Introduction to Database Systems
IDBS - Fall 2024

Lecture 1 - Introduction

Course Introduction
DBMS Introduction
Relational Data Model
SQL DDL

Readings: PDBM 1, 6.1, 7.1-7.2

Omar Shahbaz Khan

IT UNIVERSITY OF COPENHAGEN

Course Responsible

Eleni Tzirita Zacharatou

2013

MSc in Electrical & Computer Engineering
NTUA, Greece

2013 - 2019

Ph.D. in CS
EPFL, Switzerland

2016

Visiting Researcher
NYU, USA

2019 - 2022

Postdoctoral Research
TUB, Germany

2022 - Now

Assistant Professor
ITU, Denmark

Website: https://heltzi.github.io

Lecturer

Omar Shahbaz Khan

2013 - 2022

BSc (SWU), MSc (CS), PhD in CS
ITU

2022 - Now

Postdoctoral Researcher,
Reykjavik University,
ITU (20%)

Website: oskhan.com

Guest Lecturer

Martin Hentschel

- BSc, MSc, PhD in CS from ETH Zurich, Switzerland
- Microsoft
- Snowflake
- ITU

- Research Interests: database systems, open data and metadata formats, and data management in general

- https://dasya.itu.dk/people/martinhentschel/

Teaching Assistants

Anders Arvesen
Study program: MSc. Computer Science

Adam Hadou Temsamani
Study program: MSc. Computer Science

Anne-Marie Rommerdahl
Study program: BSc. Software Development

Cristina Avram
Study program: BSc. Data Science

Erling Amundsen
Study program: MSc. Data Science

Marcus Henrik Simonsen
Study program: MSc. Computer Science

Oliver Flyckt Wilhjelm
Study program: MSc. Games (Technology)

Philip Kristian Møller Flyvholm
Study program: MSc. Computer Science

Research Group

15+ People

www.dasya.itu.dk

DASYA

Data-Intensive Systems and Applications

Entire Data Lifecycle:

- Collection
- Transfer
- Storage
- Curation
- Processing
- Analytics

Lab for Research and Education:
5th floor (5A56)

Infrastructure for Data Science

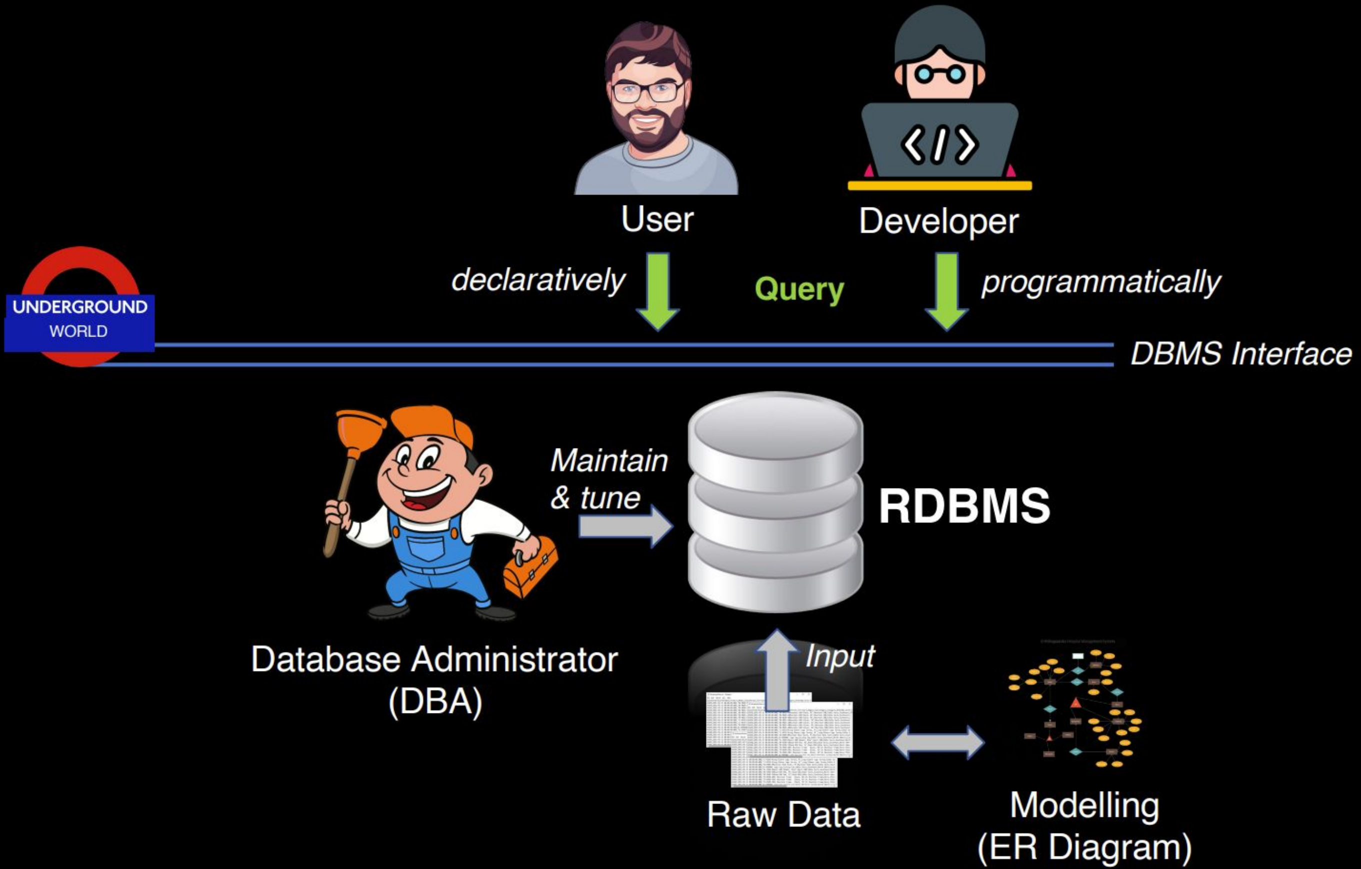
But this about you!

