Introduction to Database Systems IDBS – Spring 2024

- Week 1:
- Course Introduction
- DBMS Introduction
- Relational Data Model
- SQL DDL

Eleni Tzirita Zacharatou

Readings:

PDBM 1, 6.1, 7.1-7.2

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 Researcher
TUB, Germany

2022 - now

Assistant Professor
ITU, Denmark

Website: https://heltzi.github.io

Lecturer

Omar Shahbaz Khan

BSc in Software Development

MSc in Computer Science

PhD in Computer Science

Postdoctoral Researcher

2013 **2016** **2018** **2022** **Now**

Reykjavik University / IT University of Copenhagen

Teaching Assistants

Anders Arvesen
Study program: BSc. Software Development

Adam Hadou Temsamani
Study program: BSc. Software Development

Anne-Marie Rommerdahl
Study program: BSc. Software Development

Katrine Martos Sandø-Pedersen
Study program: BSc. Software Development

Oliver Flyckt Wilhjelm
Study program: BSc. Software Development

Erling Amundsen
Study program: BSc. Data Science

Frederik Rothe
Study program: MSc. Computer Science

Research Group

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

Entire Data Lifecycle:

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

Infrastructure for Data Science

https://dasya.itu.dk

15+ People

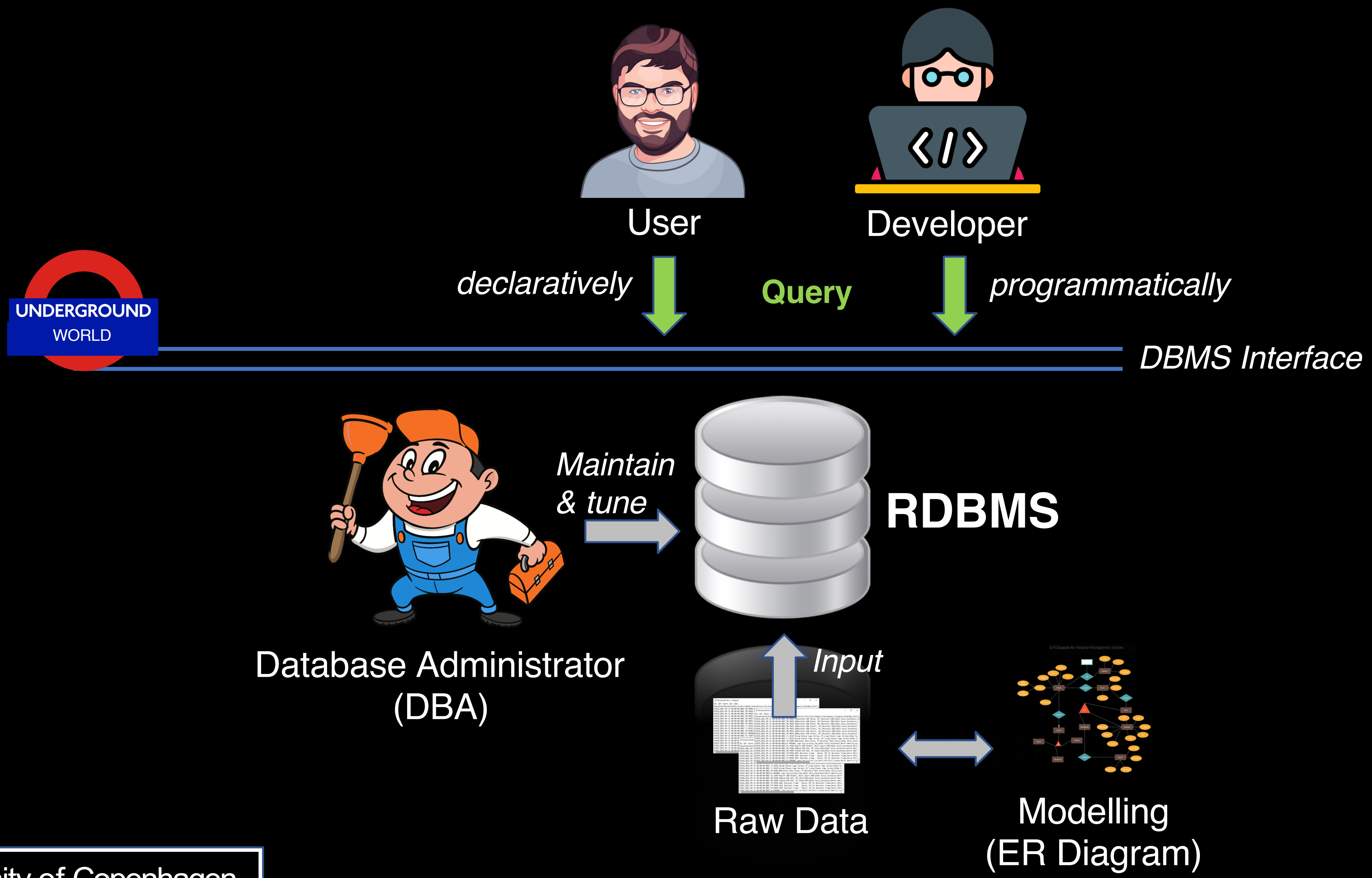
DASYA

Data-Intensive Systems and Applications

But This is About You!

- **~ 145 Students**
  Mostly from MSc. in Software Design
- **Too many for a round table**
- **What do you expect from Introduction to Database Systems?**
  Mentimeter: https://www.menti.com/al8ndx4u3bpr

What You will Learn



User

Developer

declaratively

Query

programmatically

UNDERGROUND WORLD

DBMS Interface

Maintain & tune

RDBMS

Database Administrator (DBA)

Input

Raw Data

Modelling (ER Diagram)

Intented Learning Outcomes

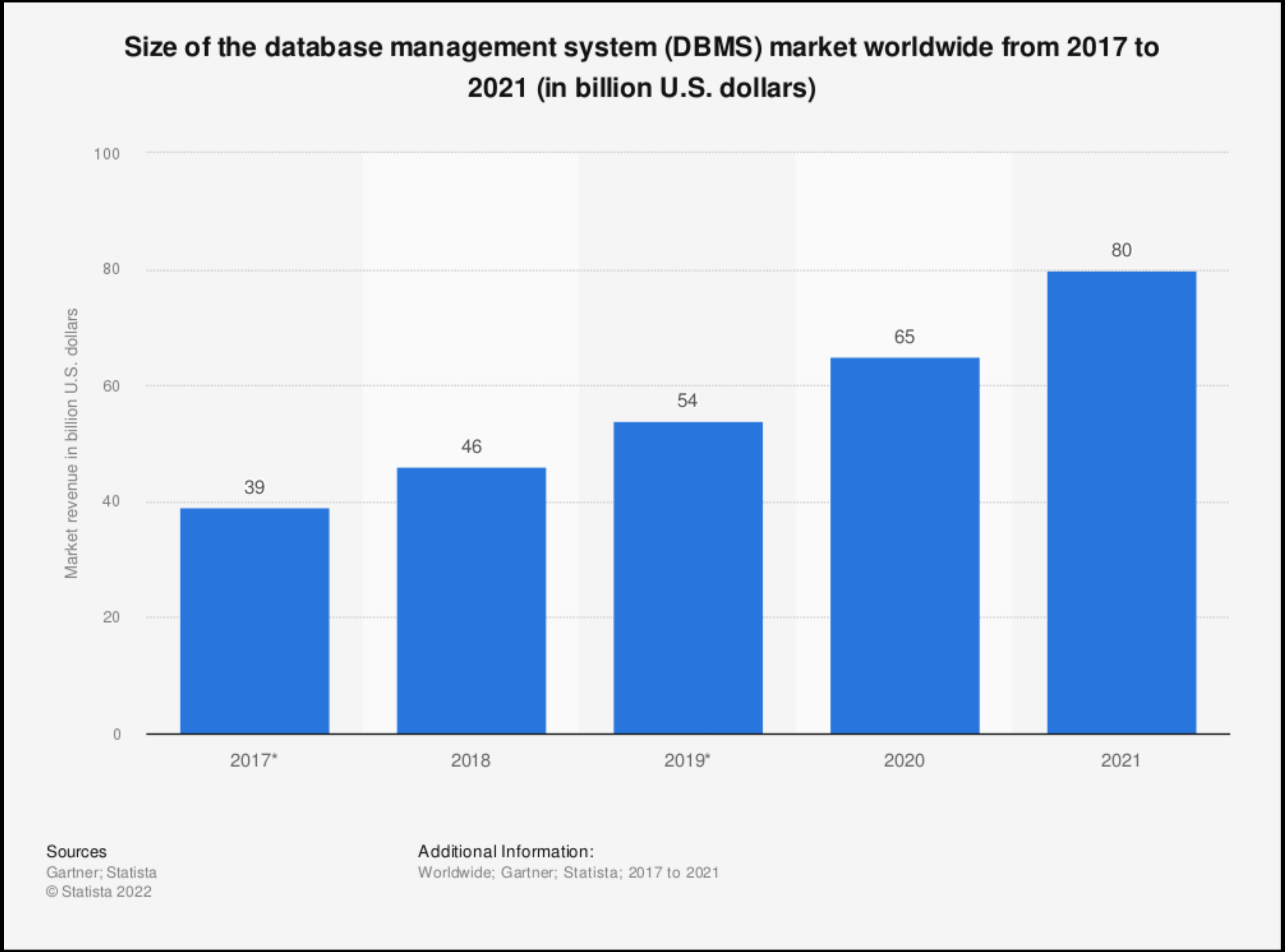
- Write SQL queries: multiple relations; compound conditions; grouping; aggregation; and subqueries.
- Use relation cure manner.
- Suggest a d abase schema in a suitable
- Analyze/pre base using indices.

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)

- 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.

Why is It Important?

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



Size of the database management system (DBMS) market worldwide from 2017 to 2021 (in billion U.S. dollars)

Sources
Gartner; Statista
© Statista 2022

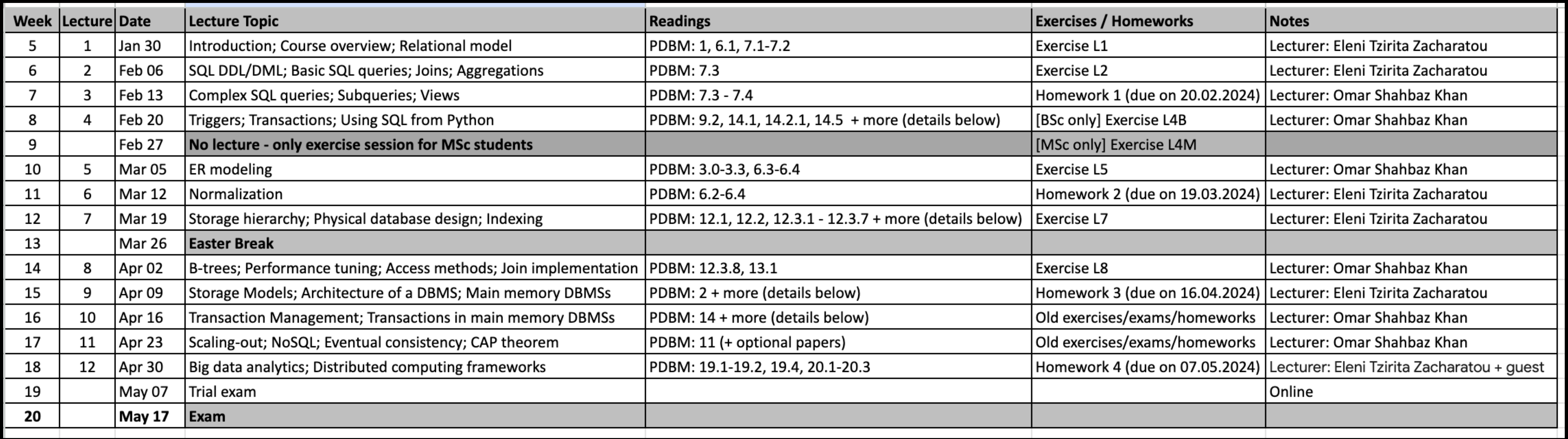
Additional Information:
Worldwide; Gartner; Statista; 2017 to 2021

