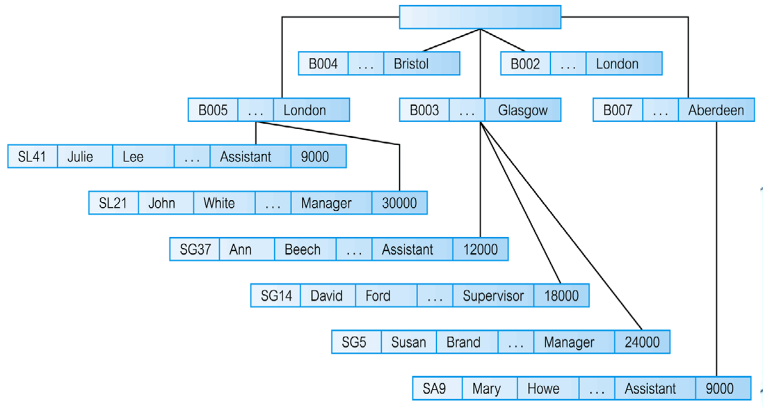
Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

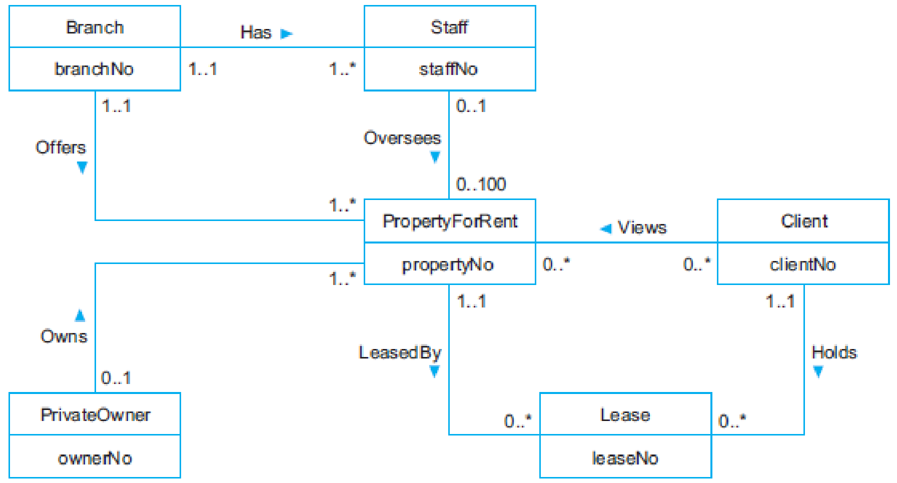
Relational Data Model


B005 | 22 Deer Rd | London
B007 | 16 Argyl St | Aberdeen
B003 | 163 Main St | Glasgow
B004 | 32 Manse Rd | Bristol
B002 | 56 Clover Dr | London
SL41 | Jul e | Lee | . . . | Assistant | 9000
SL21 | John | White | . . . | Manager | 30000
SA9 | Mary | Howe | . . . | Assistant | 9000
SG37 | Ann | Beech | . . . | Assistant | 12000
SG14 | David | Ford | . . . | Supervisor | 18000
SG5 | Susan | Brand | . . . | Manager | 24000

Network Data Model


B004 | . . . | Bristol
B002 | . . . | London
B005 | . . . | London
B003 | . . . | Glasgow
B007 | . . . | Aberdeen
SL41 | Julie | Lee | . . . | Assistant | 9000
SL21 | John | White | . . . | Manager | 30000
SG37 | Ann | Beech | . . . | Assistant | 12000
SG14 | David | Ford | . . . | Supervisor | 18000
SG5 | Susan | Brand | . . . | Manager | 24000
SA9 | Mary | Howe | . . . | Assistant | 9000

