

# Databases 2023/2024

## **Introduction**

# The menu for today

- ▶ Organisational issues
- ▶ Very short introduction to database technology
- ▶ The relational model

# Organisatie

- ▶ Hoorcolleges
  - Dinsdag 9:00 – 10:45
  - Donderdag 13:15 – 15:00
  - Laptop? Film kijken achterin
  - Clips
- ▶ Werkcolleges
  - Dinsdag: 11:00 – 12:45, start vanmiddag
  - Donderdag: 15:15 – 16:00
  - Regelmatig quizzes bij werkcollege via Teams
  - Extra online opgaven
- ▶ Materiaal via GitHub

# Organisation

- ▶ Practica (koppels)
  - Opgave 1: casusbeschrijving, modelleren, schema-ontwerp
  - Opgave 2: vulling van de database, SQL queries
  - Op zoek naar een partner? Channel *koffieautomaat* op Teams. Aanwijzingen voor registratie volgen nog.
- ▶ Huiswerkopdrachten (3x): verplicht
  - Typerend voor vragen eindtoets
  - Consequenties voor deelname herkansing

# Introduction to database technology

- ▶ What are databases?
  - Relational data model
- ▶ Why should we look at databases?
- ▶ Some aspects of database technology
  - Query languages (algebra & SQL)
  - Database applications: UI, constraints, reports
  - Domain modeling (ER-model, UML)
  - Normalization
  - Transaction processing

# What is a database?

- ▶ Example: library system
  - Books, readers, loans
  - Choosing books, returning books, searching, making reservations, subscribing readers

**Book**

<i>Bno</i>	<i>Author</i>	<i>Title</i>
327	Gates	The road ahead
535	Baars	Fun-fishing
113	Carlsen	Chess for dummies

**Reader**

<i>Rno</i>	<i>Name</i>	<i>Address</i>
212	Rutte	Torentje 1, Den Haag
431	Karjakin	Plein 2, Wladiwostok
7	Bond	Downing Str. 7, London

**Loan**

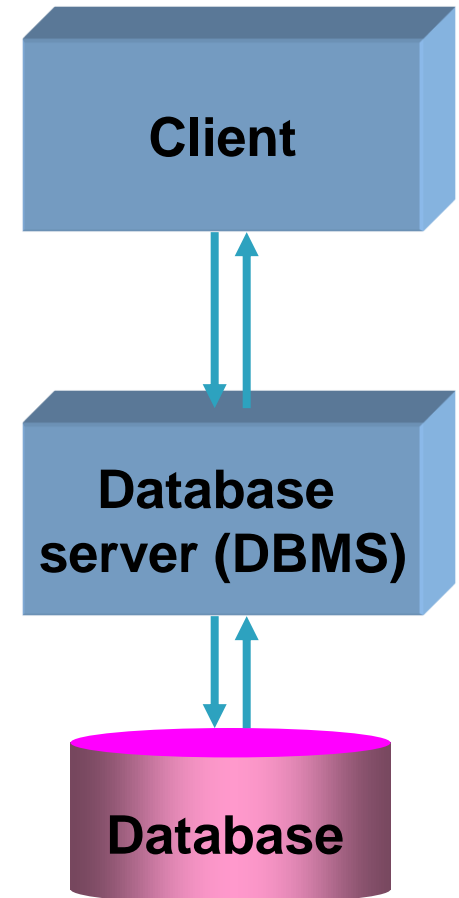
<i>Bno</i>	<i>Rno</i>	<i>Loan date</i>	<i>Return date</i>
113	431	14.10.2023	17.10.2023
327	212	21.10.2023	-
535	212	28.10.2023	-

# What is a database?

- ▶ Manipulation of data using a query language
  - For example SQL
  - Integrated in an app/ web interface

```
SELECT Title  
FROM Book  
WHERE Author = 'Rowling'
```

- ▶ Often client/server architecture
  - Application logic in the client



# What is a database?

- ▶ Characteristics of a database environment
  - Stable structure of data
    - Compared to textual data (information retrieval)
  - Large volumes (external memory, persistency)
  - Good performance
  - More than one user at a time (concurrency)
  - Reliability and integrity of data
  - Example: Amazon sells more than 400 items per second