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

Course Schedule

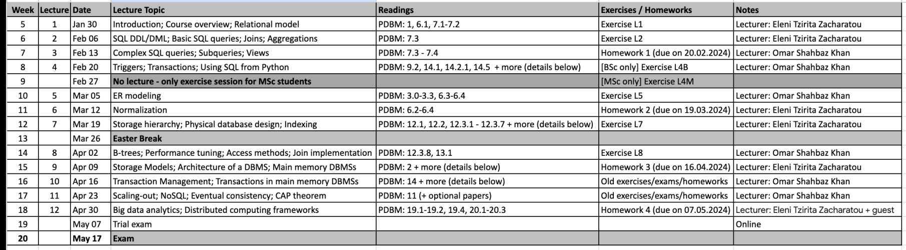
BSc vs MSc

- *Pretty much the same course*
- *One different question (5%) on exam*

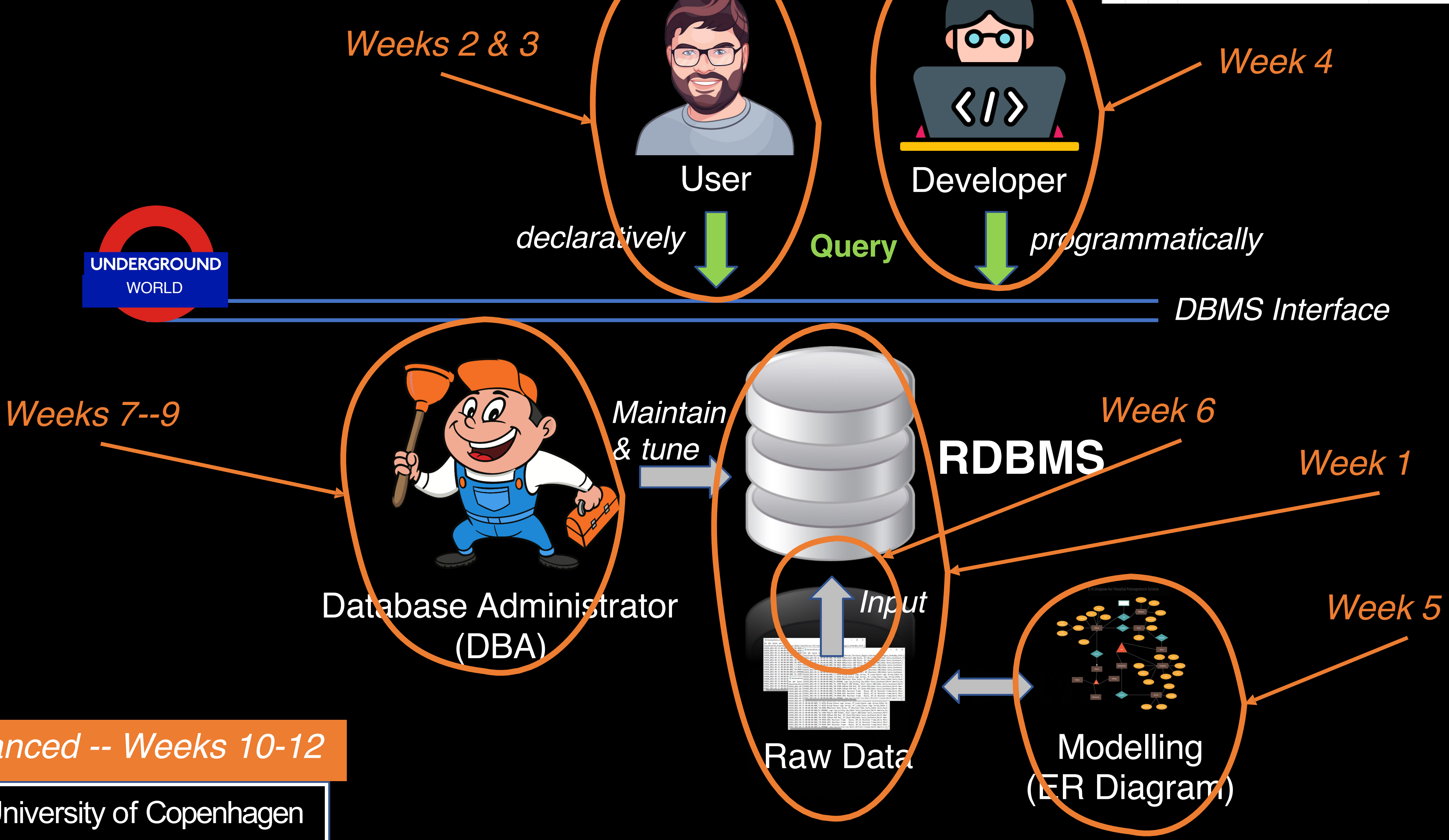


Week	Lecture	Date	Lecture Topic	Readings	Exercises / Homeworks	Notes
5	1	Jan 30	Introduction; Course overview; Relational model	PDBM: 1, 6.1, 7.1-7.2	Exercise L1	Lecturer: Eleni Tzirita Zacharatou
6	2	Feb 06	SQL DDL/DML; Basic SQL queries; Joins; Aggregations	PDBM: 7.3	Exercise L2	Lecturer: Eleni Tzirita Zacharatou
7	3	Feb 13	Complex SQL queries; Subqueries; Views	PDBM: 7.3 - 7.4	Homework 1 (due on 20.02.2024)	Lecturer: Omar Shahbaz Khan
8	4	Feb 20	Triggers; Transactions; Using SQL from Python	PDBM: 9.2, 14.1, 14.2.1, 14.5 + more (details below)	[BSc only] Exercise L4B	Lecturer: Omar Shahbaz Khan
9		Feb 27	**No lecture - only exercise session for MSc students**		[MSc only] Exercise L4M	
10	5	Mar 05	ER modeling	PDBM: 3.0-3.3, 6.3-6.4	Exercise L5	Lecturer: Omar Shahbaz Khan
11	6	Mar 12	Normalization	PDBM: 6.2-6.4	Homework 2 (due on 19.03.2024)	Lecturer: Eleni Tzirita Zacharatou
12	7	Mar 19	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
13		Mar 26	**Easter Break**			
14	8	Apr 02	B-trees; Performance tuning; Access methods; Join implementation	PDBM: 12.3.8, 13.1	Exercise L8	Lecturer: Omar Shahbaz Khan
15	9	Apr 09	Storage Models; Architecture of a DBMS; Main memory DBMSs	PDBM: 2 + more (details below)	Homework 3 (due on 16.04.2024)	Lecturer: Eleni Tzirita Zacharatou
16	10	Apr 16	Transaction Management; Transactions in main memory DBMSs	PDBM: 14 + more (details below)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
17	11	Apr 23	Scaling-out; NoSQL; Eventual consistency; CAP theorem	PDBM: 11 (+ optional papers)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
18	12	Apr 30	Big data analytics; Distributed computing frameworks	PDBM: 19.1-19.2, 19.4, 20.1-20.3	Homework 4 (due on 07.05.2024)	Lecturer: Eleni Tzirita Zacharatou + guest
19		May 07	Trial exam			Online
20		**May 17**	**Exam**			

Course Schedule (Illustrated)



Week	Lecture	Date	Lecture Topic	Readings	Exercises / Homeworks	Notes
5	1	Jan 30	Introduction; Course overview; Relational model	PDBM: 1, 6.1, 7.1-7.2	Exercise L1	Lecturer: Eleni Tzirita Zacharatou
6	2	Feb 06	SQL DDL/DML; Basic SQL queries; Joins; Aggregations	PDBM: 7.3	Exercise L2	Lecturer: Eleni Tzirita Zacharatou
7	3	Feb 13	Complex SQL queries; Subqueries; Views	PDBM: 7.3 - 7.4	Homework 1 (due on 20.02.2024)	Lecturer: Omar Shahbaz Khan
8	4	Feb 20	Triggers; Transactions; Using SQL from Python	PDBM: 9.2, 14.1, 14.2.1, 14.5 + more (details below)	[BSc only] Exercise L4B	Lecturer: Omar Shahbaz Khan
9		Feb 27	**No lecture - only exercise session for MSc students**		[MSc only] Exercise L4M	
10	5	Mar 05	ER modeling	PDBM: 3.0-3.3, 6.3-6.4	Exercise L5	Lecturer: Omar Shahbaz Khan
11	6	Mar 12	Normalization	PDBM: 6.2-6.4	Homework 2 (due on 19.03.2024)	Lecturer: Eleni Tzirita Zacharatou
12	7	Mar 19	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
13		Mar 26	**Easter Break**			
14	8	Apr 02	B-trees; Performance tuning; Access methods; Join implementation	PDBM: 12.3.8, 13.1	Exercise L8	Lecturer: Omar Shahbaz Khan
15	9	Apr 09	Storage Models; Architecture of a DBMS; Main memory DBMSs	PDBM: 2 + more (details below)	Homework 3 (due on 16.04.2024)	Lecturer: Eleni Tzirita Zacharatou
16	10	Apr 16	Transaction Management; Transactions in main memory DBMSs	PDBM: 14 + more (details below)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
17	11	Apr 23	Scaling-out; NoSQL; Eventual consistency; CAP theorem	PDBM: 11 (+ optional papers)	Old exercises/exams/homeworks	Lecturer: Omar Shahbaz Khan
18	12	Apr 30	Big data analytics; Distributed computing frameworks	PDBM: 19.1-19.2, 19.4, 20.1-20.3	Homework 4 (due on 07.05.2024)	Lecturer: Eleni Tzirita Zacharatou + guest
19		May 07	Trial exam			Online
20		**May 17**	**Exam**			



Advanced -- Weeks 10-12

Course Structure

- **Lectures**
  - *Tuesdays 8.15 – 10:00 at Aud 1*
  - *Preparation required: reading material, watching videos*
- **Exercises**
  - *Tuesdays 10.15 – 12:00 at 3A54, 4A56, 4A58 (not this week)*
  - *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!*

Note

Exercise 1 Rooms per OS:

- Windows (3A54)
- Mac (4A56)
- Linux (4A56)
- Exceptionally this week: not using room 4A58