Hierarchical Data Model



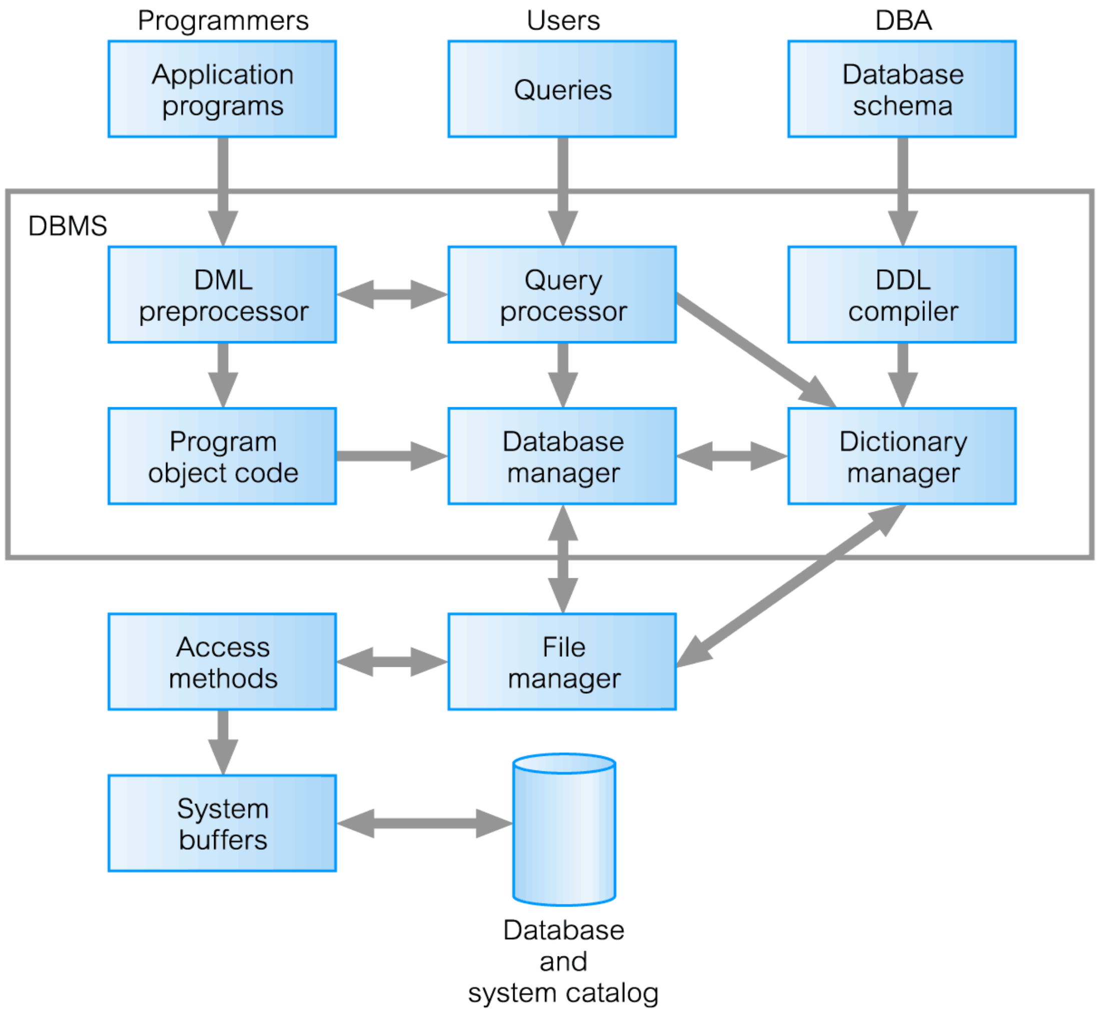
Entity–Relationship Data Model

Conceptual Modeling

- Conceptual schema is the core of a system supporting all user views.
- Should be complete and accurate representation of an organization’s data requirements.
- Conceptual modeling is process of developing a model of information use that is independent of implementation details.
- Result is a conceptual data model.

- Abstraction: What to store, not how to store.
- Entity-Relationship Diagrams (ERDs): Modeling tool
- Data Modeling Languages
- Requirements Analysis
- Identification of Entities, Attributes, and Relationships
- Normalization: Involves organizing data to minimize redundancy and improve data integrity.
- Cardinality and Cardinality Constraints
- Agnostic to Implementation
- Communication Tool
- Foundation for Logical and Physical Design

Components of a DBMS



HALMSTAD UNIVERSITY

Comments or Questions?