# Why look at databases?

- ▶ Databases are omnipresent
- ▶ Database technology is directly applicable
  - Software project
- ▶ Database technology is the backbone of most information systems
- ▶ Studying database technology provides insight in general principles of computer science
  - Layered software architecture
  - Application of predicate logic
  - Mathematical modeling

# Query languages

```
SELECT Name  
FROM Book, Loan, Reader  
WHERE Book.Title = 'Fun-fishing'  
  AND Book.Bno = Loan.Bno  
  AND Loan.Rno = Reader.Rno
```

- ▶ From “how” to “what”
  - SQL is declarative

```
Book.Title := 'Fun-fishing';  
FIND FIRST Book USING Title;  
WHILE DB-Status = 0 DO  
BEGIN  
  FIND FIRST Loan WITHIN  
    Book_Loan;  
  WHILE DB-Status = 0 DO  
  BEGIN  
    FIND OWNER WITHIN  
      Reader_Loan;  
    GET Reader;  
    PRINT(Reader.Name);  
    FIND NEXT Loan WITHIN  
      Book_Loan;  
  END;  
  FIND NEXT Book USING Title;  
END
```

# Database applications (fantasy language)

```
PROCEDURE Loan ();
{
  $today = system.call('current_date');
  read($x); // read Rno

  if (Rnocheck($x) == 0)
  {
    message("card invalid");
    exit();
  };

  read($y); # read Bno
  while ($y <> EndOfLoan)
  {
    Register_loan($today, $x, $y);
    read($y);
  }
}
```

```
int Rnocheck ($x);
{
  SELECT COUNT (*)
  FROM Reader
  WHERE Rno = $x;
}
```

```
void Register_loan
  ($d, $x, $y);
{
  INSERT INTO Loan
  VALUES ($y, $x, $d, NULL);
}
```

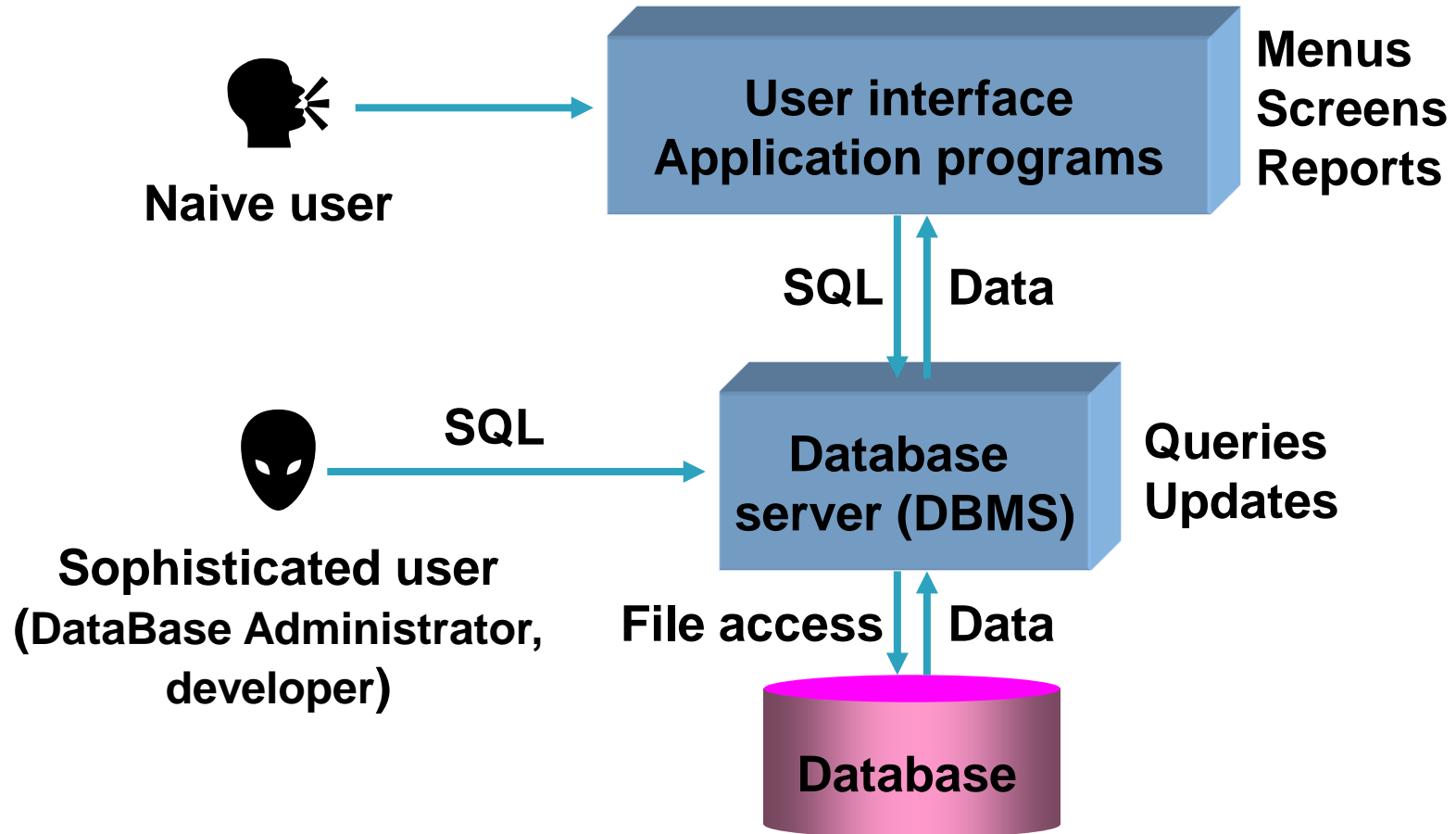
# Integrity constraints

```
CONSTRAINT constr1
(SELECT COUNT (*)
 FROM Loan
 WHERE Return_date IS NULL
 GROUP BY Rno)
<= 6
ON VIOLATION ...
```