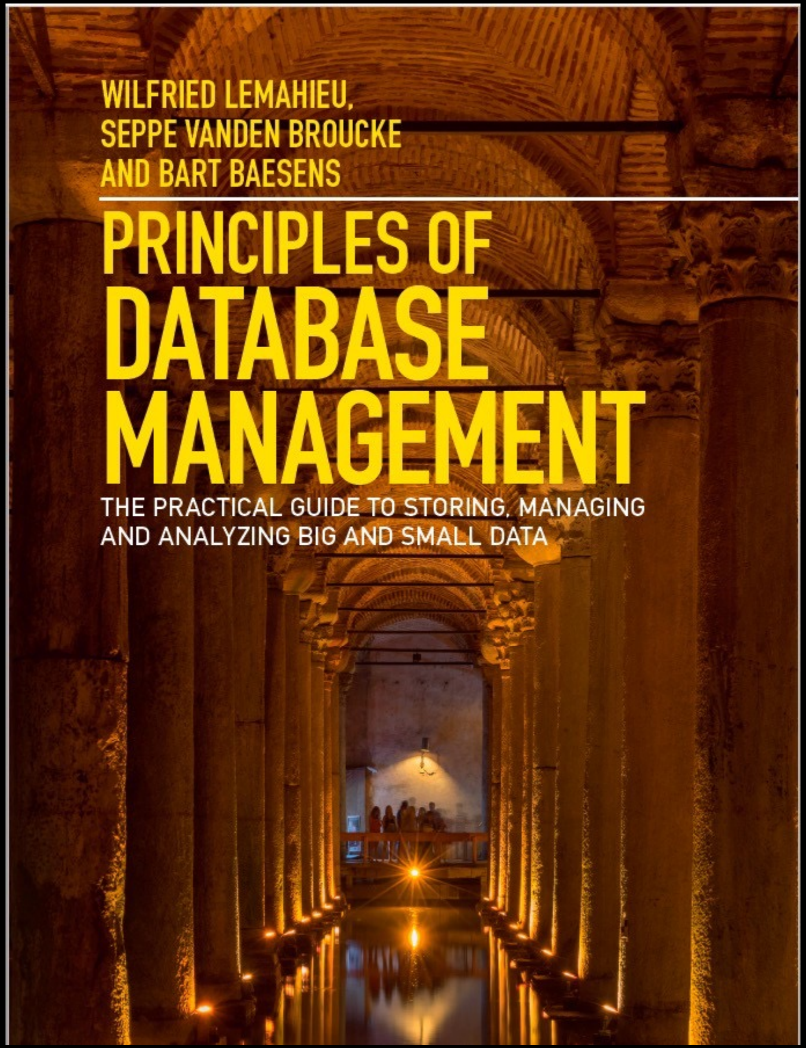
Course Methodology

- **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!**

Advice

Prepare and Ask Questions

Book and Database System



WILFRIED LEMAHIEU,
SEPPE VANDEN BROUCKE
AND BART BAESENS

PRINCIPLES OF DATABASE MANAGEMENT

THE PRACTICAL GUIDE TO STORING, MANAGING AND ANALYZING BIG AND SMALL DATA

+

PostgreSQL
(pgAdmin + psql)

How will We Assess your Learning?

100% Exam (Quiz on LearnIT)

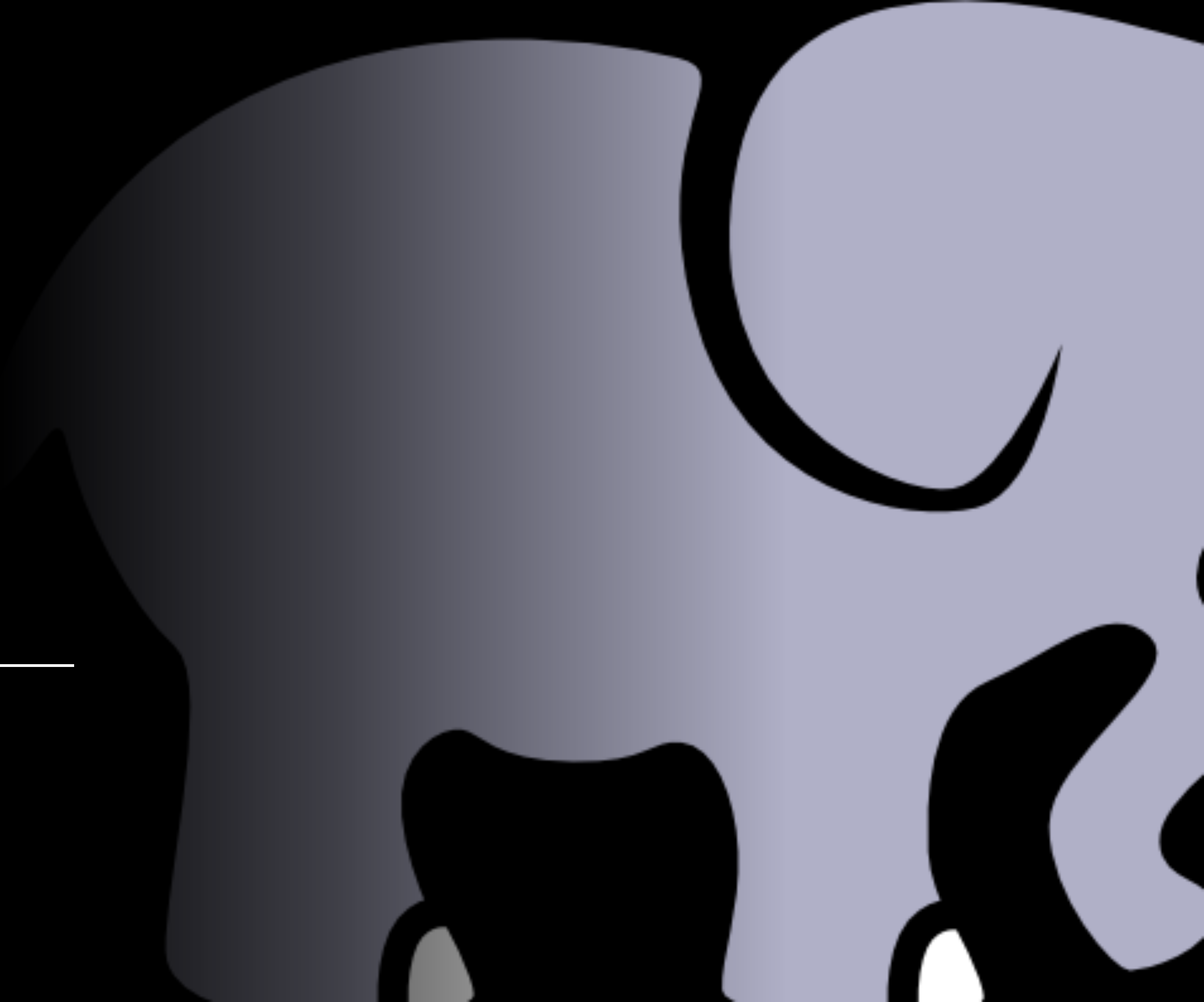
- Restricted: no Internet access!
- All course materials are allowed **offline**
- Communication is not allowed

Exercises and homeworks will help you prepare!