- ~250 Students
  - Mostly from BSc. in Software Development

- Too many for a round table

- What do you expect from Introduction to Database Systems?
  - Mentimeter: www.menti.com | Code: 2244 4401

What you will learn



Intended Learning Outcomes

- Write SQL queries: multiple relations; compound conditions; grouping; aggregation; and subqueries.
- Use relational DBMSs from a conventional programming language in a secure manner
- Suggest a database design in the ER model and convert to a relational database schema in a suitable normal form
- Analyze/predict/improve query processing efficiency of the designed database using indices
- Reflect upon the evolution of the hardware and storage hierarchy and its impact on data management system design.
- Discuss the pros and cons of different classes of data systems for modern analytics and data science applications.

Intended Learning Outcomes

- Write SQL queries: multiple relations; compound conditions; grouping; aggregation; and subqueries.
- Use r … man …
- Sugg … ase sche …
- Anal … se using indic …
- Reflec … pact on data management system design.
- Discuss the pros and cons of different classes of data systems for modern analytics and data science applications.

1. Getting into Database Systems
2. Getting data using SQL and from your apps
3. Design a database
4. Tune a database
5. Advanced databases (internals and big data)

Course Schedule

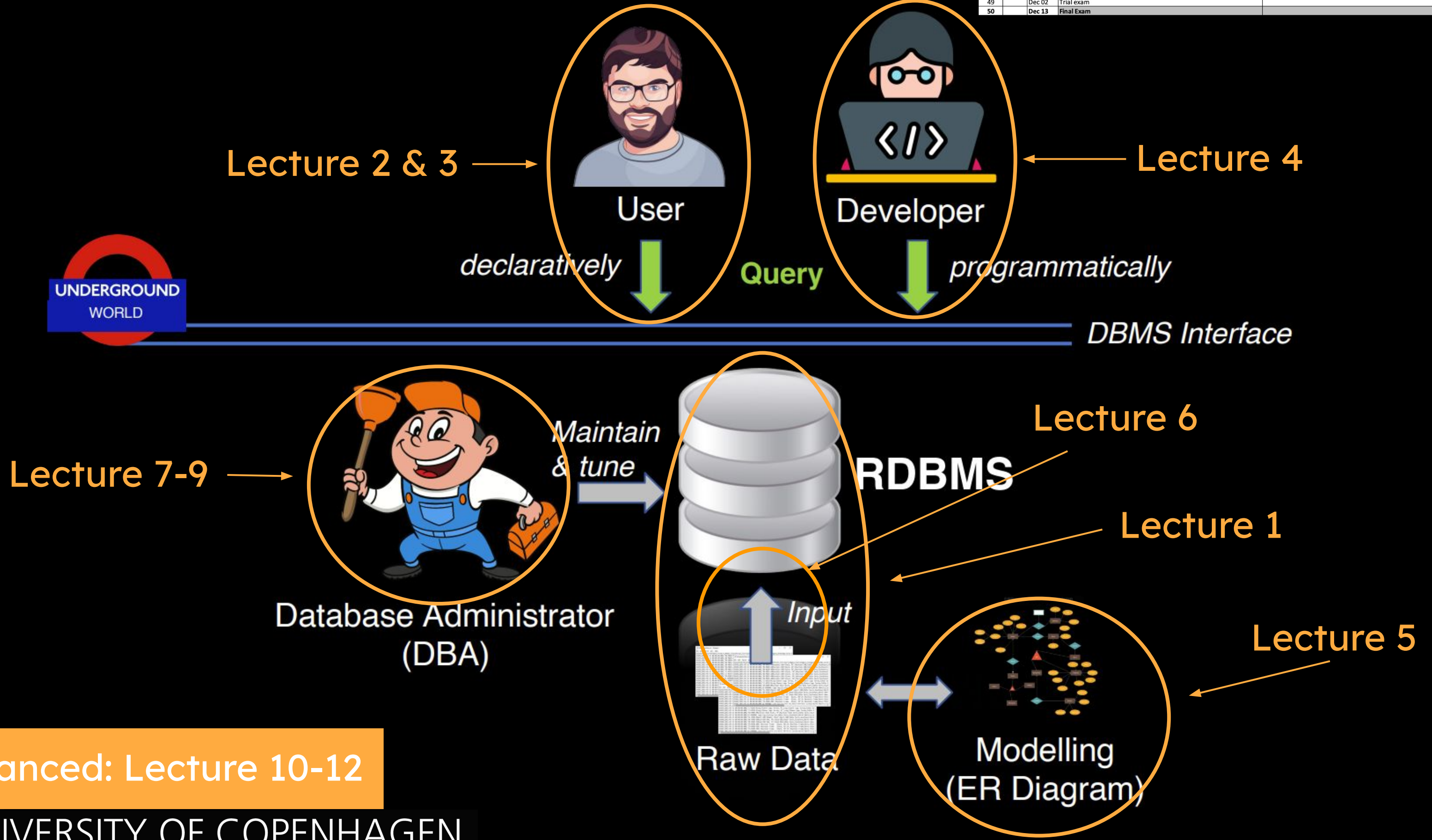
Week	Lecture	Date	Lecture Topic	Readings	Exercises / Homeworks	Notes
36	1	Sep 2	Introduction; Course overview; Relational model	PDBM: 1, 6.1, 7.1-7.2	Exercise L1	Lecturer: Omar Shahbaz Khan
37	2	Sep 09	SQL DDL/DML; Basic SQL queries; Joins; Aggregations	PDBM: 7.3	Exercise L2	Lecturer: Eleni Tzirita Zacharatou
38	3	Sep 16	Complex SQL queries; Subqueries; Views	PDBM: 7.3 - 7.4	Homework 1 (due on 23.09.2024)	Lecturer: Eleni Tzirita Zacharatou
39	4	Sep 23	Triggers; Transactions; Using SQL from Python	PDBM: 9.2, 14.1, 14.2.1, 14.5 + more (details below)	Exercise L4	Lecturer: Omar Shahbaz Khan
40	5	Sep 30	ER modeling	PDBM: 3.0-3.3, 6.3-6.4	Exercise L5	Lecturer: Omar Shahbaz Khan
41	6	Oct 07	Normalization	PDBM: 6.2-6.4	Homework 2 (due on 14.10.2024)	Lecturer: Eleni Tzirita Zacharatou
42		Oct 14	**Semester Break**			
43	7	Oct 21	Storage hierarchy; Physical database design; Indexing	PDBM: 12.1, 12.2, 12.3.1 - 12.3.7 + more (details below)	Exercise L7	Lecturer: Eleni Tzirita Zacharatou
44	8	Oct 28	B-trees; Performance tuning; Access methods; Join implementation	PDBM: 12.3.8, 13.1	Exercise L8	Lecturer: Omar Shahbaz Khan
45	9	Nov 04	Storage Models; Architecture of a DBMS; Main memory DBMSs	PDBM: 2 + more (details below)	Homework 3 (due on 11.11.2024)	Lecturer: Martin Hentschel
46	10	Nov 11	Transaction Management; Transactions in main memory DBMSs	PDBM: 14 + more (details below)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
47	11	Nov 18	Scaling-out; NoSQL; Eventual consistency; CAP theorem	PDBM: 11 (+ optional papers)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
48	12	Nov 25	Big data analytics; Distributed computing frameworks	PDBM: 19.1-19.2, 19.4, 20.1-20.3	Homework 4 (due on 02.12.2024)	Lecturer: Martin Hentschel
49		Dec 02	Trial exam			Online
50		**Dec 13**	**Final Exam**			