HALMSTAD UNIVERSITY

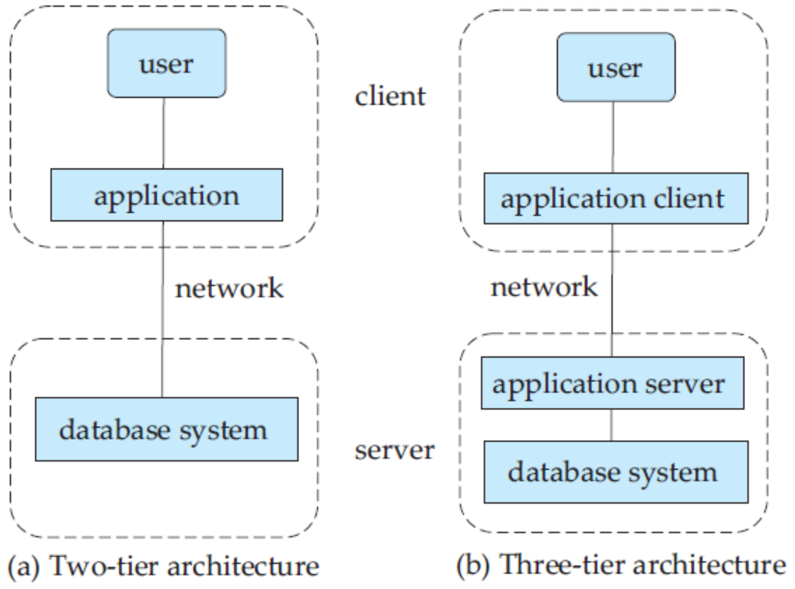
Chapter 3

Database Architectures and the Web

HALMSTAD UNIVERSITY

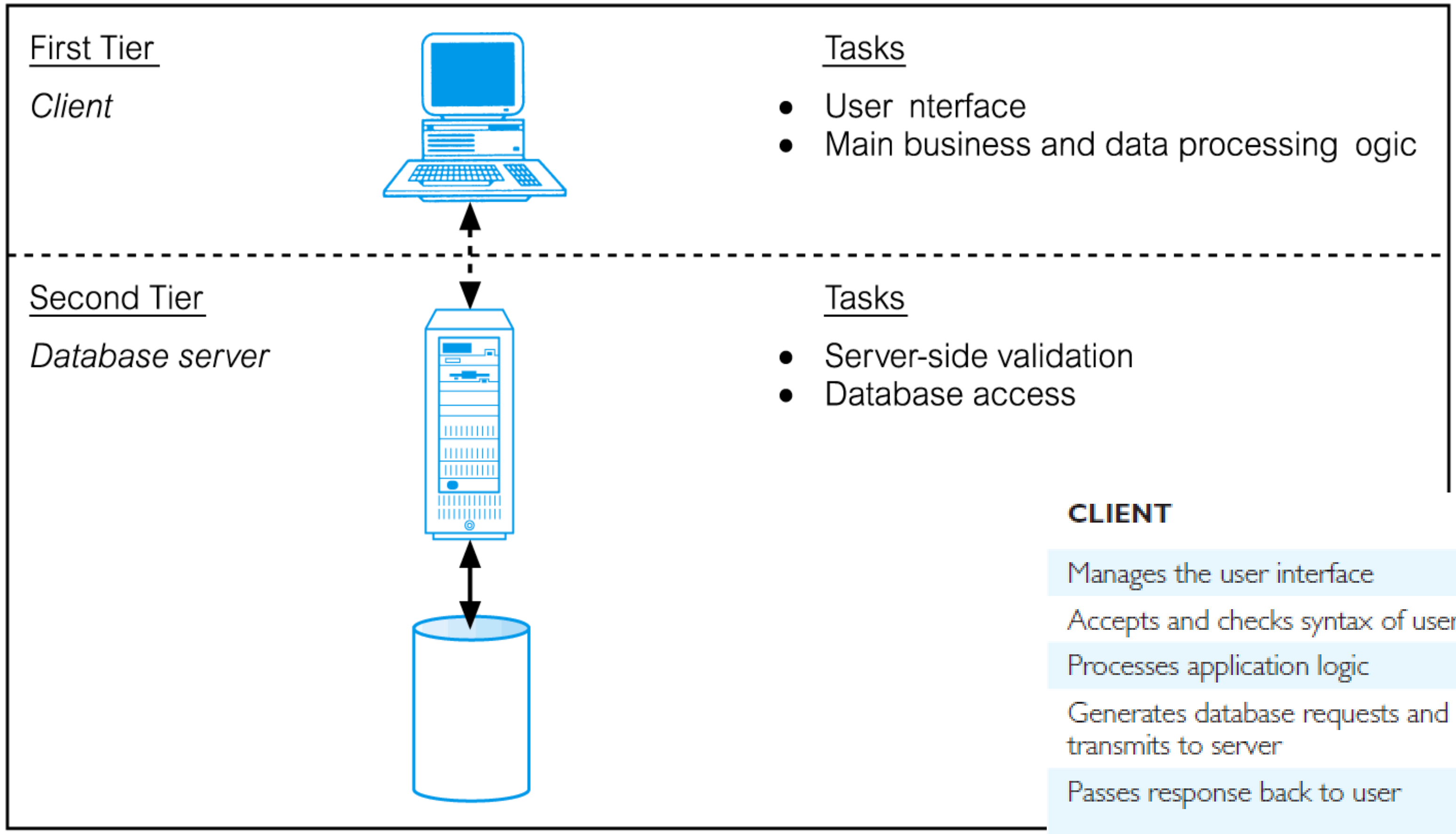
Multi-user DBMS Architectures

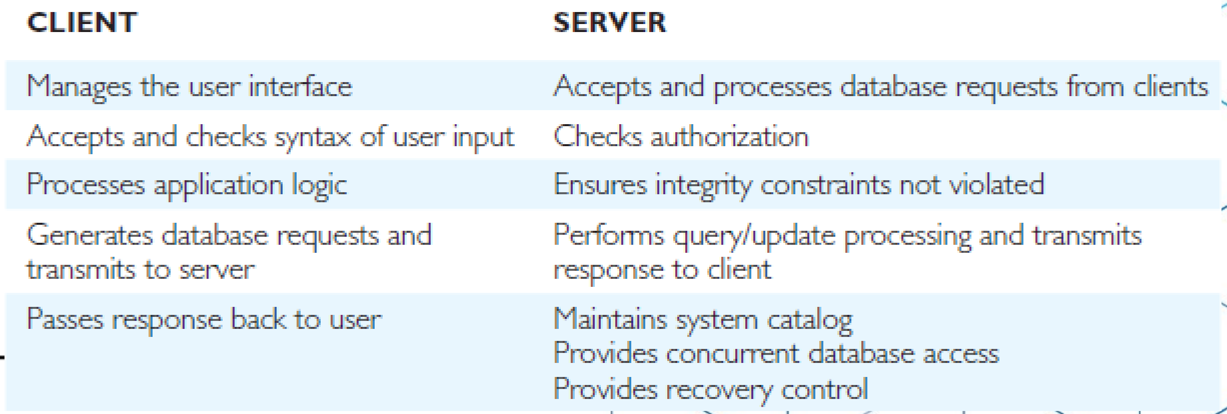
- Teleprocessing
  - Traditional architecture using a single mainframe with a number of terminals attached.
- File-server
  - File-server is connected to several workstations across a network.
  - Database resides on file-server.
  - DBMS and applications run on each workstation.
  - Disadvantages
    - Significant network traffic, copy of DBMS on each workstation, and concurrency, recovery and integrity control more complex.
- Client-server
  - Client (tier 1) manages UI and runs applications.
  - Server (tier 2) holds database and DBMS.
  - Advantages:
    - wider access to existing databases;
    - increased performance;
    - possible reduction in hardware costs;
    - reduction in communication costs;
    - increased consistency.