Advice

Study the material weekly

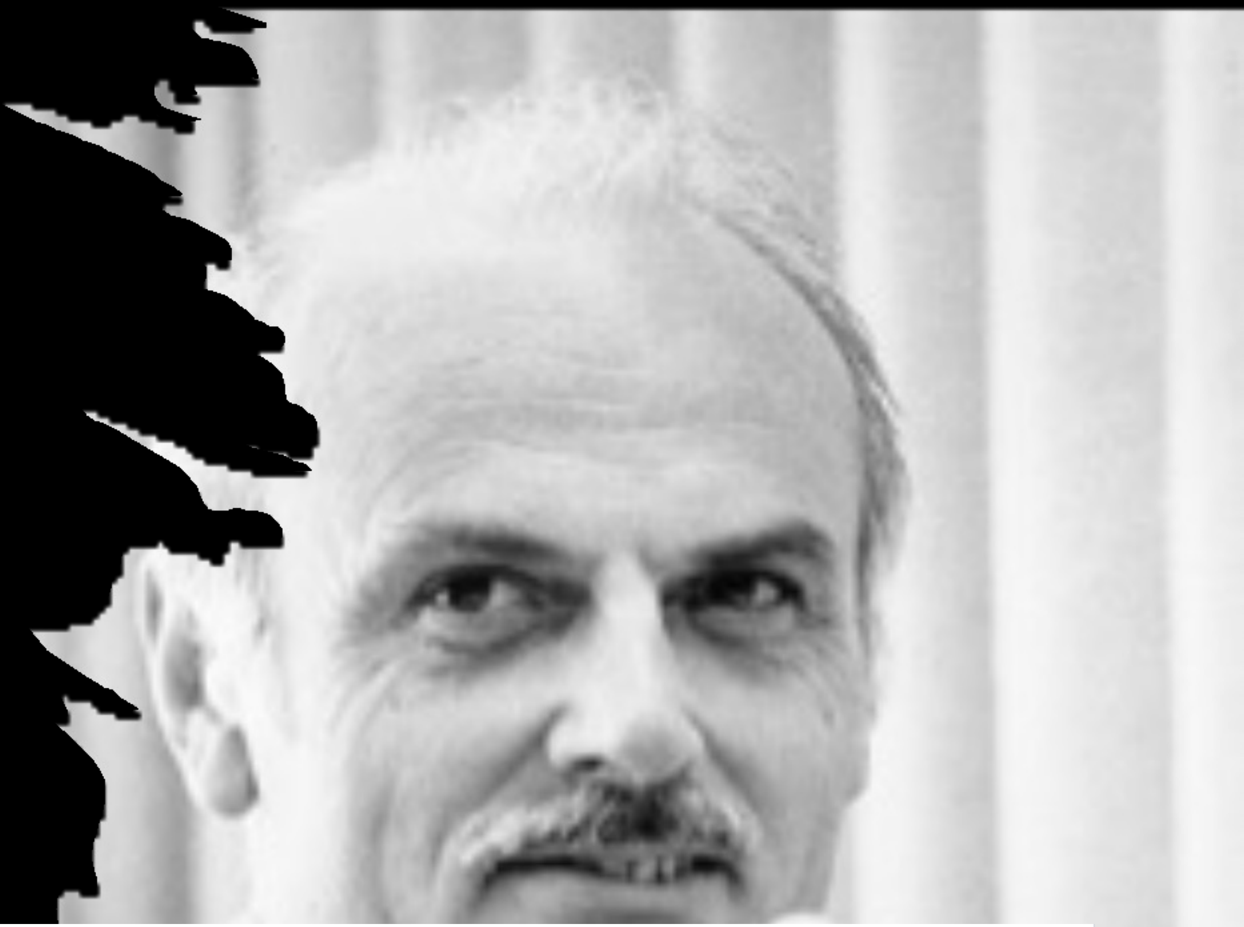
Profile of the Week

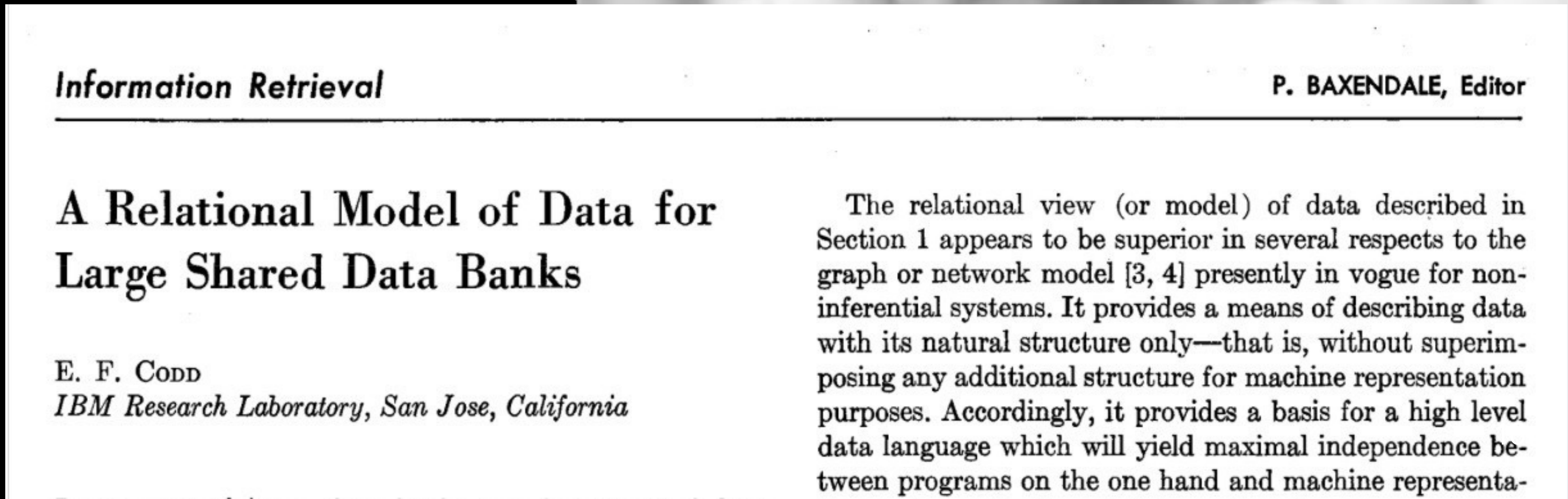


Edgar F. Codd

***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-

RDBMS

DBMS Brief Introduction

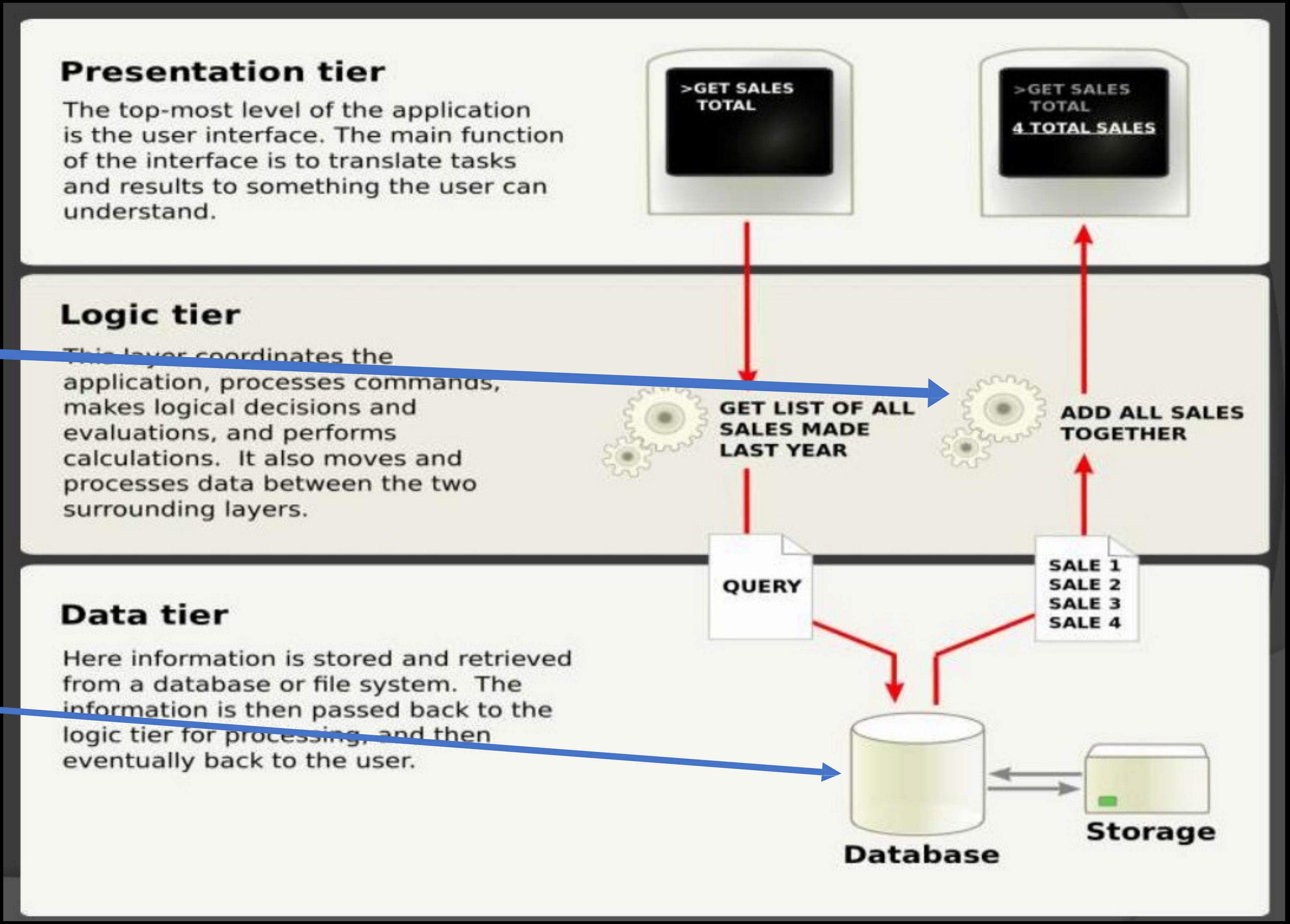
Readings:

PDBM 1

Three-Layer Applications

pgAdmin + psql

PostgreSQL



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

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

Database Definition

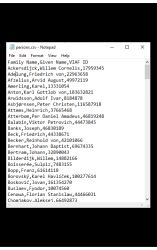
- A **database** is a collection of related data items within a specific business process or problem setting
- A **database system** provides a way to systematically organize, store, retrieve, and query a database

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

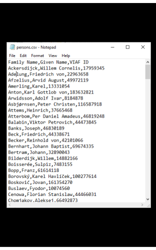
File-based Database

Apps

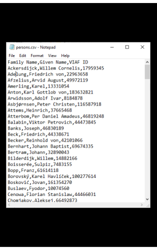
File storage



```
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov.Aleksei.66492873
```



Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov.Aleksei.66492873



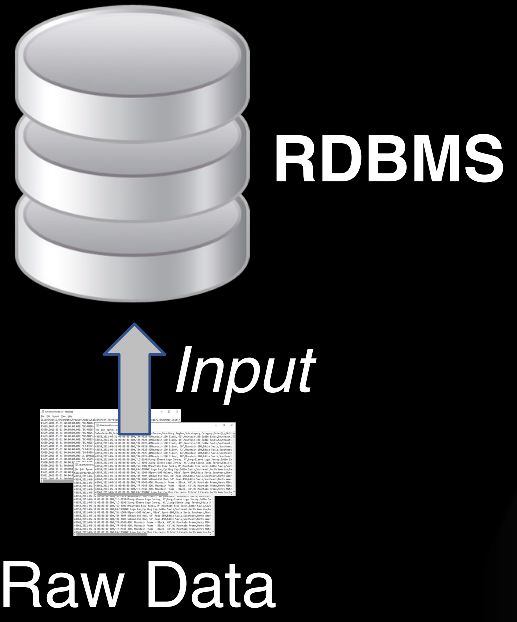
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bosković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov.Aleksei.66492873

Relational Database

- 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 independency**
  - High concurrency
  - Cheap maintenance
  - High performance



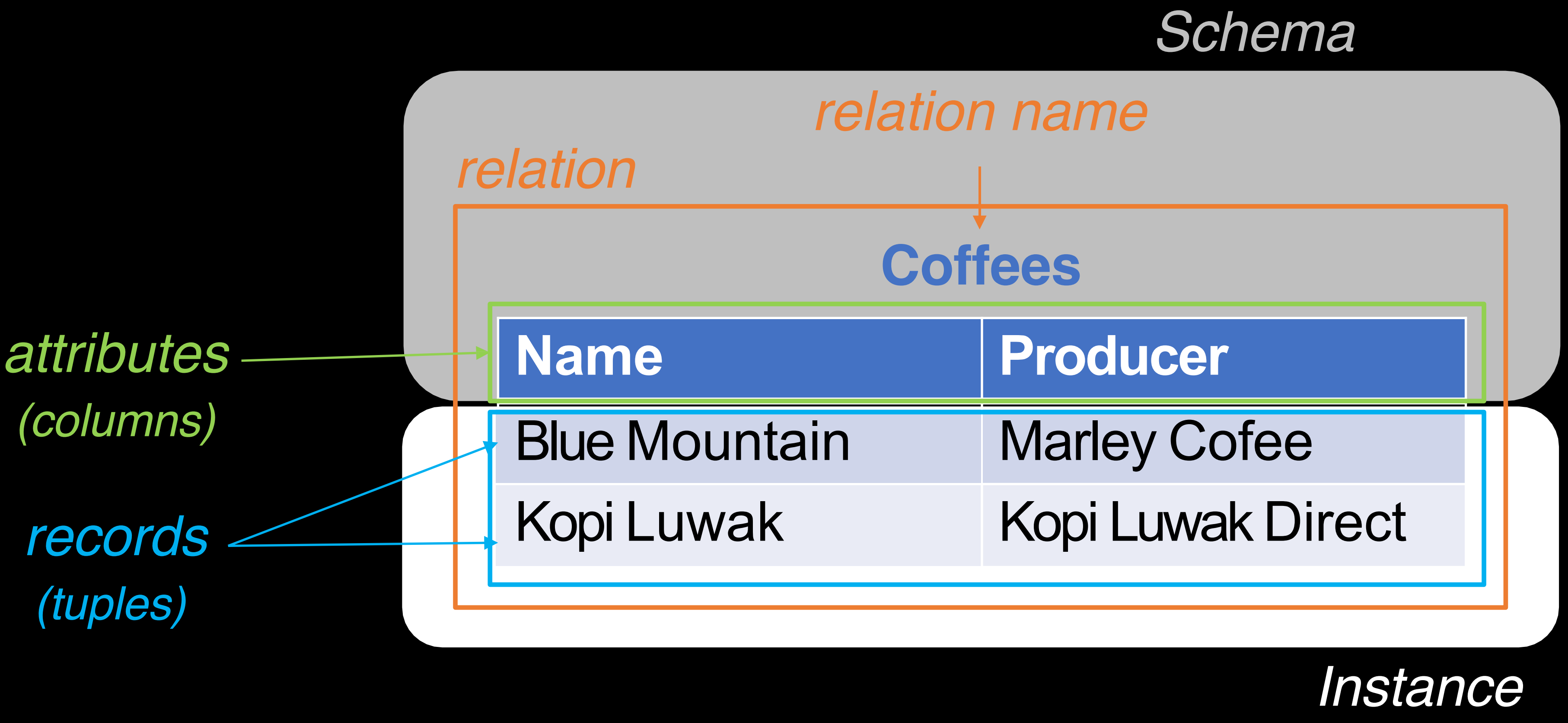
Relational Model

Readings:

PDBM 1, 6.1

Basic Concepts

Relation



Coffees

Name	Producer
Blue Mountain	Marley Cofee
Kopi Luwak	Kopi Luwak Direct

Basic Concepts

Schema vs Instance vs Database

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

- **Database = 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

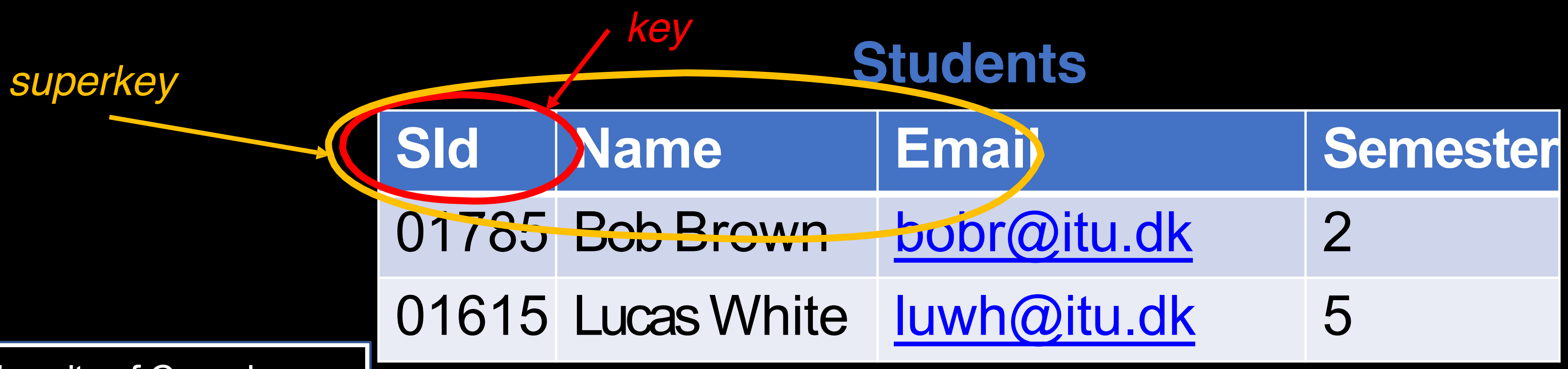
identifier

- **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)

Keys in Relations

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 identify records: *Uniqueness* property
  - The entire set of attributes of a relation is a superkey
  - Minimal superkey: *Minimality* property
    - No attribute can be removed from a superkey without violating the uniqueness property



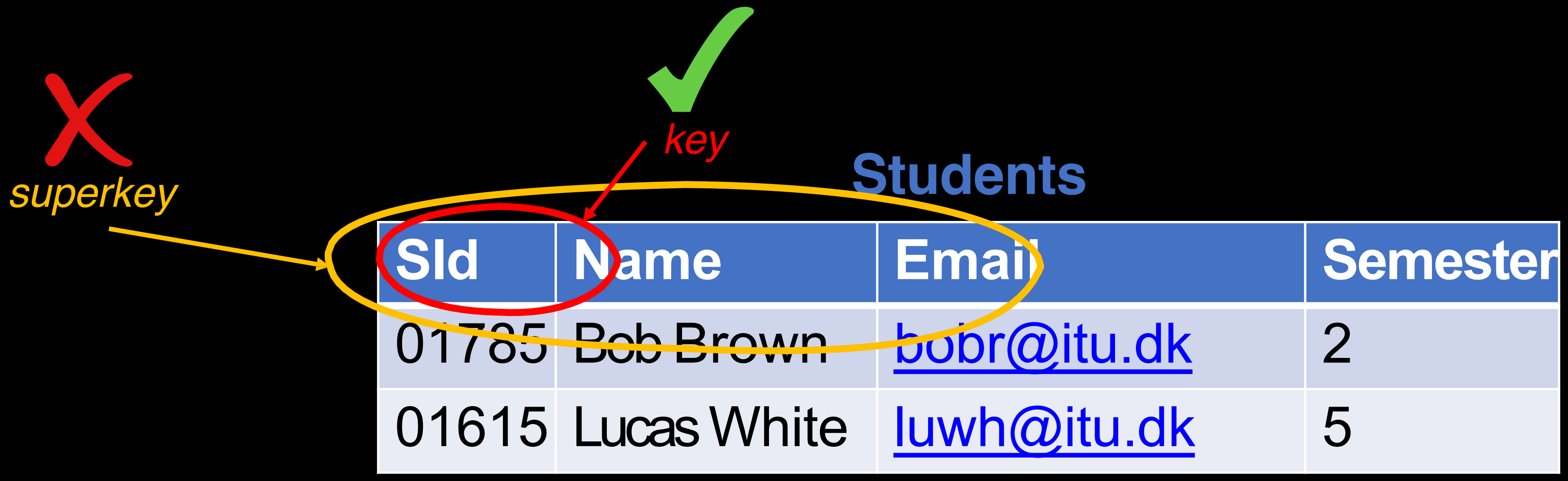
Students

SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5

Keys in Relations

Candidate Keys

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



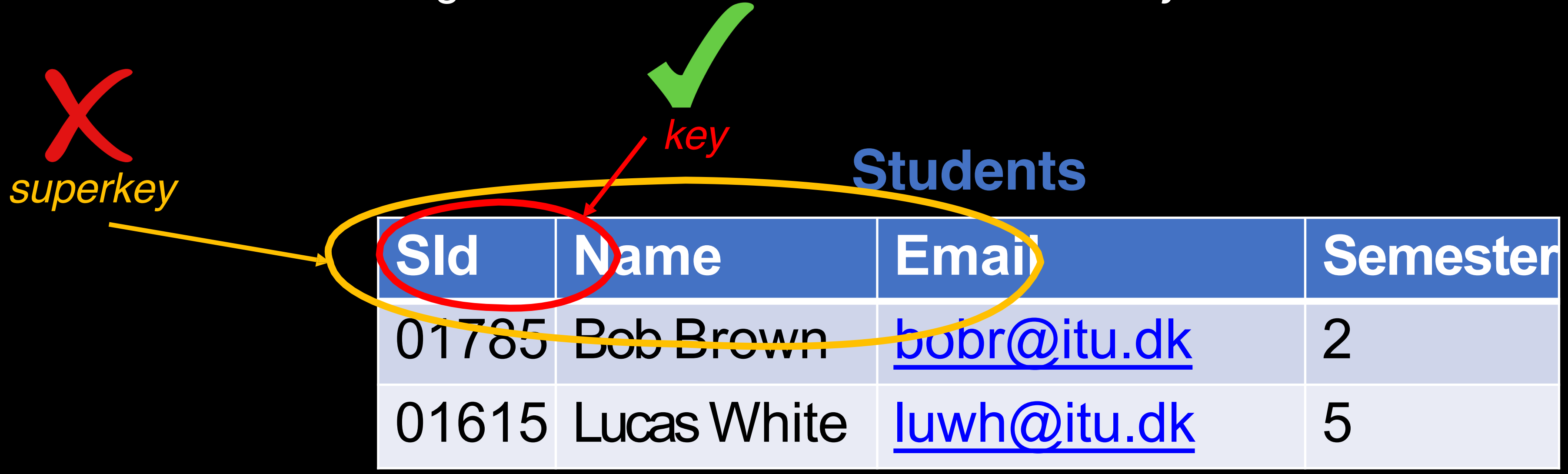
Students

SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5

Keys in Relations

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 primary key
    - The remaining ones are known as *Alternative Keys*



SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5

Keys in Relations

Live Exercise

- **What are superkeys and keys?**
  - (SId)
  - (Email)
  - (SId, Name)
  - (Semester)
  - (Email, Semester)
  - (Name)
  - (Name, Semester)

Students

SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5
01436	Olga Marx	olma@itu.dk	6
01875	Jens Schuh	jesc@itu.dk	1
01803	Olga Marx	olmr@itu.dk	2
01567	Peter Pitt	pepi@itu.dk	1

- **What is the best key for being the primary key?**

- **Which of these keys (does) not make sense in practice?**

Relationships

- **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)

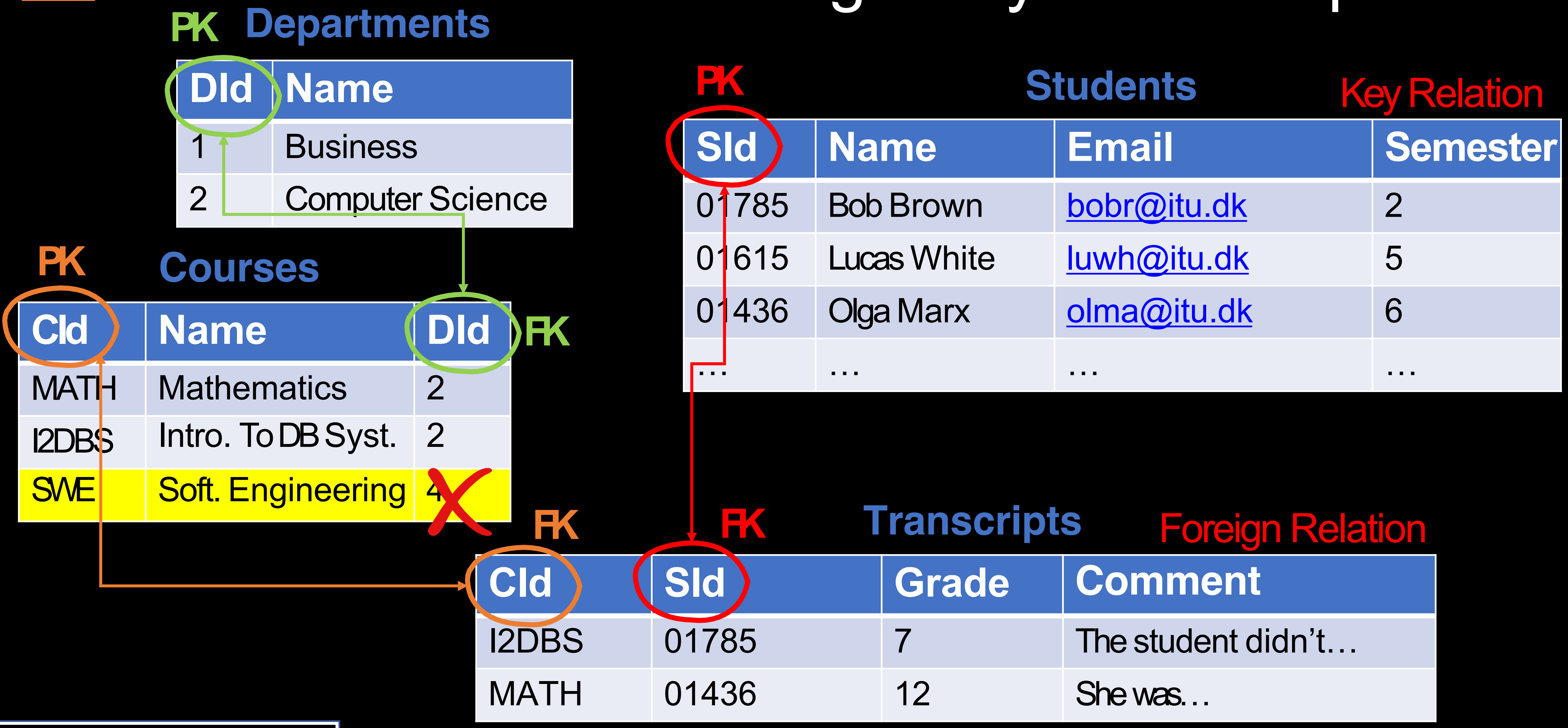
relationship

Keys in Relations

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 tuple *j* in S, or
    - is null
  - I.e., FK = PK (domain and values)
- **A relation can have several foreign keys**

Foreign Keys -- Example



Departments

DId	Name
1	Business
2	Computer Science

Courses

CId	Name	DId
MATH	Mathematics	2
I2DBS	Intro. To DB Syst.	2
SWE	Soft. Engineering	4