- BSc. vs MSc.
  - Pretty much the same schedule
  - One different question (5%) on the exam

The Course

Course Schedule (Illustrated)

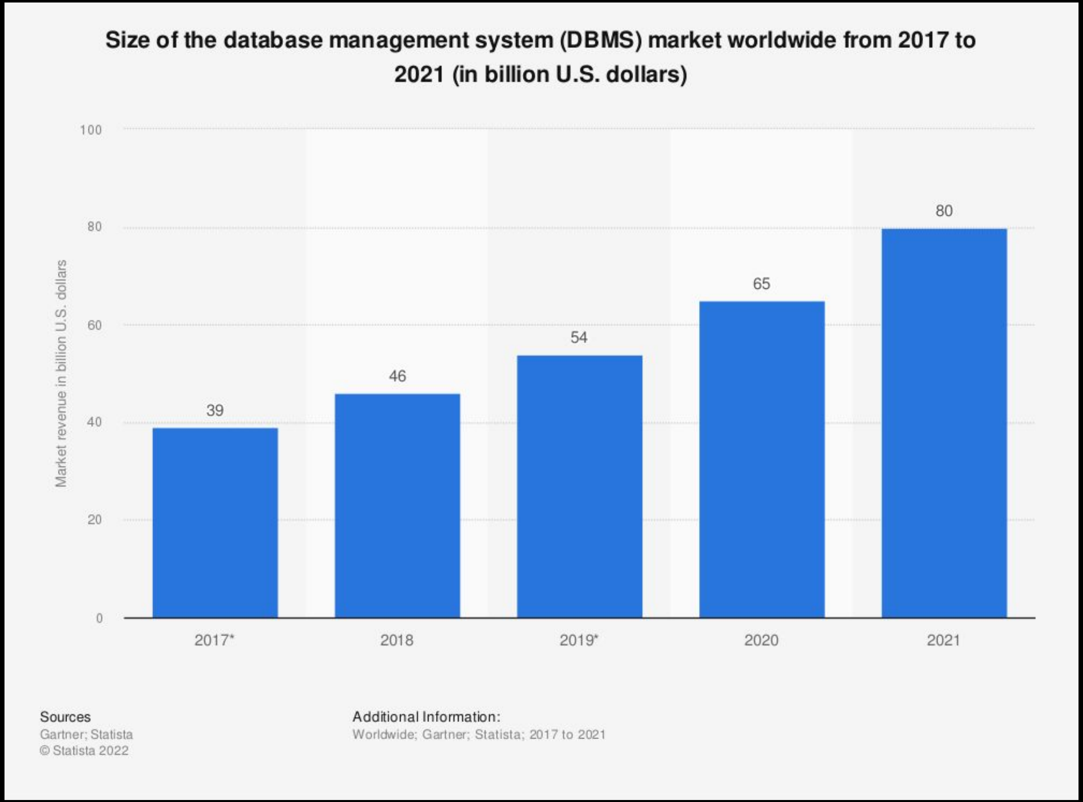
Week	Lecture	Date	Lecture Topic	Readings	Exercises / Homeworks	Notes
36	1	Sep 2	Introduction; Course overview; Relational model	PDBM: 1, 6.1, 7.1-7.2	Exercise L1	Lecturer: Omar Shahbaz Khan
37	2	Sep 09	SQL DDL/DML; Basic SQL queries; Joins; Aggregations	PDBM: 7.3	Exercise L2	Lecturer: Eleni Tzirita Zacharatou
38	3	Sep 16	Complex SQL queries; Subqueries; Views	PDBM: 7.3 - 7.4	Homework 1 (due on 23.09.2024)	Lecturer: Eleni Tzirita Zacharatou
39	4	Sep 23	Triggers; Transactions; Using SQL from Python	PDBM: 9.2, 14.1, 14.2.1, 14.5 + more (details below)	Exercise L4	Lecturer: Omar Shahbaz Khan
40	5	Sep 30	ER modeling	PDBM: 3.0-3.3, 6.3-6.4	Exercise L5	Lecturer: Omar Shahbaz Khan
41	6	Oct 07	Normalization	PDBM: 6.2-6.4	Homework 2 (due on 14.10.2024)	Lecturer: Eleni Tzirita Zacharatou
42		Oct 14	**Semester Break**			
43	7	Oct 21	Storage hierarchy; Physical database design; Indexing	PDBM: 12.1, 12.2, 12.3.1 - 12.3.7 + more (details below)	Exercise L7	Lecturer: Eleni Tzirita Zacharatou
44	8	Oct 28	B-trees; Performance tuning; Access methods; Join implementation	PDBM: 12.3.8, 13.1	Exercise L8	Lecturer: Omar Shahbaz Khan
45	9	Nov 04	Storage Models; Architecture of a DBMS; Main memory DBMSs	PDBM: 2 + more (details below)	Homework 3 (due on 11.11.2024)	Lecturer: Martin Hentschel
46	10	Nov 11	Transaction Management; Transactions in main memory DBMSs	PDBM: 14 + more (details below)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
47	11	Nov 18	Scaling-out; NoSQL; Eventual consistency; CAP theorem	PDBM: 11 (+ optional papers)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
48	12	Nov 25	Big data analytics; Distributed computing frameworks	PDBM: 19.1-19.2, 19.4, 20.1-20.3	Homework 4 (due on 02.12.2024)	Lecturer: Martin Hentschel
49		Dec 02	Trial exam			Online
50		**Dec 13**	**Final Exam**			