2010 - Silberschatz, Korth & Sudarshan - Database System Concepts

Traditional Two-Tier Client-Server

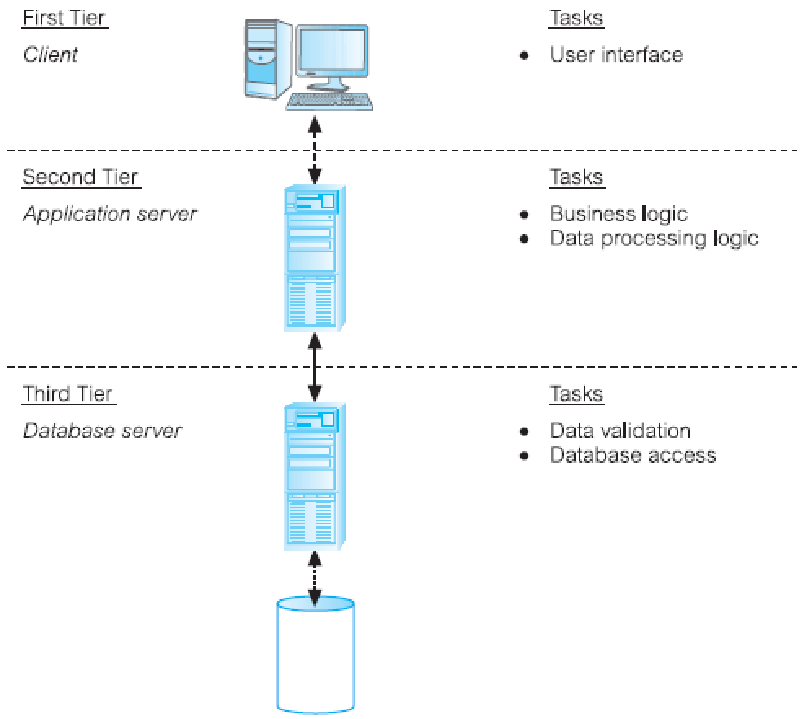


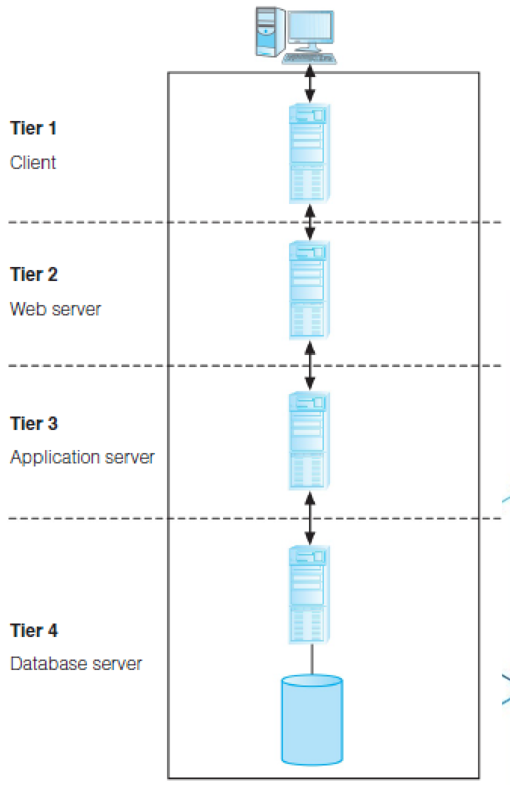


CLIENT	SERVER
Manages the user interface	Accepts and processes database requests from clients
Accepts and checks syntax of user input	Checks authorization
Processes application logic	Ensures integrity constraints not violated
Generates database requests and transmits to server	Performs query/update processing and transmits response to client
Passes response back to user	Maintains system catalog Provides concurrent database access Provides recovery control

Three-Tier and n-Tier Client-Server

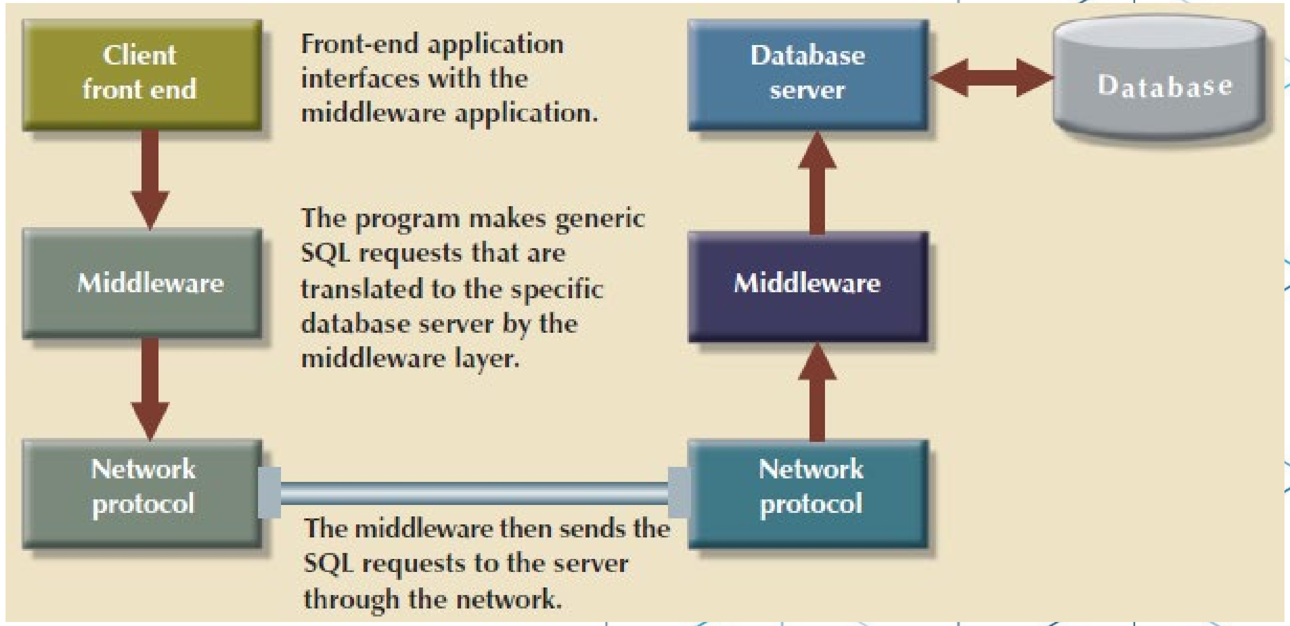
- The need for enterprise scalability challenged the traditional two-tier client–server model.
  - The three-tier architecture can be expanded to n tiers (e.g. 4 tiers), with additional tiers providing more flexibility and scalability.
  - Applications servers host API to expose business logic and business processes for use by other applications.