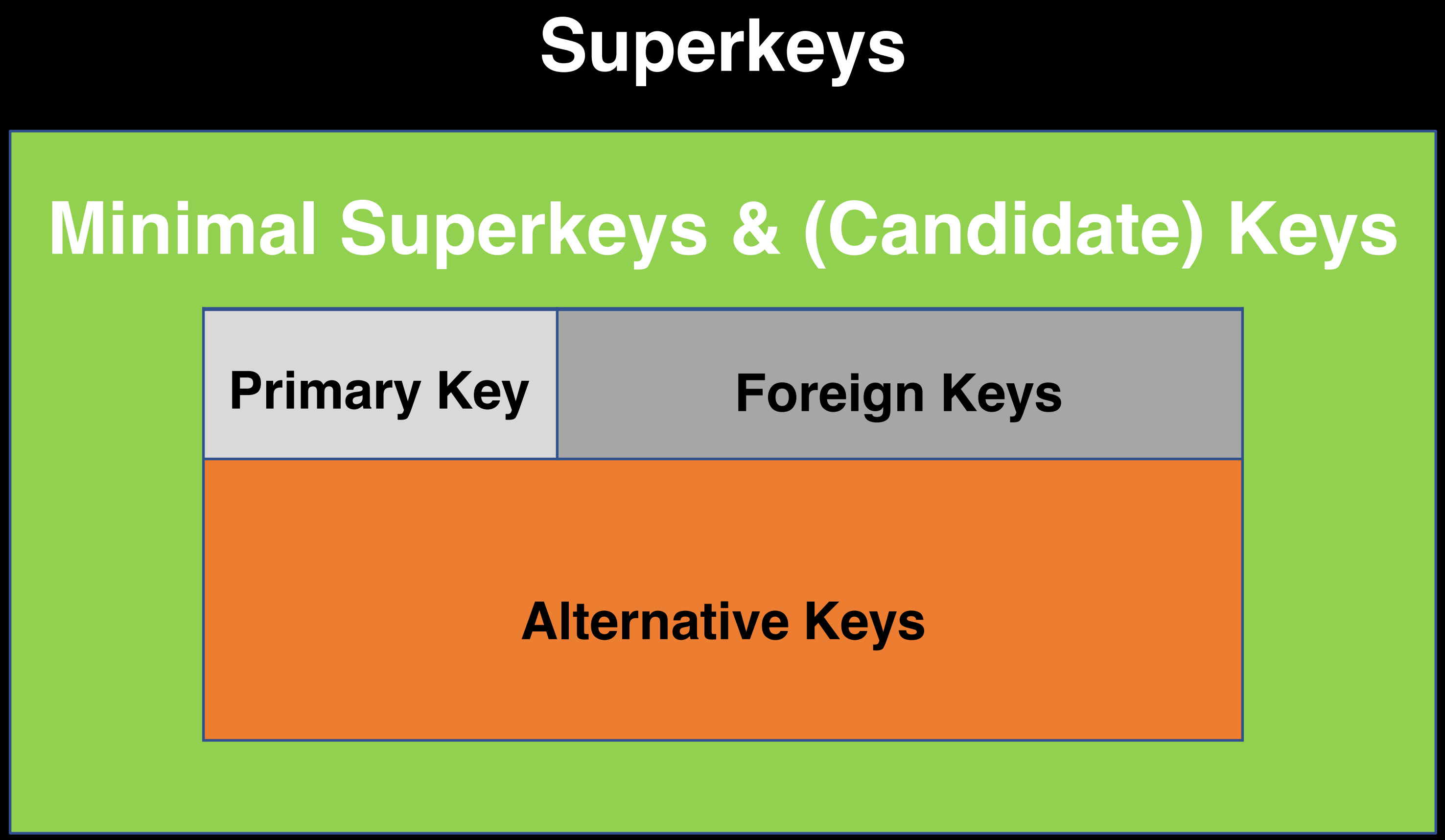
Students

SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5
01436	Olga Marx	olma@itu.dk	6
…	…	…	…

Transcripts

CId	SId	Grade	Comment
I2DBS	01785	7	The student didn't…
MATH	01436	12	She was…

All Keys in a Relation



Superkeys

Minimal Superkeys & (Candidate) Keys

Primary Key

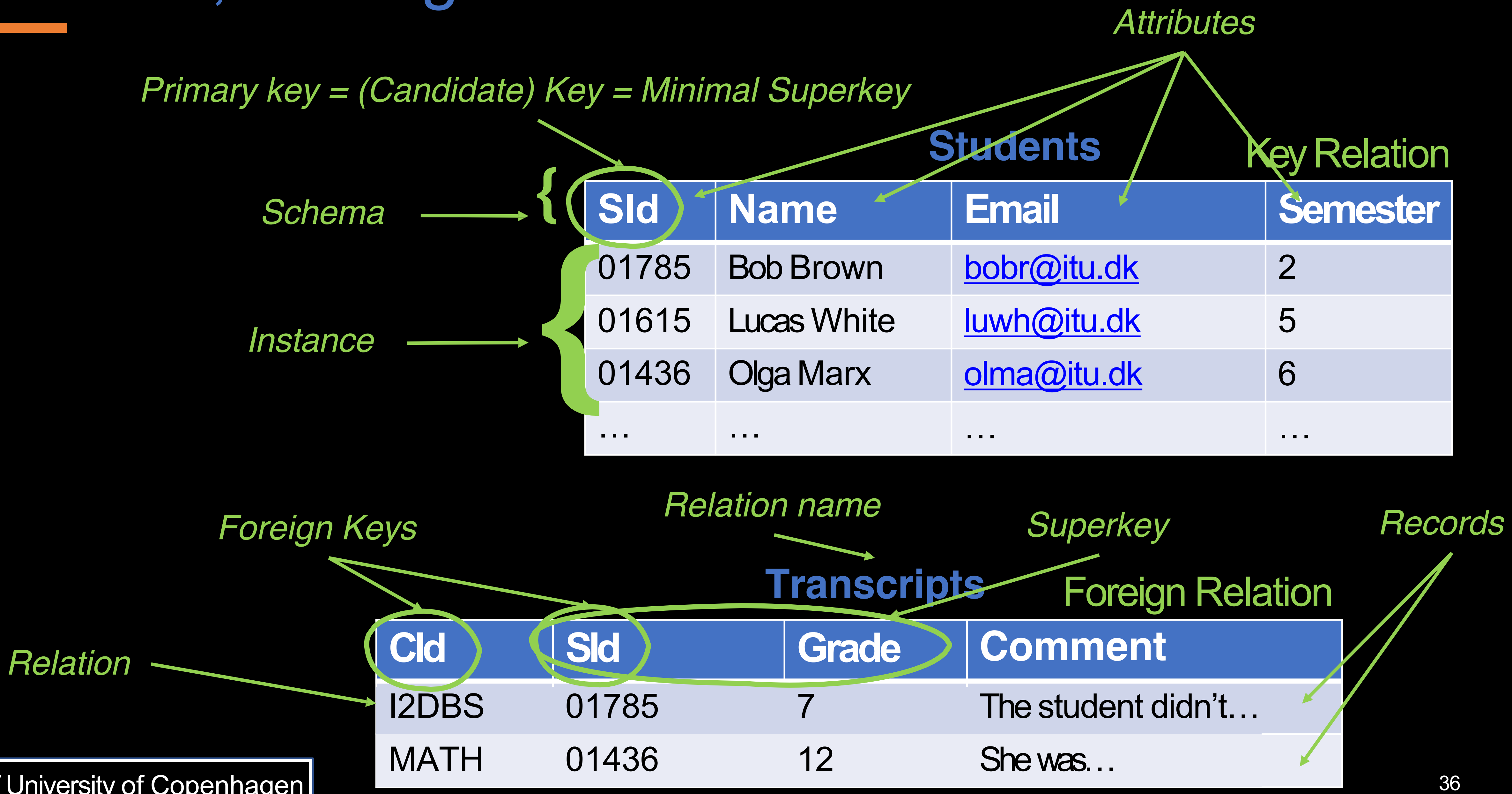
Foreign Keys

Alternative Keys

Note

Keys are part of the schema

So Far, All Together



Attributes

Primary key = (Candidate) Key = Minimal Superkey

Schema

Instance

Students

Key Relation

SId	Name	Email	Semester
01785	Bob Brown	bobr@itu.dk	2
01615	Lucas White	luwh@itu.dk	5
01436	Olga Marx	olma@itu.dk	6
…	…	…	…

Relation name

Foreign Keys

Superkey

Records

Transcripts

Foreign Relation

Relation

CId	SId	Grade	Comment
I2DBS	01785	7	The student didn't…
MATH	01436	12	She was…

Integrity Constraints

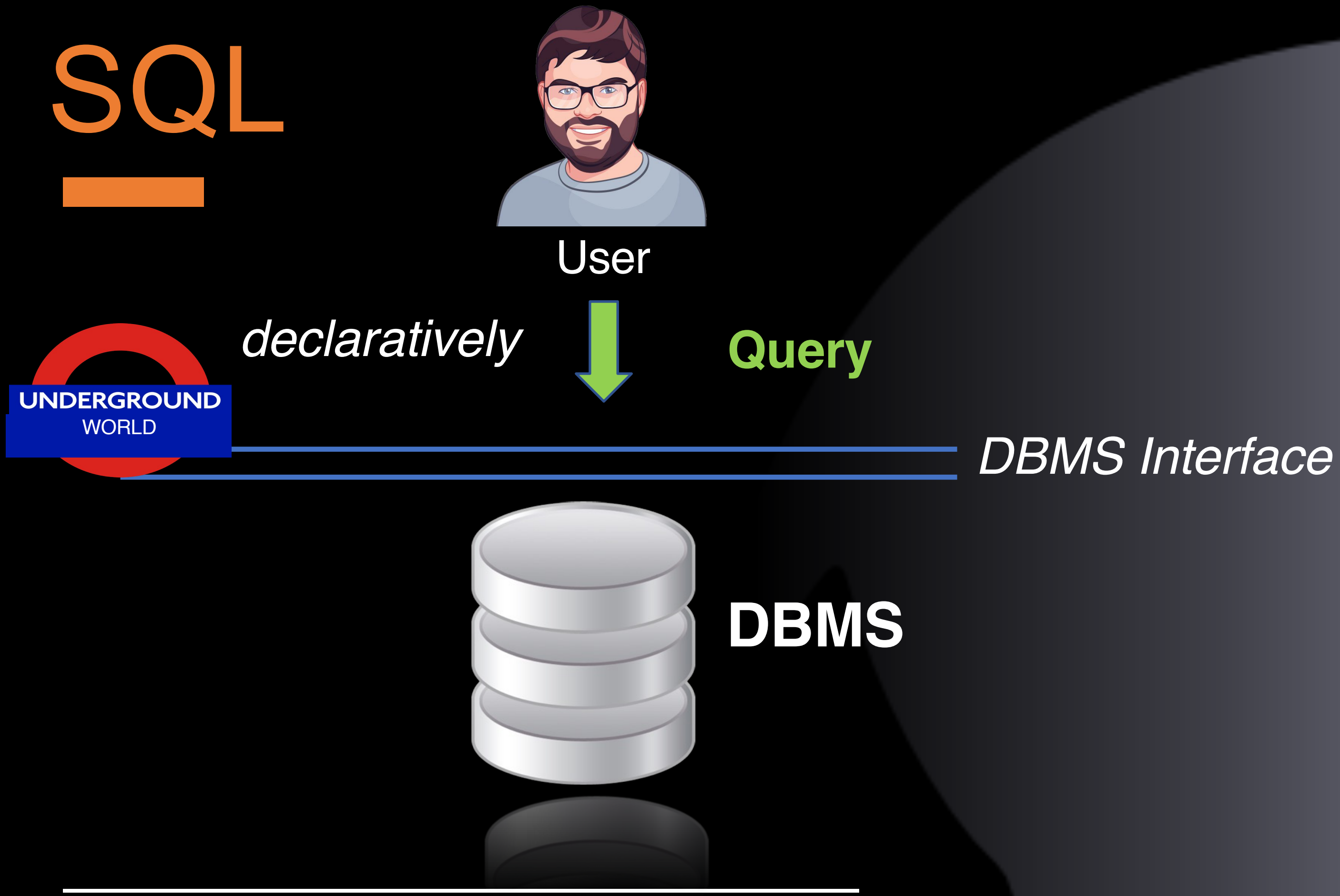
- **An integrity constraint (IC) is a limitation of the allowed content (or development) of a database**
  - Ensures that the data are always correct and consistent
  - There exist various ICs