Advanced: Lecture 10-12

Why is it important?

- Crucial to effectively manage and utilize data
- Help maintain data integrity and security
- Ease app development



https://www.statista.com/statistics/724611/worldwide-database-market/

Course Structure

- Lectures
  - Mondays 12:15 - 14:00 in Aud 1
  - *Preparation required: reading material, watching videos*

- Exercises
  - Mondays 14:15 - 16:00 in 3A54, 3A52, and 4A14-16.
  - If all rooms are full we also have 3A18.
  - *Preparation not required (related to previous lecture)*

- Homeworks
  - 4 Homeworks (deadlines published on learnIT)
  - *Mandatory (3 out of 4), yet easy to get accepted!*
  - *Feedback in the following weeks (if submitted on time)*

- LearnIT: course outline, materials, announcements, …

- Piazza: Q&A, messages, and updates
  - *Ask consistently throughout the semester*
  - *Help your peers!*

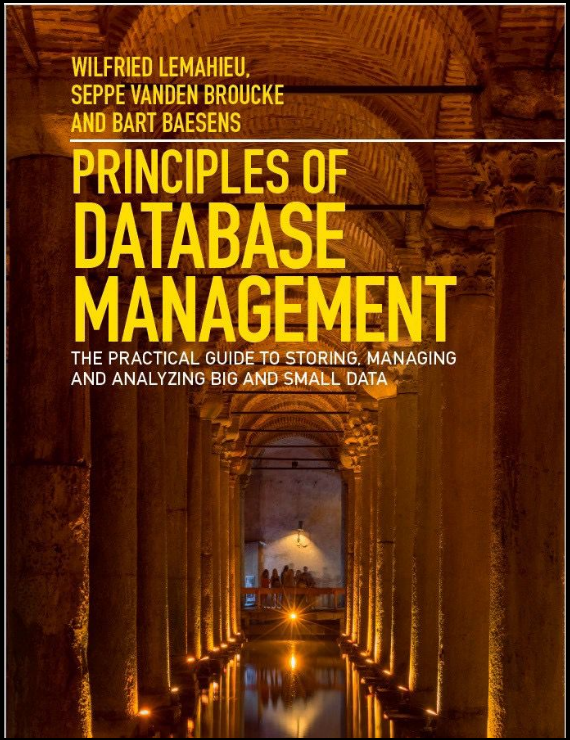
Course Methodology

- You learn: We are here to help you!

- You need to read the book beforehand
  - Yes, we often assume you have done so
  - All readings are in the schedule on learnIT
  - In some weeks, there can be some video recording for you to watch *before* the lecture

- We work in a pull model fashion: ask questions!

Book and Database System



PostgreSQL
(pgAdmin + psql)

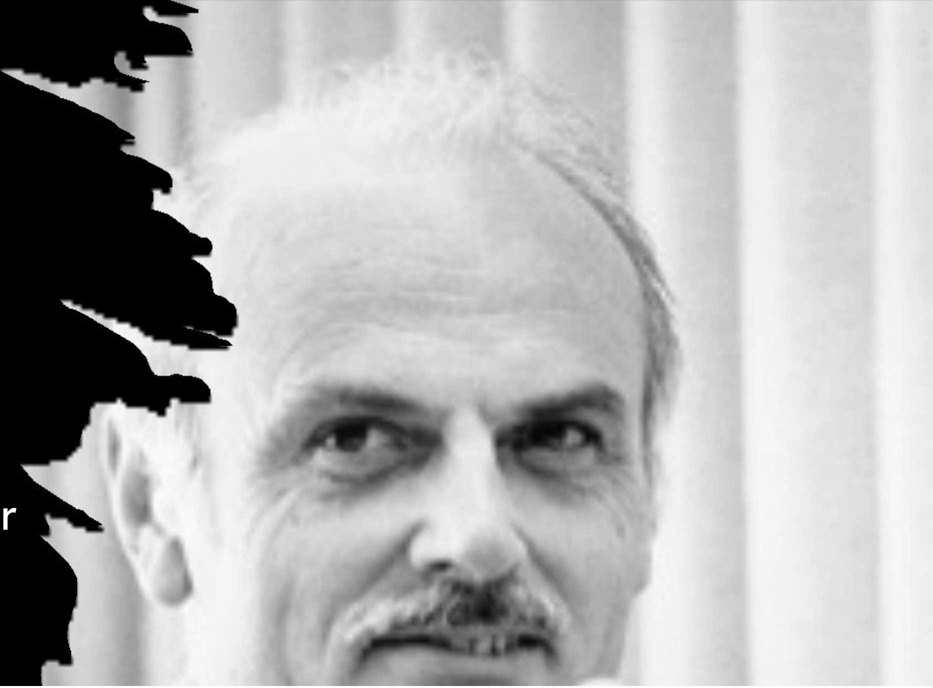
How will we assess your learning?

