

# Databases 2025/2026

## Introduction

# The menu for today

- ▶ Organizational aspects
- ▶ Introduction to database technology
- ▶ The relational model

# Organisatie

## ▶ Hoorcolleges

- Dinsdag 9:00 – 10:45
- Donderdag 13:15 – 15:00
- Wooclap quizzes: links verschijnen in Teams

## ▶ Werkcolleges (verplicht)

- Dinsdag: 11:00 – 12:45
- Donderdag 15:15 – 17:00
- Start: donderdag a.s.
- Uitwerkingen verschijnen met een vertraging van één tot anderhalve week in Teams

# Organisatie

- ▶ Practica (koppels)
  - Opgave 1: case description, modeling, schema design
  - Opgave 2: reconsider your design using AI
- ▶ Huiswerkopdrachten (3x): verplicht
  - Typerend voor vragen eindtoets
  - Zeer sterke correlatie tussen het niet maken of niet serieus nemen van de huiswerkopdrachten en slechte resultaten voor de eindtoets

# 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
  - Query processing

# What is a database?

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

Book

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

Reader

Rno	Name	Address
212	Schoof	Torentje 1, Den Haag
431	Niemann	Avenue 2, Somewhere
7	Bond	Downing Str. 7, London

Loan

Bno	Rno	Loan date	Return date
113	431	14.10.2025	17.10.2025
327	212	21.10.2025	-
535	212	28.10.2025	-

# 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
  - Mathematical modeling
  - Application of predicate logic and set theory

# History of databases

- ▶ During the eighties, the relational data model (Codd, Turing Award 1981) received widespread commercial attention
  - In 1983, more than 100 RDBMSes existed
  - DB2, ORACLE, SYBASE, INFORMIX, INGRES
  - DBASE, PARADOX, MS-ACCESS
  - POSTGRES, MySQL, SQLite
  - *NoSQL*: MongoDB, MapReduce, GraphDBs
- ▶ SQL became a “standard” in 1986
- ▶ SQL92/SQL2, SQL3: ANSI standards