- **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

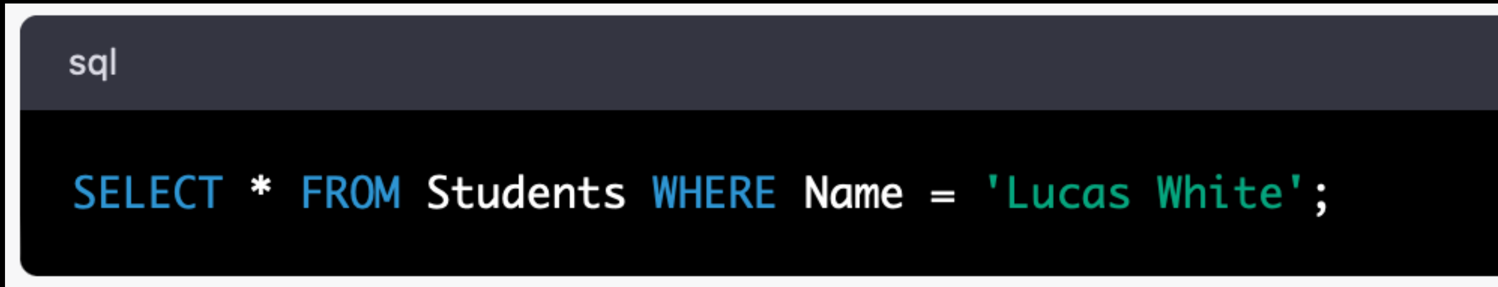


Readings:

PDBM 7.1-7.2

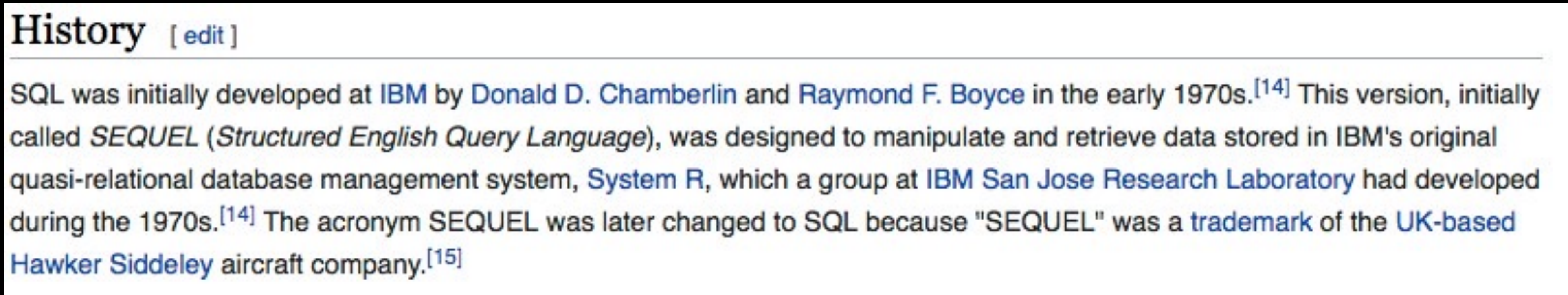
Structured Query Language (SQL)

- **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



sql

```sql
SELECT * FROM Students WHERE Name = 'Lucas White';
```



History [edit]

SQL was initially developed at IBM by Donald D. Chamberlin and Raymond F. Boyce in the early 1970s.[14] This version, initially called *SEQUEL* (*Structured English Query Language*), was designed to manipulate and retrieve data stored in IBM's original quasi-relational database management system, System R, which a group at IBM San Jose Research Laboratory had developed during the 1970s.[14] The acronym SEQUEL was later changed to SQL because "SEQUEL" was a trademark of the UK-based Hawker Siddeley aircraft company.[15]

Two in One Language

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 in a relation

Benefits:

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

SQL -- DDL

Data Types

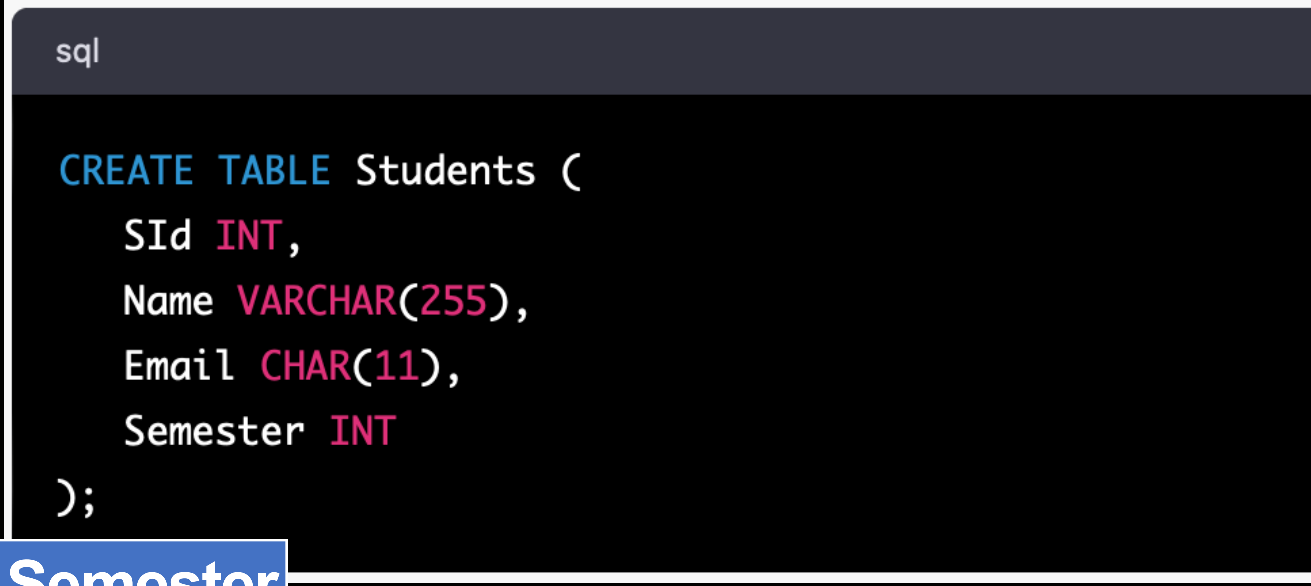
Type	Description
CHAR(n)	Fixed-length string of size *n*
VARCHAR(n)	Variable-length string of maximum size *n*
SMALLINT	Small integer (-32,768 and 32,767)
INT	Integer (-2,147,483,648 and 2,147,483,647)
FLOTAT(n, d)	Small number with a floating decimal point: n = max digits and d = max decimals
DOUBLE(n, 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 types in PostgreSQL

SQL -- DDL

Create Relation Students



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

Students

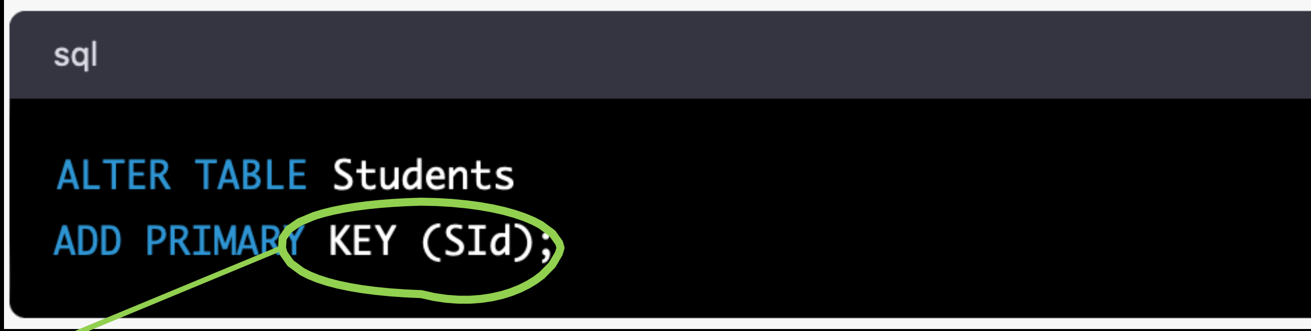
SId	Name	Email	Semester

Advice

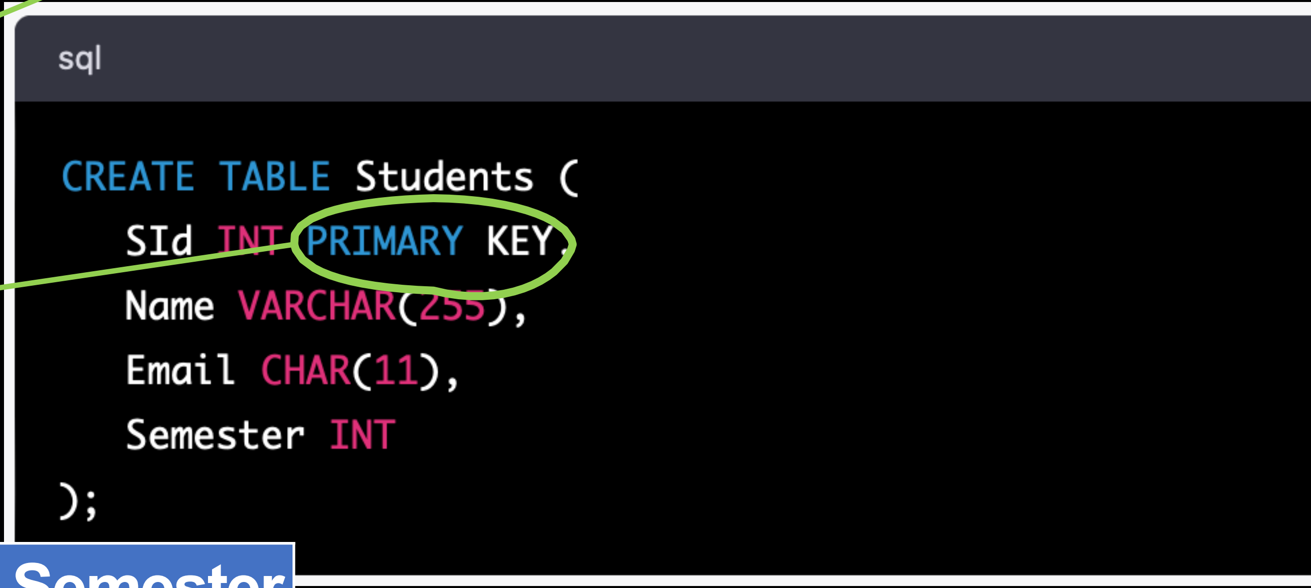
Start playing with PostgreSQL ASAP!