- 100% Exam (Quiz on LearnIT)
  - Restricted: no Internet access!
  - All course materials allowed offline
  - Communication is not allowed

- Exercises and homeworks will prepare you!

Profile of the Week

Edgar F. Codd



Father of Databases (*Relational Model*)

- 1923: Born 23/8, Isle of Portland, England
- 1965: PhD in CS from University of Michigan
- 1967: Moved to IBM Almaden Research Center
- 1969: Invented the relational model
- 1976: IBM Fellow
- 1981: Turing Award
- 1994: ACM Fellow

Information Retrieval P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

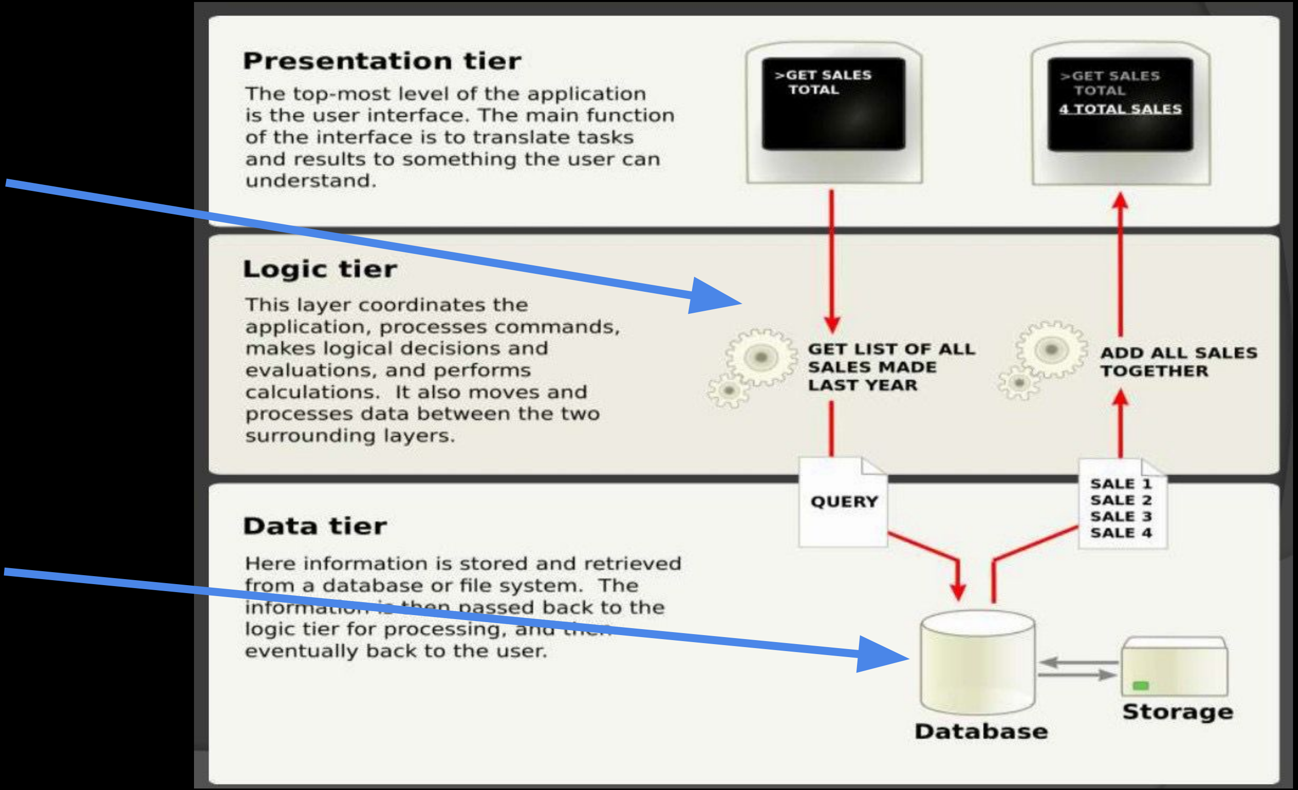
The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representa-

DBMS Brief Introduction

Three-Layer Architecture

pgAdmin + psql

PostgreSQL

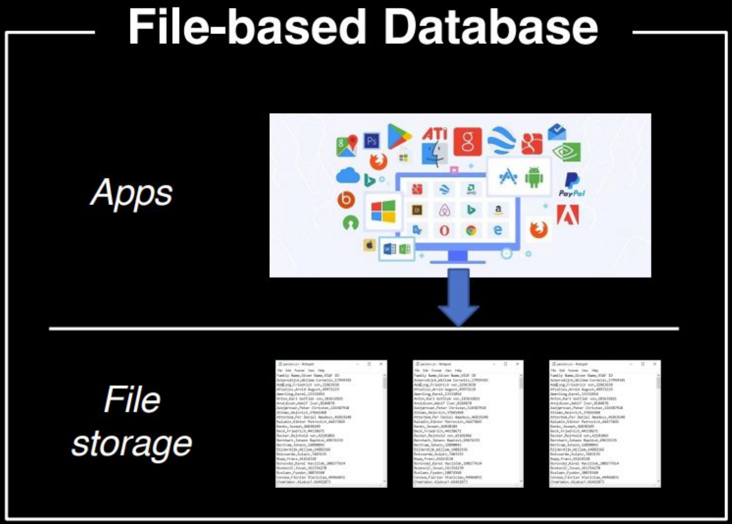


https://www.slideshare.net/shubhamdwivedi3939/dbms-architecture

Database Definition

- A database is a collection of related data items within a specific business process or problem setting
- A database system provides a way to systematically organize, store, retrieve, and query a database

- Cons:
  - Redundant data
  - Inconsistent data
  - **Data dependency**
  - Limited concurrency
  - Expensive maintenance
  - Poor performance