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

    read($y); # read Bno
    while ($y <> EndOfLoan)
    {
        call(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);
}
```

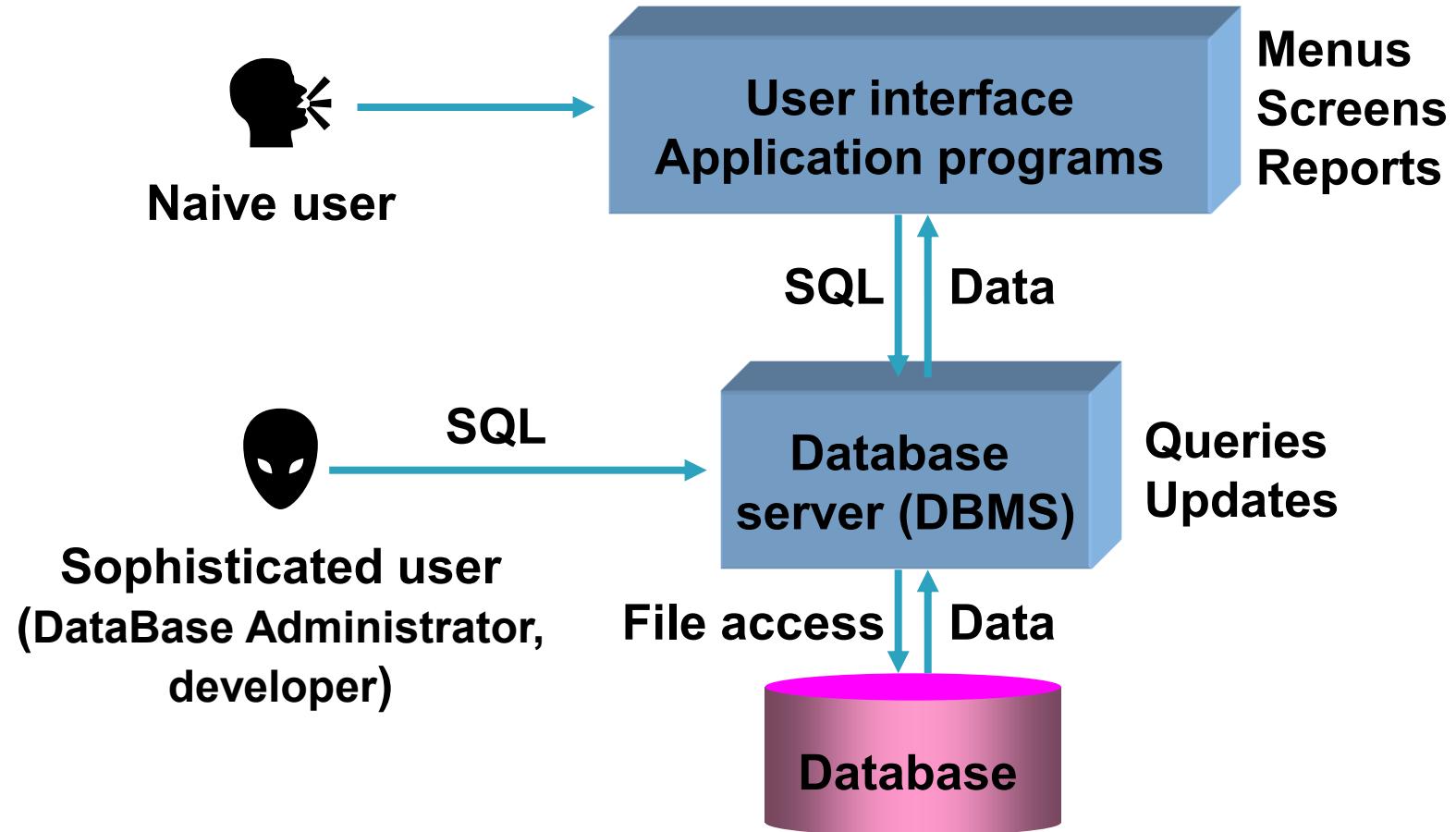
# Integrity constraints

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

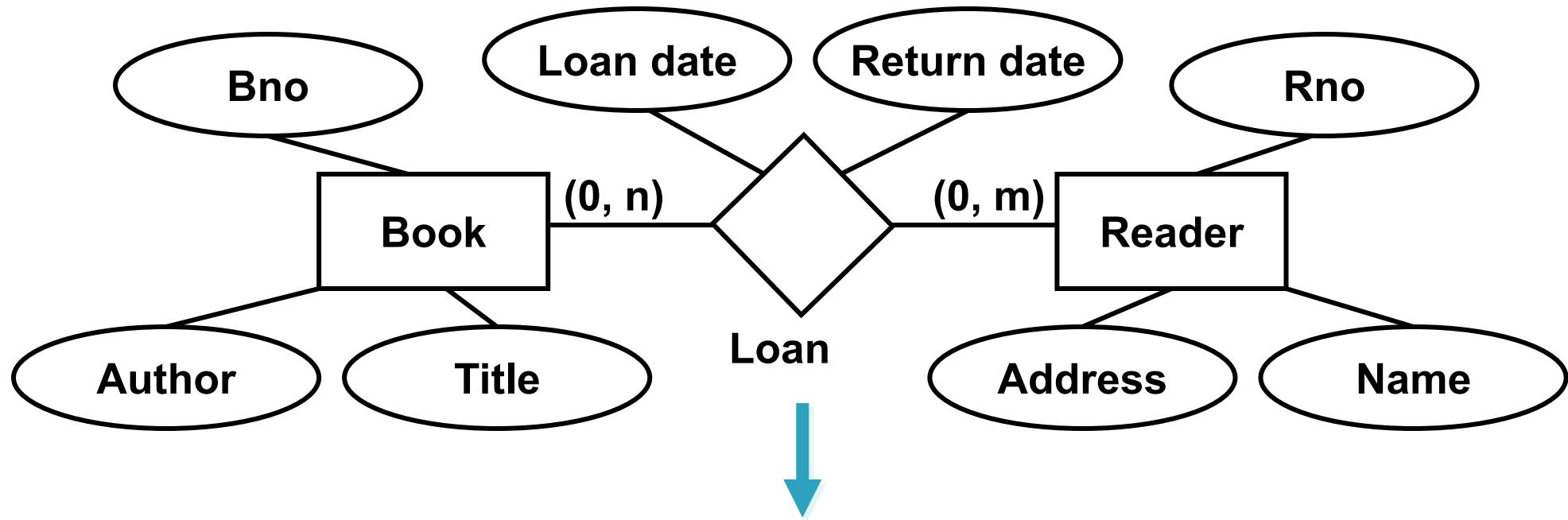
```
-- CONSTRAINT constr3  
(SELECT Bno  
  FROM Loan)  
 IS CONTAINED IN  
(SELECT Bno  
  FROM Book)  
-- ON VIOLATION ...
```

```
-- 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 (avoiding NULLs)

Rno	Name	Address	Bno	Author	Title
212	Schoof	Torentje 1, Den Haag	327	Gates	The road ahead
212	Schoof	Torentje 2, Den Haag	535	Baars	Fun-fishing
431	Niemann	Avenue 2, Somewhere	113	Carlsen	Chess for dummies
7	Bond	Downing Str. 7, London	NULL	NULL	NULL

# Normalization

Rno	Name	Address	Bno	Author	Title
212	Schoof	Torentje 1, Den Haag	327	Gates	The road ahead
212	Schoof	Torentje 1, Den Haag	535	Baars	Fun-fishing
431	Niemann	Avenue2, Somewhere	113	Kasparov	Chess for dummies
7	Bond	Downing Str. 7, London	NULL	NULL	NULL

Rno	Name	Address	Bno	Author	Title
212	Schoof	Torentje 1, Den Haag	327	Gates	The road ahead
431	Niemann	Avenue2, Somewhere	535	Baars	Fun-fishing
7	Bond	Downing Str. 7, London	113	Kasparov	Chess for dummies

Bno	Rno	Loan_date	Return_date
113	431	14.10.2015	17.11.2015
327	212	21.10.2015	NULL
535	212	28.10.2015	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!

The diagram illustrates a transaction log enclosed in a blue border. On the left, a teal arrow points from a box labeled "CRASH!" towards the log. The log contains the following entries:  
**Read balance accno. 1234567**  
**Read balance accno. 7654321**  
**Withdraw € 50,- from 1234567**  
**Deposit € 50,- on 7654321**  
**Write balance accno. 1234567**  
**Write balance accno. 7654321**

**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

1. Read balance accno. 1234567
  2. Read balance accno. 1234567
1. Withdraw € 500,- from balance
  2. Withdraw € 500,- from balance
1. Write balance accno. 1234567
  2. Write balance accno. 1234567

- ▶ Concurrency problem
- ▶ Solved by locking based techniques

# Why relational databases?

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