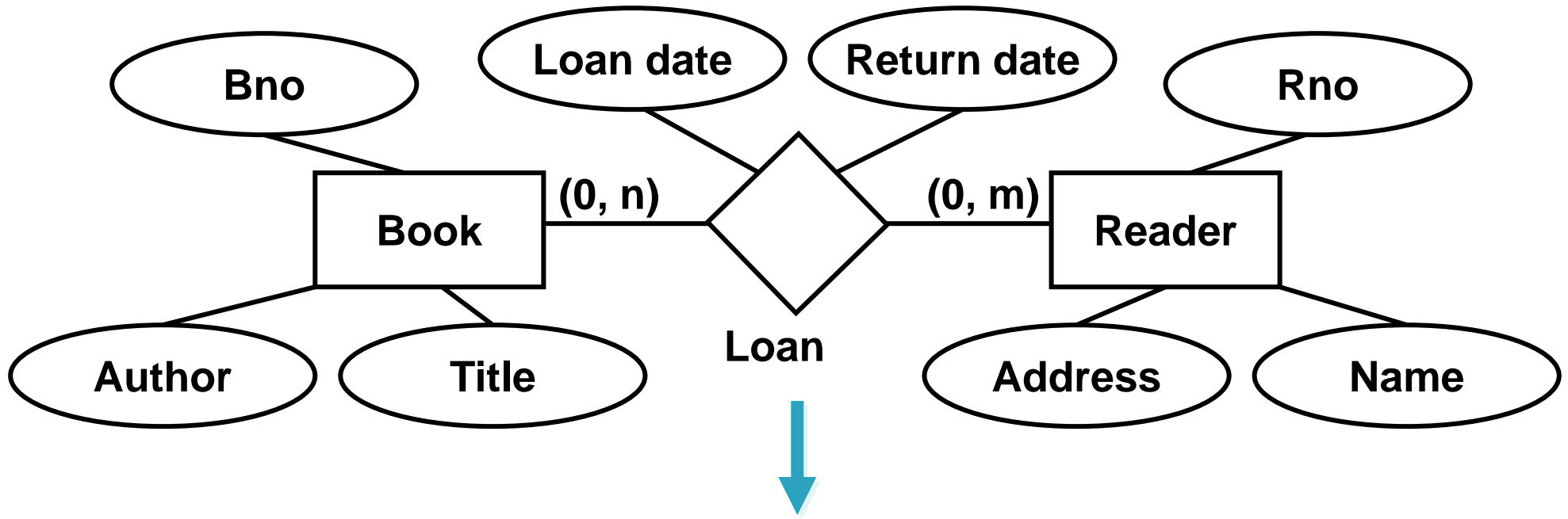
```
CONSTRAINT constr3
(SELECT Bno
 FROM Loan)
IS CONTAINED IN
(SELECT Bno
 FROM Book)
ON VIOLATION ...
```

```
CONSTRAINT constr2
(SELECT COUNT (*)
 FROM Loan
 WHERE Return_date IS NULL
 GROUP BY Bno)
<= 1
ON VIOLATION ...
```