Relational Database Definition

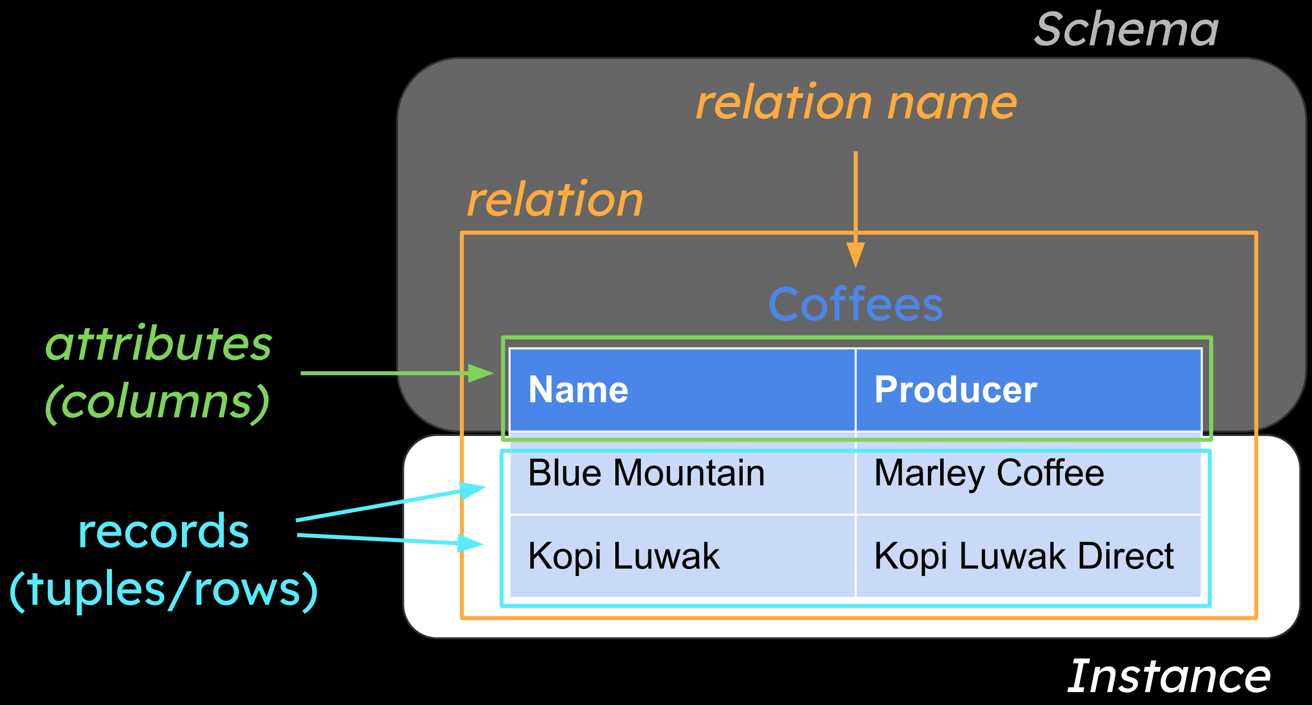
- A relational database is a type of database that is based on the relational model
  - stores data in a set of tables with rows and columns (a.k.a. relations)
  - uses relationships between these tables to manage the data.

- A relational database system (RDBMS) implements and manages relational databases.

- Pros:
  - Unique data entities
  - Data integrity
  - **Data independence**
  - High concurrency
  - Cheap maintenance
  - High performance

Relational Model

Basic Concepts



Schema vs Instance vs Database

- Relation
  - Schema = name + list of attributes
    - Optional: attribute types
    - **Coffees** (Name, Producer)
    - **Coffees** (Name : STRING, Producer : STRING)
  - Instance
    - Records in a relation
      - E.g., (Blue Mountain, Marley Coffee)

- A Database is a collection of relations
  - Database schema = set of all relations names in the database
  - Database instance = set of all relations instances in the database

Example of a Database Schema

- **Students** (SID : INT, Name : STRING, Email : STRING, Semester : INT)
- **Faculty** (FID : INT, Name : STRING, DID : INT)
- **Courses** (CID : STRING, Name : STRING, DID : INT)
- **Departments** (DID : INT, Name : STRING)
- **Transcripts** (CID : STRING, SID : INT, Grade : STRING, Comment : STRING)

Why Relations?

Pros

- Very simple model
- How we typically think about structured data
- Conceptual model behind SQL, which is the most important query language today

Cons

- Too simple for some things:
  - hierarchical multi-valued data
- Much of today's data is NOT structured
- It could be complex to implement well

Identifiers

- **Students** (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)
- **Faculty** (<u>FID</u> : INT, Name : STRING, DID : INT)
- **Courses** (<u>CID</u> : STRING, Name : STRING, DID : INT)
- **Departments** (<u>DID</u> : INT, Name : STRING)
- **Transcripts** (CID : STRING, SID : INT, Grade : STRING, Comment : STRING)

Keys and Superkeys

- What is a key?
  - Defines unique records (instances)
  - Helps in setting relationships between relations
  - Ensures the mathematical definition of a relation (*set* of records)

- Superkeys
  - Is a set of attributes that uniquely identifies records: *Uniqueness* property
  - The entire set of attributes of a relation is a superkey
  - Minimal superkey: *Minimality* property
    - No attributes can be removed from a superkey without violating the uniqueness property

key

superkey

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5

Candidate Keys

- Attributes that satisfy the uniqueness and minimality properties
  - Minimal superkey = (candidate) key
  - Superkeys contain at least one (candidate) key
  - A relation can have many (candidate) keys

key

superkey

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5

Primary Keys

- A key to identify records in a relation
  - Important to define indexes and for storage purposes (later in the course)
  - Cannot be null
  - Also used to establish relationships with other relations
  - From all candidate keys only one can be the Primary Key
    - The remaining ones are known as *Alternative* Keys

key

superkey

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5

Live Exercise

- What are superkeys and (candidate) keys?
  - (SID)
  - (Email)
  - (SID, Name)
  - (Semester)
  - (Email, Semester)
  - (Name)
  - (Name, Semester)
- What is the best key for being the primary key?
- Which of these keys does <u>not</u> make sense in practice?