Middleware

- Middleware is a generic term used to describe software that mediates with other software and allows for communication between disparate applications in a heterogeneous system.
- The need for middleware arises when distributed systems become too complex to manage efficiently without a common interface.
- Network independence
- Database server independence
- Techniques
  - RPC
  - Publish/subscribe
  - Message-oriented
  - …


Client front end
Front-end application interfaces with the middleware application.
Database server
Database
Middleware
The program makes generic SQL requests that are translated to the specific database server by the middleware layer.
Middleware
Network protocol
The middleware then sends the SQL requests to the server through the network.
Network protocol

2009 - Rob & Coronel - Database Systems - Design, Implementation, and Management

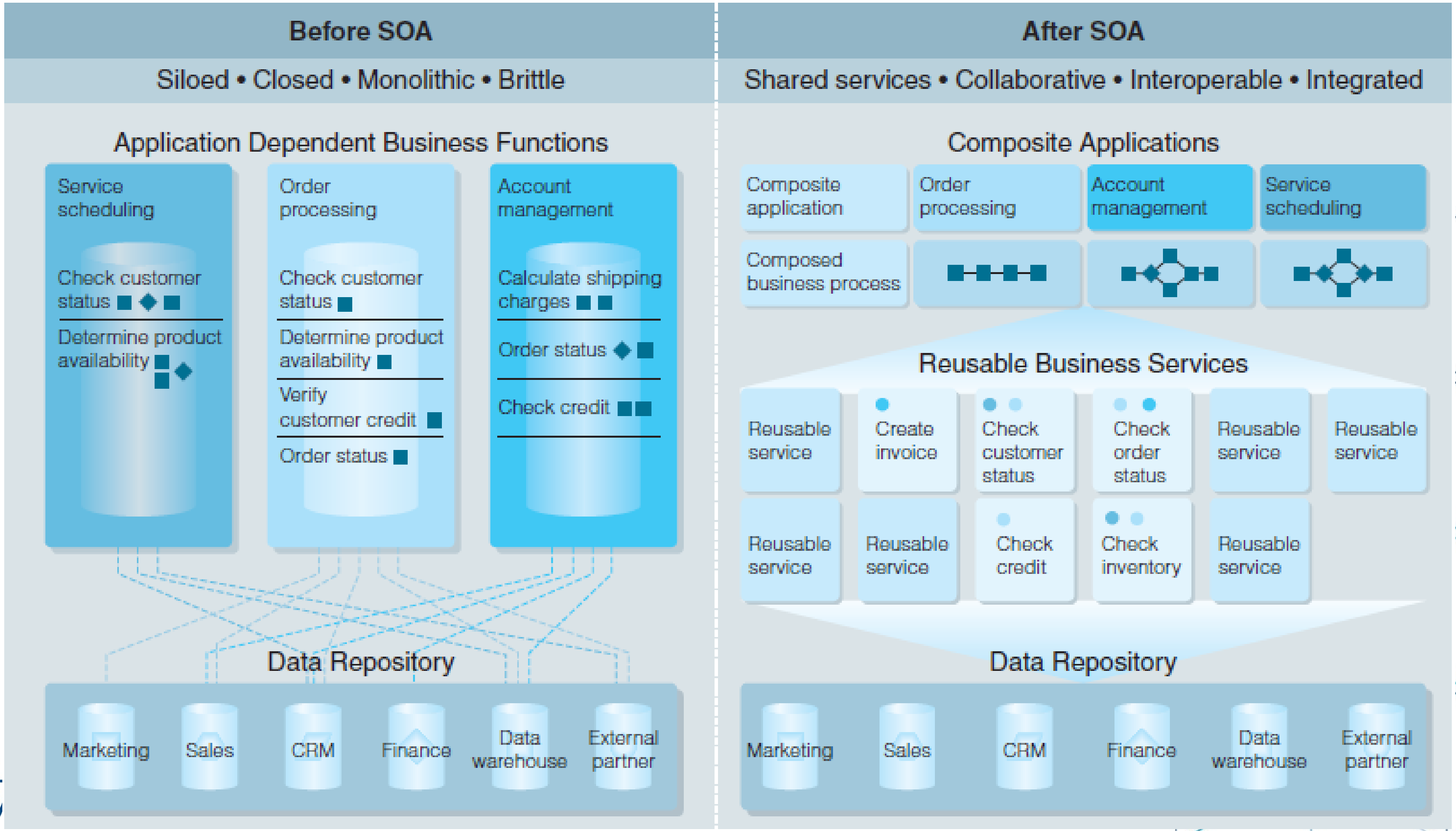
Web Services and SOA

- Web services
  - Software system designed to support interoperable machine-to-web service machine interaction over a network.
  - Share business logic, data, and processes through a programmatic interface across a network.
  - Developers can add the Web service to a Web page (or an executable program) to offer specific functionality to users.

- Service-Oriented Architectures (SOA)
  - A business-centric software architecture for building applications that implement business processes as sets of services published at a granularity relevant to the service consumer.
  - Services can be invoked, published, and discovered, and are abstracted away from the implementation using a single standards-based form of interface.