# Database applications



# DB design: ER modeling



**Book(Bno, Author, Title)**

**Reader(Rno, Name, Address)**

**Loan(Bno, Rno, Loan\_date, Return\_date)**

# Normalization

- ▶ Why don't we put everything in one table?
  - Manageability
  - To prevent redundancy and inconsistency
  - Adequate representation (without NULLs)

<i>Rno</i>	<i>Name</i>	<i>Address</i>	<i>Bno</i>	<i>Author</i>	<i>Title</i>
212	Rutte	Torentje 1, Den Haag	327	Gates	The road ahead
212	Rutte	Torentje 2, Den Haag	535	Baars	Fun-fishing
431	Karjakin	Plein 2, Wladiwostok	113	Carlsen	Chess for dummies
7	Bond	Downing Str. 7, London	NULL	NULL	NULL

# Normalization

<i>Rno</i>	<i>Name</i>	<i>Address</i>	<i>Bno</i>	<i>Author</i>	<i>Title</i>
212	Rutte	Torentje 1, Den Haag	327	Gates	The road ahead
212	Rutte	Torentje 1, Den Haag	535	Baars	Fun-fishing
431	Kramnik	Plein 2, Wladiwostok	113	Kasparov	Chess for dummies
7	Bond	Downing Str. 7, London	NULL	NULL	NULL

<i>Rno</i>	<i>Name</i>	<i>Address</i>	<i>Bno</i>	<i>Author</i>	<i>Title</i>
212	Rutte	Torentje 1, Den Haag	327	Gates	The road ahead
431	Kramnik	Plein 2, Wladiwostok	535	Baars	Fun-fishing
7	Bond	Downing Str. 7, London	113	Kasparov	Chess for dummies

<i>Bno</i>	<i>Rno</i>	<i>Loan_date</i>	<i>Return_date</i>
113	431	14.10.2022	17.11.2022
327	212	21.10.2022	NULL
535	212	28.10.2022	NULL



# Transaction processing


- ▶ Transactions are important in case of crashes and simultaneous use of the database by multiple users
  - In case of a crash, no partial results of a transaction should be visible: *all or nothing*

**Read balance accno. 1234567**  
**Read balance accno. 7654321**  
**Withdraw € 50,- from 1234567**  
**Deposit € 50,- on 7654321**  
**Write balance accno. 1234567**  
**Write balance accno. 7654321**

# Transaction processing

- ▶ Transactions are important in case of crashes and simultaneous use of the database by multiple users
  - In case of a crash, no partial results of a transaction should be visible: *all or nothing*

**CRASH!**



**Read balance accno. 1234567**  
**Read balance accno. 7654321**  
**Withdraw € 100,- from 1234567**  
**Deposit € 100,- on 7654321**  
**Write balance accno. 1234567**  
**Write balance accno. 7654321**

# Transaction processing

<b>1. Read balance accno. 1234567</b>
<b>2. Read balance accno. 1234567</b>
<b>1. Withdraw € 500,- from balance</b>
<b>2. Withdraw € 500,- from balance</b>
<b>1. Write balance accno. 1234567</b>
<b>2. Write balance accno. 1234567</b>

- ▶ Concurrency problem
- ▶ Solved by locking based techniques

# Why relational databases?

- ▶ Software Engineering
  - High level data specification and manipulation
  - Black box behaviour
- ▶ Philosophy with regard to data oriented system development
  - Start with rigorous design of database structure
    - Stable; detailed assessment is possible
  - Development of operations is secondary
    - Difficult to analyze, rapid prototyping, continuous adaptation
- ▶ Successful application of computer science
  - Set theory, predicate logic, optimization, design theory