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
04321	Jens Schuh	jesc@itu.dk	1
01439	Olga Marx	olmr@itu.dk	2
01832	Peter Pitt	pepi@itu.dk	1
02378	Line Dunn	lidu@itu.dk	3

Live Exercise

- What are superkeys and (candidate) keys?
  - (SID) -- (Candidate) Key
  - (Email) -- (Candidate) Key
  - (SID, Name) -- Superkey
  - (Semester) -- None
  - (Email, Semester) -- Superkey
  - (Name) -- None
  - (Name, Semester) -- (Candidate) Key
- What is the best key for being the primary key?
- Which of these keys does not make sense in practice?

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
04321	Jens Schuh	jesc@itu.dk	1
01439	Olga Marx	olmr@itu.dk	2
01832	Peter Pitt	pepi@itu.dk	1
02378	Line Dunn	lidu@itu.dk	3

Live Exercise

- What are superkeys and (candidate) keys?
  - (SID) -- (Candidate) Key
  - (Email) -- (Candidate) Key
  - (SID, Name) -- Superkey
  - (Semester) -- None
  - (Email, Semester) -- Superkey
  - (Name) -- None
  - (Name, Semester) -- (Candidate) Key

- What is the best key for being the primary key?
  - (SID) is ideal for being the Primary Key

- Which of these keys does <u>not</u> make sense in practice?
  - (Name, Semester)

Students

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
04321	Jens Schuh	jesc@itu.dk	1
01439	Olga Marx	olmr@itu.dk	2
01832	Peter Pitt	pepi@itu.dk	1
02378	Line Dunn	lidu@itu.dk	3

Relationships

- **Students** (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)
- **Faculty** (<u>FID</u> : INT, Name : STRING, DID : INT)
- **Courses** (<u>CID</u> : STRING, Name : STRING, DID : INT)
- **Departments** (<u>DID</u> : INT, Name : STRING)
- **Transcripts** (CID : STRING, SID : INT, Grade : STRING, Comment : STRING)

relationships

Foreign Keys

- Defines the relationship between relations

- A key FK in a relation R is a foreign key iff:
  - The attributes in FK matches a primary key PK of a relation S and they are of the same type
  - Any record *i* in R has a value in FK that either
    - occurs as a value of PK for some record j in S, or
    - is null
  - I.e., FK = PK (domain and values)

- A relation can have several foreign keys

Foreign Keys: Example

Departments

PK

DID	Name
1	Business
2	Computer Science

Courses

PK FK

CID	Name	DID
MATH	Bob Brown	2
I2DBS	Intro. To DB Syst.	2
SWE	Soft. Engineering	4

Students

PK Key Relation

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
…	…	…	…

Transcripts

FK FK Foreign Relation

CID	SID	Grade	Comment
I2DBS	01789	7	Student didn't…
MATH	01234	12	Clear display of…

Foreign Keys: Example

PK **Students** Key Relation

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
…	…	…	…

Departments

PK

DID	Name
1	Business
2	Computer Science

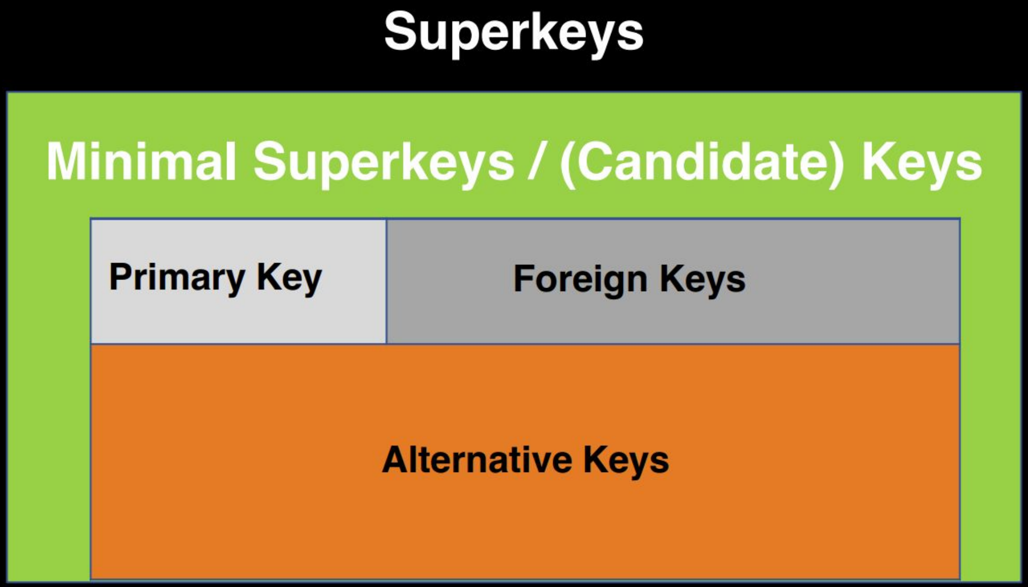
PK **Courses** FK

CID	Name	DID
MATH	Bob Brown	2
I2DBS	Intro. To DB Syst.	2
SWE	Soft. Engineering	4

FK FK **Transcripts** Foreign Relation

CID	SID	Grade	Comment
I2DBS	01789	7	Student didn't…
MATH	01234	12	Clear display of…

All keys in a Relation



NOTE:
Keys are part of the schema

So far, all together

Primary Key = (Candidate) Key = Minimal Superkey

Attributes

Schema

Instance

Students

Key Relation

SID	Name	Email	Semester
01789	Bob Brown	bobr@itu.dk	2
01883	Lucas White	luwh@itu.dk	5
01234	Olga Marx	olma@itu.dk	6
…	…	…	…

Relation name

Superkey

Foreign Keys

Transcripts

Foreign Relation

Records

Relation

CID	SID	Grade	Comment
I2DBS	01789	7	Student didn't…
MATH	01234	12	Clear display of…

Integrity Constraints

- An integrity constraint (IC) is a limitation of the allowed content (or development) of a database
  - Ensures that the data is always correct and consistent
  - There exist various ICs