HALMSTAD UNIVERSITY

Service-Oriented Architectures

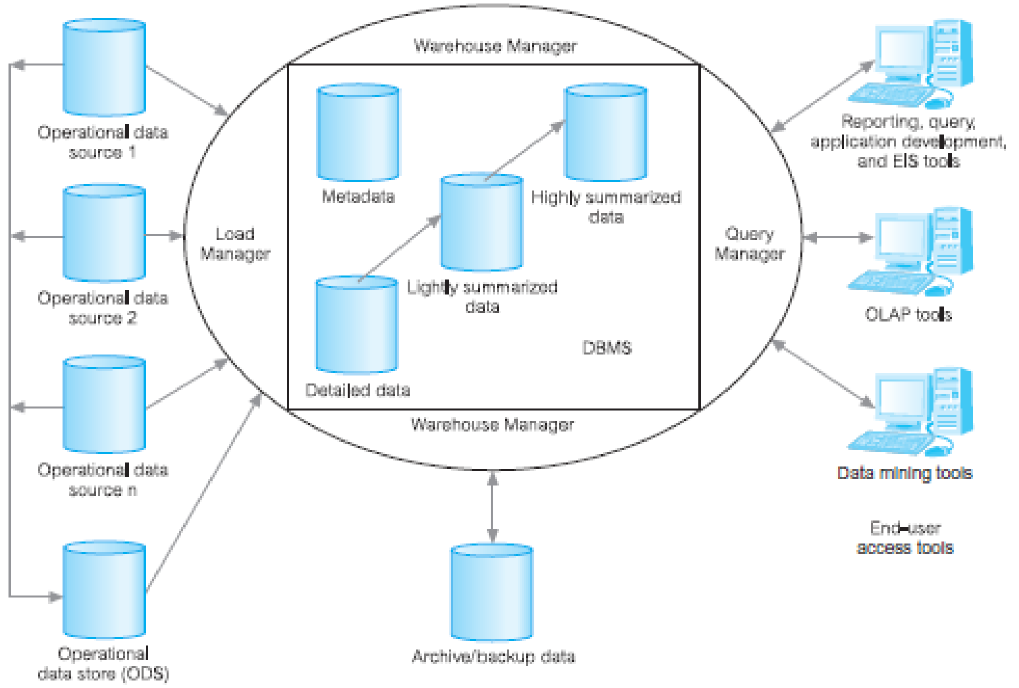


Centralized vs Distributed DBMSs

- With a centralized database system, the DBMS and database are stored at a single site that is used by several other systems too.
- In a distributed database system, the actual database and the DBMS software are distributed from various sites that are connected by a computer network.
  - A distributed database is a logically interrelated collection of shared data, physically distributed over a computer network.
  - A distributed DBMS is the software system that permits the management of the distributed database and makes the distribution transparent to users.
  - A DDBMS consists of a single logical database split into a number of fragments.
  - Each fragment is stored on one or more computers (replicas) under the control of a separate DBMS, with the computers connected by a network.
  - Each site is capable of independently processing user requests that require access to local data (that is, each site has some degree of local autonomy) and is also capable of processing data stored on other computers in the network.

Data Warehouse

- Solution to meet the requirements of a system capable of supporting decision making, receiving data from multiple operational data sources.
- A consolidated/integrated view of corporate data drawn from disparate operational data sources and a range of end-user access tools capable of supporting simple to highly complex queries to support decision making.



HALMSTAD UNIVERSITY

Cloud Computing

- "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [National Institute of Standards and Technology (NIST)]

Cloud Computing – Key Characteristics

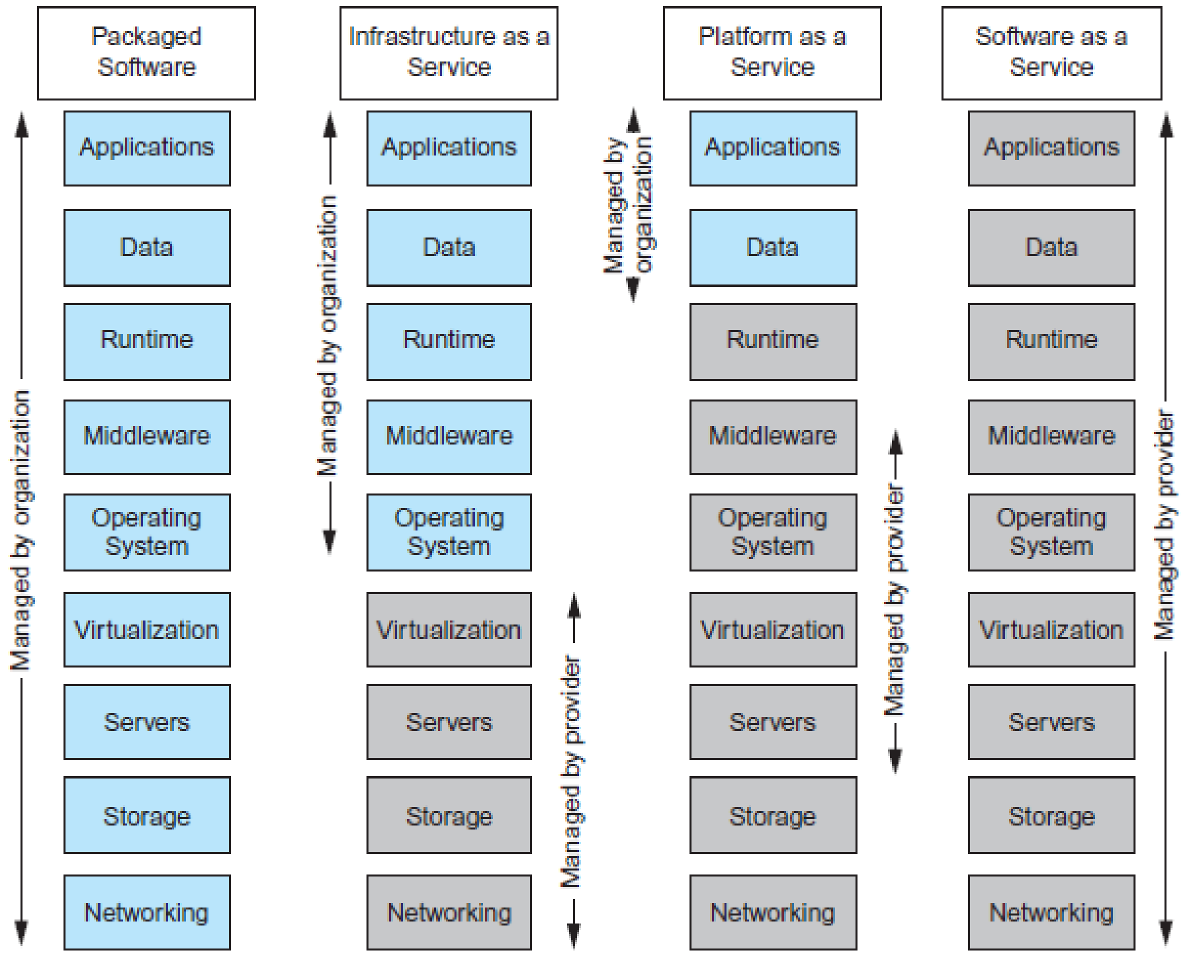
- On-demand self-service
  - – Consumers can obtain, configure and deploy cloud services without help from provider.
- Broad network access
  - – Accessible from anywhere, from any standardized platform (PC, smart phone)
- Resource pooling
  - – Provider's computing resources are pooled to serve multiple consumers, with different physical and virtual resources (processing, memory, storage) dynamically assigned and reassigned according to consumer demand.
- Rapid elasticity
  - – Provider's capacity caters for customer's spikes in demand and reduces risk of outages and service interruptions. Capacity can be automated to scale rapidly based on demand.
- Measured service
  - – Provider uses a metering capability to measure usage of service (e.g. storage, processing, bandwidth, and active user accounts).

