Databases 2022/2023

Introduction

The menu for today

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

Organisation

- Hoorcolleges (classroom flipped)
  - Dinsdag 9:00 – 10:45 *online*
  - Donderdag 13:15 – 15:00 *on campus*
- Werkcolleges
  - Dinsdag: 11:00 – 12:45 start volgende week, *online*
  - Donderdag: 15:00 – 17:00 *on campus*
  - *parallel online via QA sessions Teams*
- Practica
  - Assistentie via werkcollege

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

Komende tijd: de cyclus

- Nu: hoorcollege
- Straks: nog geen werkcollege (begint 15 feb)
- Voor donderdagmiddag: bekijk clips RA1, RA2
- Voor donderdagmiddag: vragen over clips RA? Mail deze naar Hans
- Donderdag (13-15): ER-modelering
- Donderdag (13-15): reflectiecollege over clips
- Donderdag (15-17): werkcollege
- Voor dinsdagochtend: bekijk clip FD1
- Vragen over clip FD1? Mail deze naar Hans
- ….

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, 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	Rutte	Torentje 1, Den Haag
431	Karjakin	Plein 2, Wladiwostok
7	Bond	Downing Str. 7, London

Loan

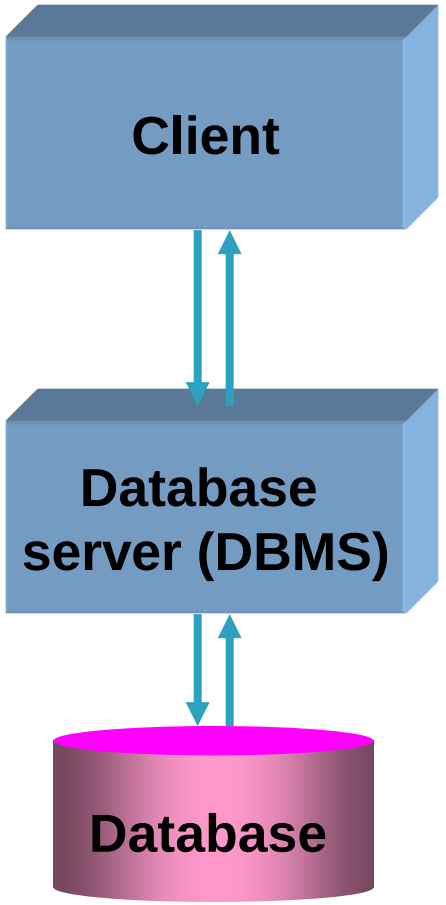
Bno	*Rno*	*Loan date*	*Return date*
113	431	14.10.2019	17.10.2019
327	212	21.10.2019	-
535	212	28.10.2019	-

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
    - Compare 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