- It is the RDBMS that takes care of ensuring the ICs in a database

- ICs already seen so far
  - Domain constraint -- attribute type and format (e.g., DATE)
  - Key constraint -- uniqueness & minimality
  - Entity constraint (PK) -- NOT NULL
  - Referential constraint (FK) -- PK = FK

- More advanced ICs
  - Functional dependencies
  - Temporal constraint…

SQL

Structured Query Language (SQL)

- SEQUEL if you worked for IBM in the 80s
- SQL is primarily a query language, for getting information from a database (DML)
  - also includes a data-definition component for describing database schemas (DDL)
- Invented in the 70s by IBM
- The three most common commands in SQL queries
  - SELECT, FROM, WHERE

```
SELECT * FROM Students WHERE Name = ‘Lucas White’;
```

SQL

- Data Definition Language (DDL)
  - Used by the database administrator (DBA) to define the database's data model
  - Three common commands:
    - CREATE TABLE, ALTER TABLE, and DROP TABLE

Today's Focus

- Data Manipulation Language (DML)
  - Used by applications and users to retrieve, insert, modify, and delete records
  - Four statements:
    - SELECT, INSERT, UPDATE, and DELETE

First Normal Form (1NF)

- Each attribute in a relation has:
  - a primitive type (atomic values), and;
  - a unique name

- The main goal is to eliminate redundant data definition

- Benefits:
  - Data integrity
  - Data consistency
  - Easy data manipulation
  - Better data organization

Data Types

Type	Description
CHAR(n)	Fixed-length string of size *n*
VARCHAR(n)	Variable-length string of maximum size *n*
SMALLINT	Small integer (-32,768 and 32,767)
INT	Integer (-2,147,483,648 and 2,147,483,647)
FLOTAT(n, d)	Small number with a floating decimal point: n = max digits and d = max decimals
DOUBLE(n, d)	Large number with a floating decimal point: n = max digits and d = max decimals
DATE	Date in format YYYY-MM-DD
DATETIME	Date and time in format YYYY-MM-DD HH:MI:SS
TIME	Time in format HH:MI:SS
BOOLEAN	True or false
BLOB	Binary large object (typically unstructured)

NOTE:
Check PostgreSQL types here:
https://www.postgresql.org/docs/current/datatype.html

Create Relation Students

Students (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)

```
CREATE TABLE Students (
    SID INT,
    Name VARCHAR(255),
    Email CHAR(11),
    Semester INT
);
```

Define a Primary Key

Students (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)

```
ALTER TABLE Students ADD PRIMARY KEY (SID);
```

```
CREATE TABLE Students (
    SID INT PRIMARY KEY,
    Name VARCHAR(255),
    Email CHAR(11),
    Semester INT
);
```

Courses (<u>CID</u> : STRING, Name : STRING, DID : INT)
Departments (<u>DID</u> : INT, Name : STRING)

Define a Foreign Key

```sql
CREATE TABLE Departments (
    DID INT PRIMARY KEY,
    Name VARCHAR(255)
);

CREATE TABLE Courses (
    CID VARCHAR(25) PRIMARY KEY,
    Name VARCHAR(255),
    DID INT,
    FOREIGN KEY (DID) REFERENCES Departments (DID)
);
```

ALTERNATIVE

```sql
CREATE TABLE Courses (
    CID VARCHAR(25) PRIMARY KEY,
    Name VARCHAR(255),
    DID INT REFERENCES Departments (DID)
);
```

Multiple-Attribute Primary Keys

Students (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)
Courses (<u>CID</u> : STRING, Name : STRING, DID : INT)
Transcripts (<u>CID</u> : STRING, <u>SID</u> : INT, Grade : STRING, Comment : STRING)

```
CREATE TABLE Transcripts (
    CID VARCHAR(25),
    SID INT,
    Grade INT,
    Comment VARCHAR(255),
    PRIMARY KEY (CID, SID),
    FOREIGN KEY (CID) REFERENCES Courses (CID),
    FOREIGN KEY (SID) REFERENCES Students (SID)
);
```

NULL

- What if a value is missing?
  - Does not exist?
  - Unknown?
  - Secret?

- SQL solution: NULL
  - NULL = no value
  - More next week...

- By default, attributes can be NULL
  - Except: PRIMARY KEY attributes
  - Except: NOT NULL attributes

- Allowing NULL values is a design decision!

Transcripts Relation with NOT NULL

Students (<u>SID</u> : INT, Name : STRING, Email : STRING, Semester : INT)
Courses (<u>CID</u> : STRING, Name : STRING, DID : INT)
Transcripts (<u>CID</u> : STRING, <u>SID</u> : INT, Grade : STRING, Comment : STRING)

```
CREATE TABLE Transcripts (
    CID VARCHAR(25) NOT NULL,
    SID INT NOT NULL,
    Grade INT NOT NULL,
    Comment VARCHAR(255),
    PRIMARY KEY (CID, SID),
    FOREIGN KEY (CID) REFERENCES Courses (CID),
    FOREIGN KEY (SID) REFERENCES Students (SID)
);
```

Drop All Created Relations

```
DROP TABLE Transcripts;
DROP TABLE Courses;
DROP TABLE Departments;
DROP TABLE Students;
```

ORDER MATTERS!!!

Drop Foreign Relations before Key Relations

Takeaways

Can’t wait for the SQL!

Relational Model

- Relations, attributes, keys, primary & foreign keys…

SQL DDL = Data Definition Language

- CREATE TABLE, DROP TABLE, ALTER TABLE, …
- Allows to create complex schemas and maintain them

SQL DML = Data Manipulation Language

- INSERT, DELETE, UPDATE, SELECT
- Simple set of commands for complicated selections
- *(we will dive into it next week)*

Next Time in IDBS…

Introduction to Database Systems IDBS - Fall 2024

Lecture 2 - SQL

SQL DML
Basic SQL queries
Joins
Aggregations

Readings: PDBM 7.3

Eleni Tzirita Zacharatou

IT UNIVERSITY OF COPENHAGEN