HALMSTAD UNIVERSITY

Cloud Computing – Comparison of Services Models

Service Model

- Software as a Service (SaaS)
  - Software and data hosted on cloud.
  - Accessed via thin clients (e.g. web browser).
  - Example: Google's Gmail.
- Platform as a Service (PaaS)
  - Allows creation of web applications without buying/maintaining the software and underlying infrastructure.
  - Provider manages the infrastructure.
  - Customer controls deployment of applications and possibly configuration.
  - Ex: Google's App Engine and Microsoft's Azure.
- Infrastructure as a Service (IaaS)
  - Provider's offer servers, storage, network and operating systems – typically a platform virtualization environment – to consumers as an on-demand service, in a single bundle and billed according to usage.
  - Hosting websites.
  - Example: Amazon's Elastic Compute Cloud (EC2)



HALMSTAD UNIVERSITY

Cloud-based database solutions

- As a type of Software as a Service (SaaS), cloud-based database solutions fall into two basic categories:
  - Data as a Service (DaaS)
    - Offers full database functionality to application developers.
    - Provides a management layer that provides continuous monitoring and configuring of the database to optimized scaling, high availability, multi-tenancy (i.e, serving multiple client organizations), and effective resource allocation in the cloud, thereby sparing the developer from ongoing database administration tasks.
    - IBM offers weather data DaaS.
  - Database as a Service (DBaaS).
    - Services enables data definition in the cloud and subsequently querying.
    - Does not implement typical DBMS interfaces (e.g. SQL) but instead data is accessed via web interface or vendor-provided API.
    - Enables organization with valuable data to offer access to others.
  - Key difference between the two options is mainly how the data is managed.

HALMSTAD UNIVERSITY

Benefits of Cloud Computing

- Cost-Reduction
  - Avoid up-front capital expenditure.
- Scalability/Agility
  - Organisations set up resources on an as-needs basis.
- Improved Security
  - Providers can devote expertise & resources to security; not affordable by customer.
- Improved Reliability
  - Providers can devote expertise & resources on reliability of systems; not affordable by customer.
- Access to new technologies
  - Through use of provider's systems, customers may access latest technology.
- Faster development
  - Provider's platforms can provide many of the core services to accelerate development cycle.
- Large scale prototyping/load testing
  - Providers have the resources to enable this.
- More flexible working practices
  - Staff can access files using mobile devices.
- Increased competitiveness
  - Allows organizations to focus on their core competencies rather than their IT infrastructures.