History of databases

- During the eighties, the relational data model (Codd, Turing Award 1981) received widespread commercial attention
  - In 1983, more than 100 **R**DBMSes 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 (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);
}
```

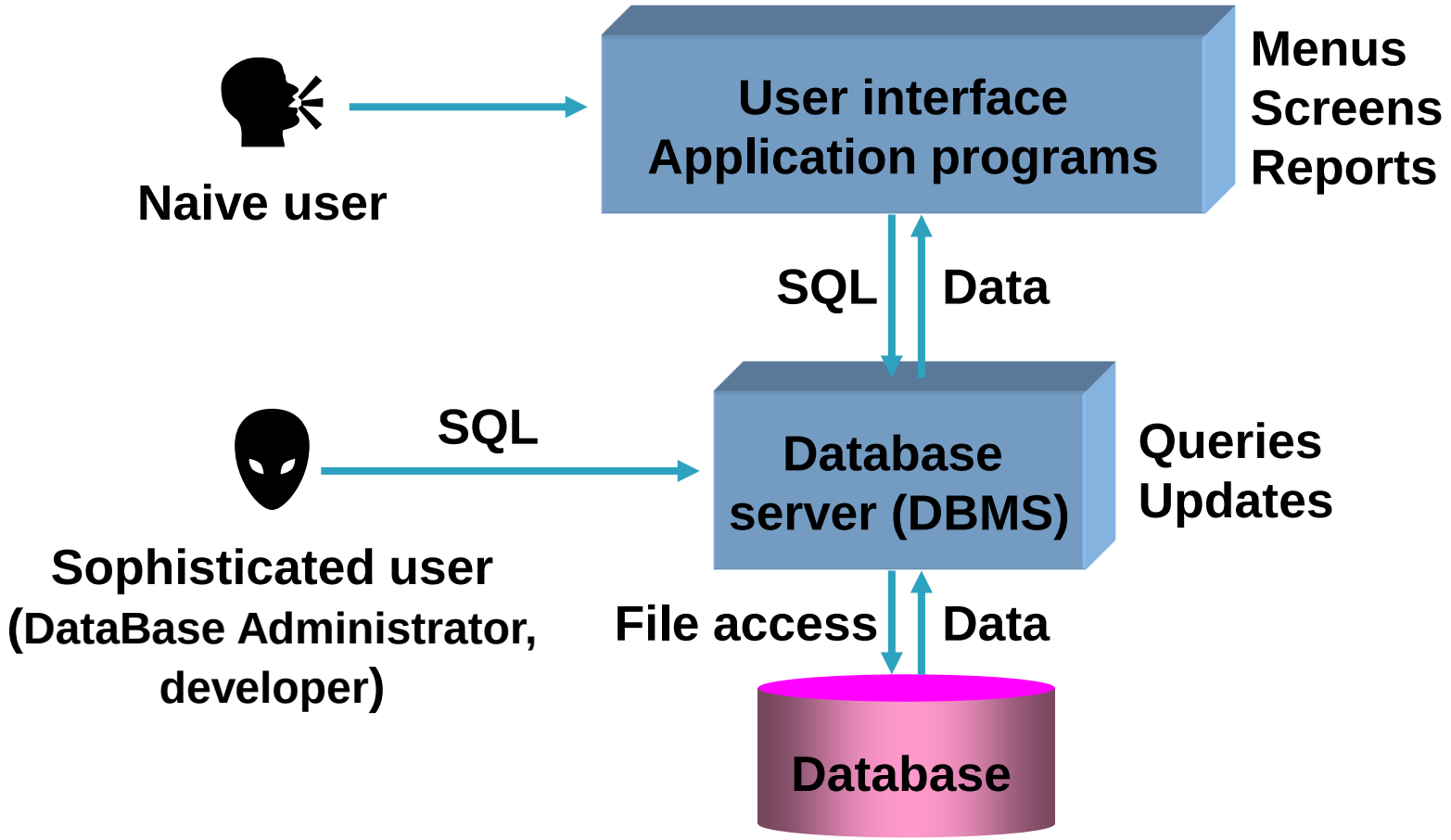
Integrity constraints

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

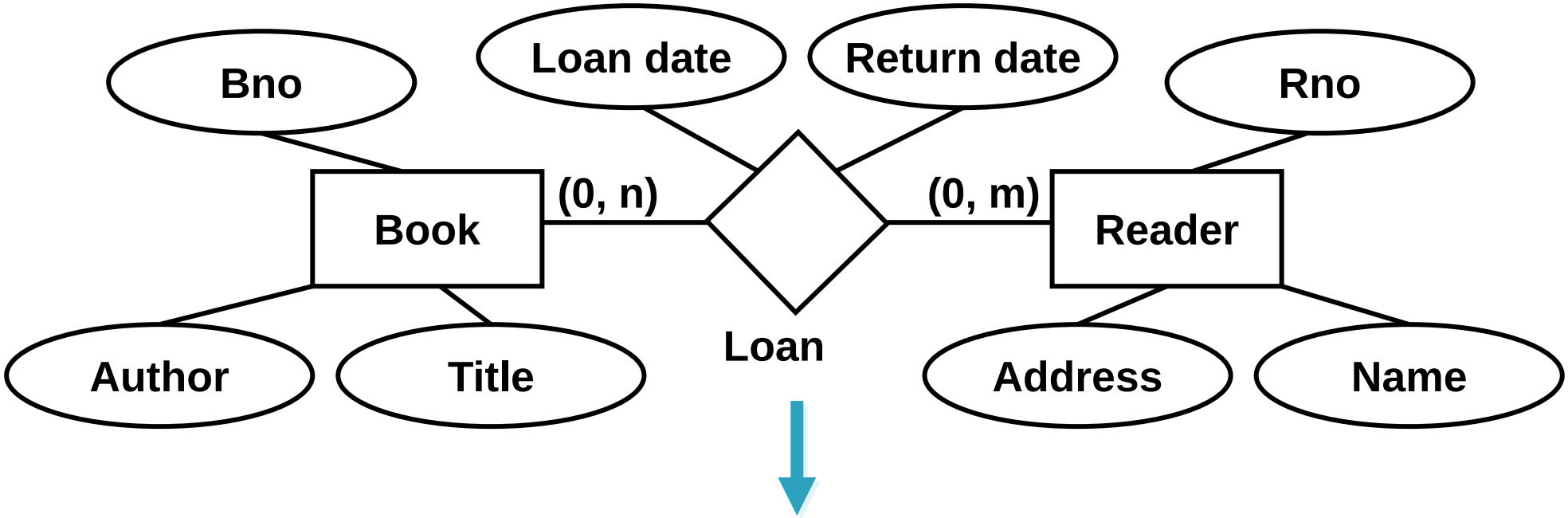
```
CONSTRAINT constr2
  (SELECT COUNT (*)
   FROM Loan
   WHERE Return_date IS NULL
   GROUP BY Bno)
  <= 1
ON VIOLATION …
```

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

Rno	*Name*	*Address*	*Bno*	*Author*	*Title*
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

Rno	Name	Address	Bno	Author	Title
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

Rno	Name	Address
212	Rutte	Torentje 1, Den Haag
431	Kramnik	Plein 2, Wladiwostok
7	Bond	Downing Str. 7, London

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

Bno	Rno	Loan_date	Return_date
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*

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 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