HALMSTAD UNIVERSITY

Risks of Cloud Computing

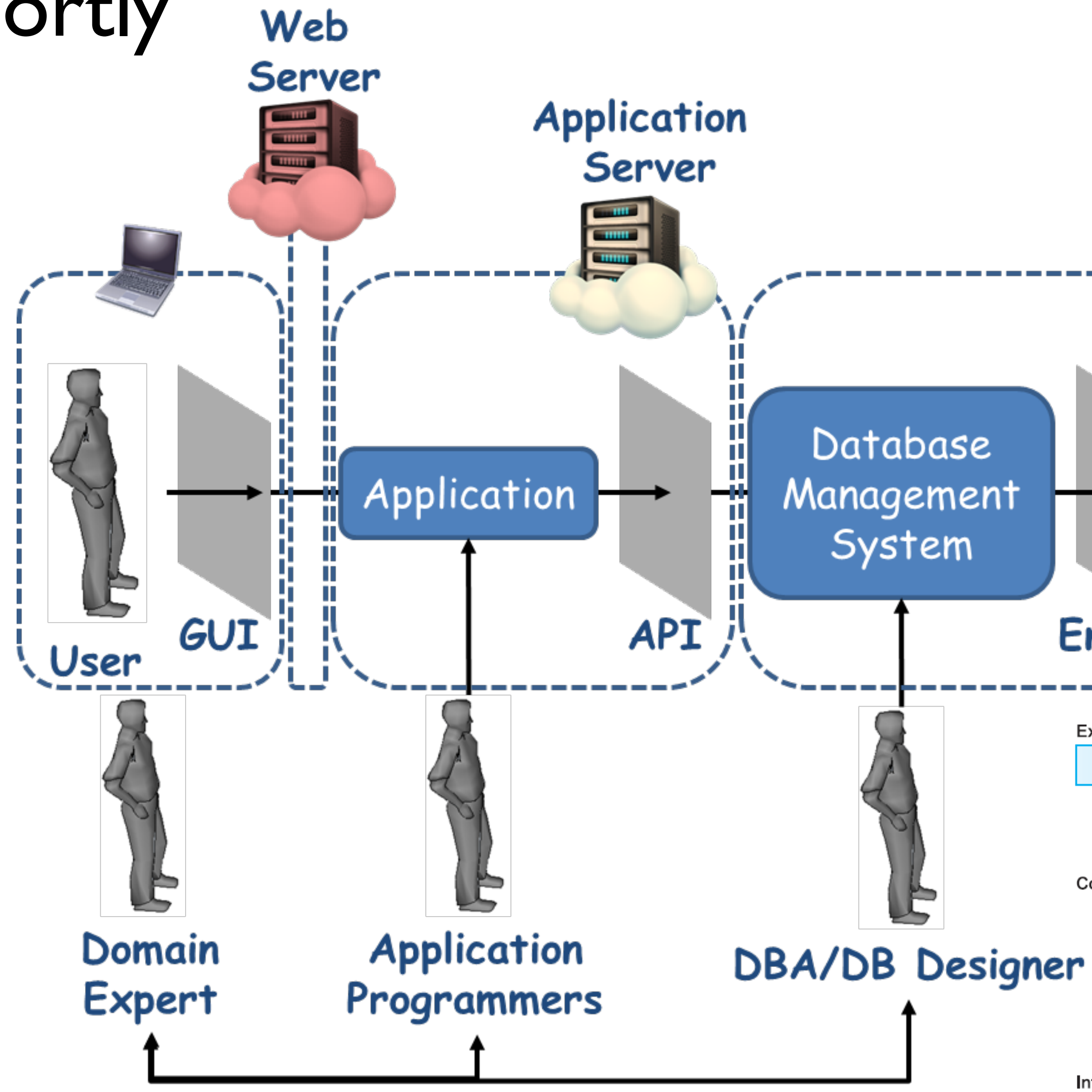
- Network Dependency
  - Power outages, bandwidth issues and service interruptions.
- System Dependency
  - Customer's dependency on availability and reliability of provider's systems.
- Cloud Provider Dependency
  - Provider could become insolvent or acquired by competitor, resulting in the service suddenly terminating.
- Lack of control
  - Customers unable to deploy technical or organizational measures to safeguard the data. May result in reduced availability, integrity, confidentiality, intervenability and isolation.
- Lack of information on processing transparency

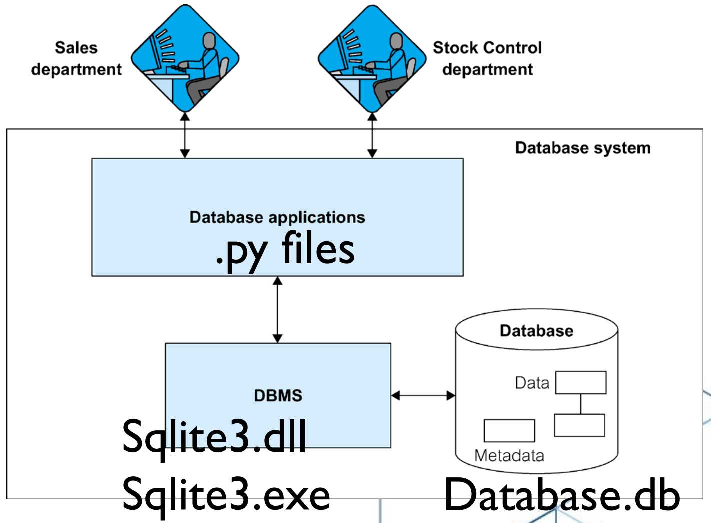
HALMSTAD UNIVERSITY

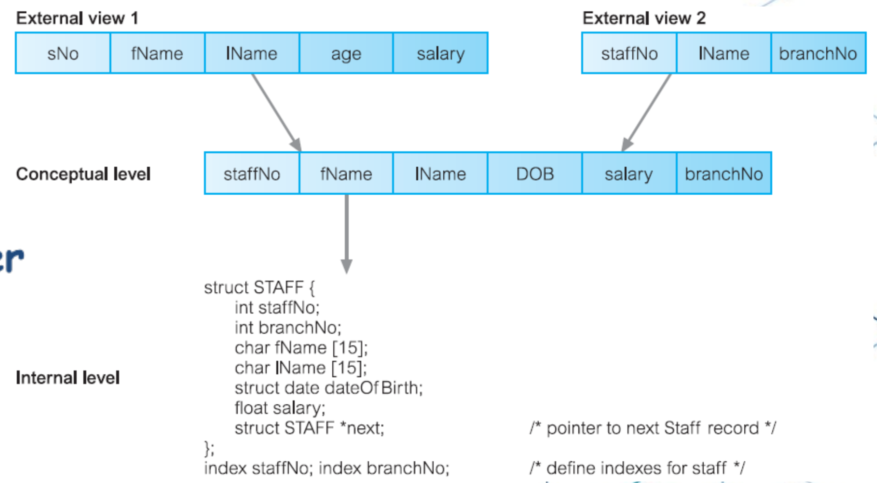
Comments or Questions?

HALMSTAD UNIVERSITY

Shortly







```
struct STAFF {
    int staffNo;
    int branchNo;
    char fName [15];
    char lName [15];
    struct date dateOfBirth;
    float salary;
    struct STAFF *next;                /* pointer to next Staff record */
};
index staffNo; index branchNo;         /* define indexes for staff */
```

HALMSTAD UNIVERSITY

Next

- Introduce and discuss database design
  - – Main stages of the database system development lifecycle
  - – Importance of database design (conceptual, logical, and physical).
  - – Introduce the concepts of the Entity–Relationship (ER) model
    - High-level conceptual data model technique in database design.
  - – Methodology for conceptual database design.
- Lab Exercise 1
  - – Available on BB. Please check!

HALMSTAD UNIVERSITY