SQL -- DDL

Define a Primary Key



```sql
ALTER TABLE Students
ADD PRIMARY KEY (SId);
```



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

Students

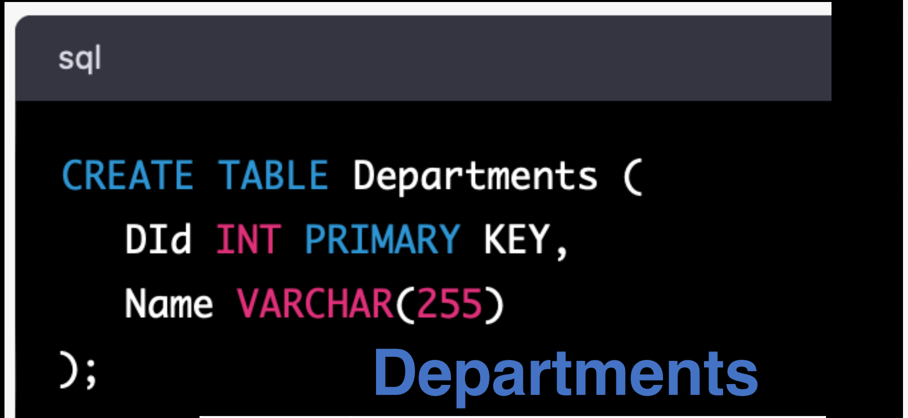
SId	Name	Email	Semester

SQL -- DDL

Define a Foreign Key

Students

SId	Name	Email	Semester

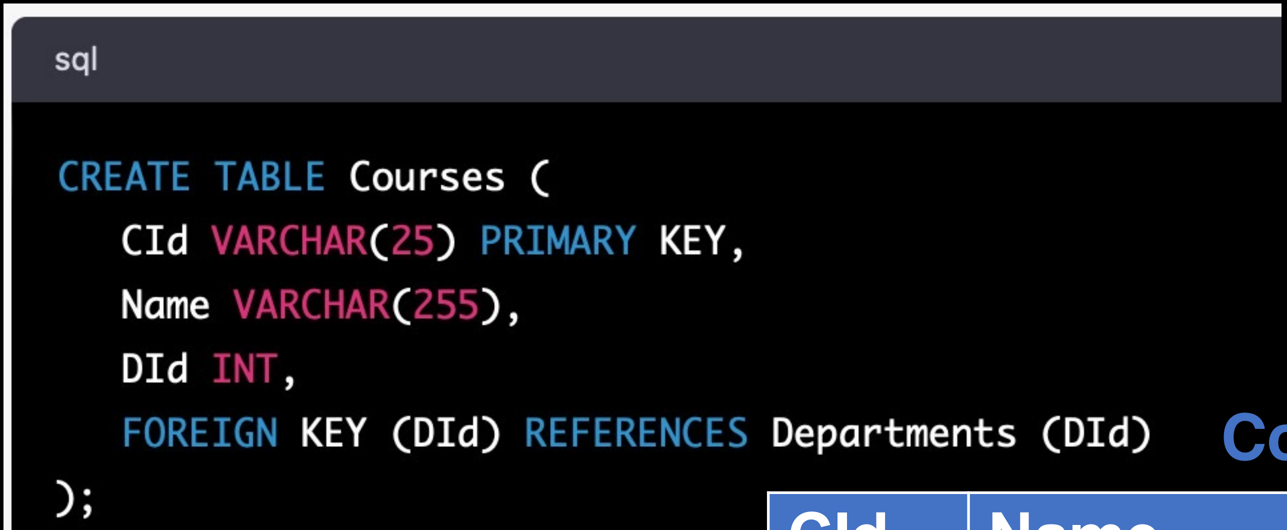


sql

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

Departments

DId	Name



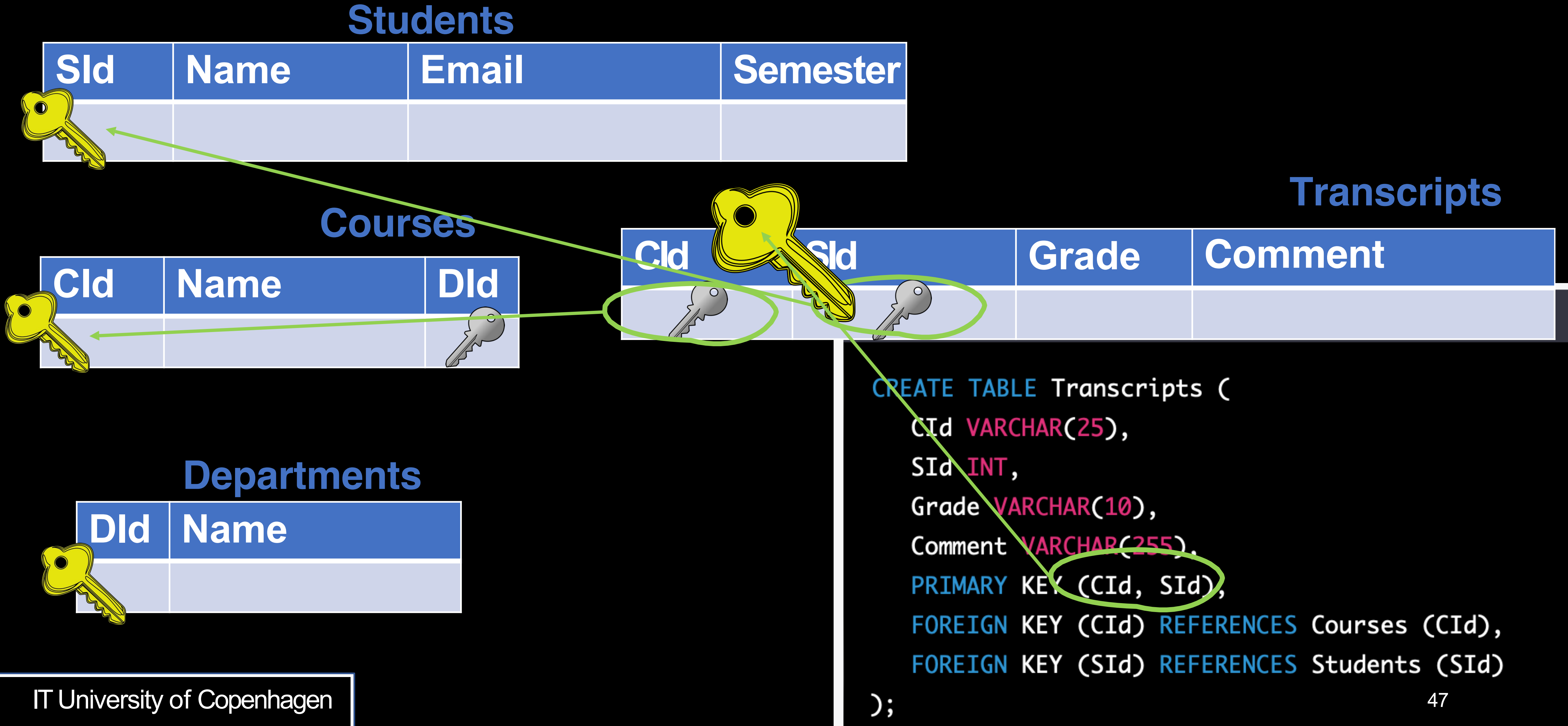
sql

```sql
CREATE TABLE Courses (
    CId VARCHAR(25) PRIMARY KEY,
    Name VARCHAR(255),
    DId INT,
    FOREIGN KEY (DId) REFERENCES Departments (DId)
);
```

Courses

CId	Name	DId

Multiple-Attribute Primary Keys



Students

SId	Name	Email	Semester

Courses

CId	Name	DId

Departments

DId	Name

Transcripts

CId	SId	Grade	Comment

```
CREATE TABLE Transcripts (
    CId VARCHAR(25),
    SId INT,
    Grade VARCHAR(10),
    Comment VARCHAR(255),
    PRIMARY KEY (CId, SId),
    FOREIGN KEY (CId) REFERENCES Courses (CId),
    FOREIGN KEY (SId) REFERENCES Students (SId)
);
```

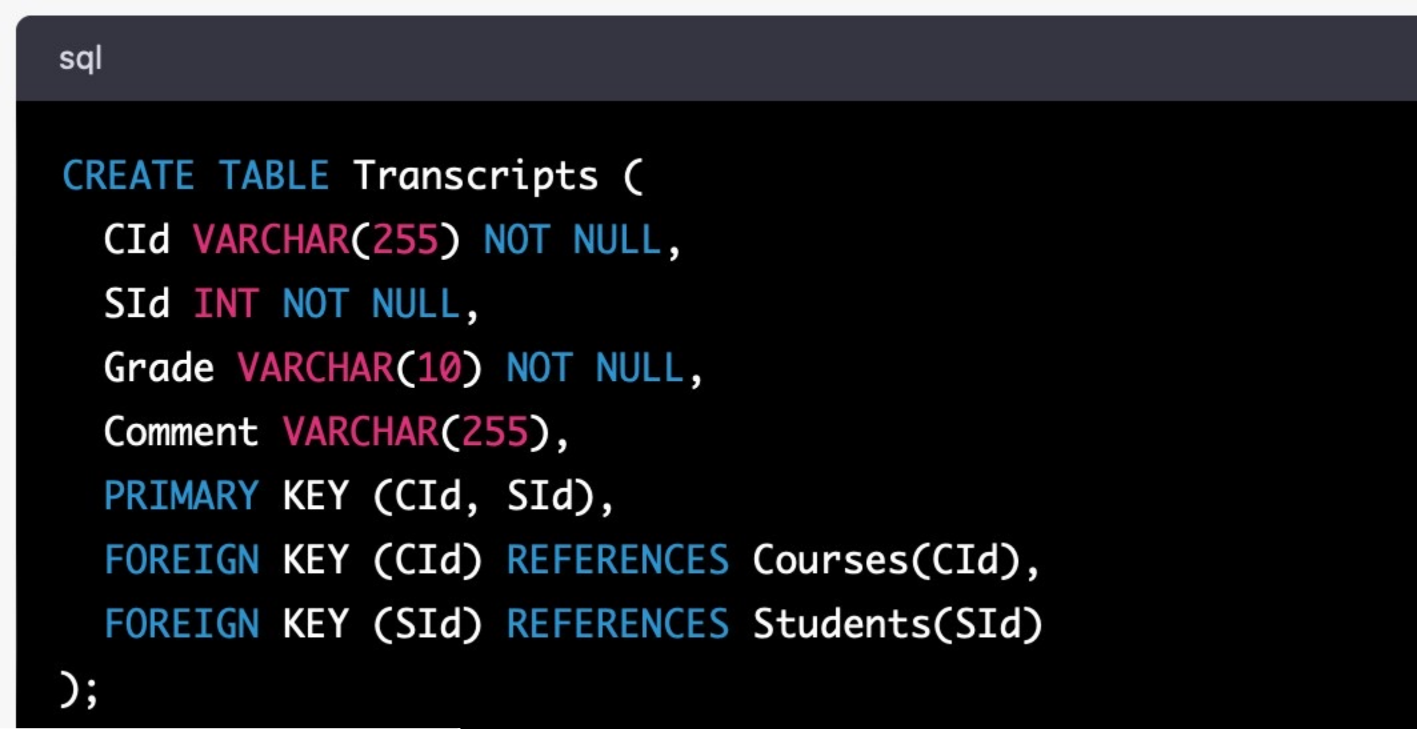
SQL -- DDL

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!**

SQL -- DDL

Relation Students with Not Nulls



sql

```sql
CREATE TABLE Transcripts (
  CId VARCHAR(255) NOT NULL,
  SId INT NOT NULL,
  Grade VARCHAR(10) NOT NULL,
  Comment VARCHAR(255),
  PRIMARY KEY (CId, SId),
  FOREIGN KEY (CId) REFERENCES Courses(CId),
  FOREIGN KEY (SId) REFERENCES Students(SId)
);
```

Transcripts

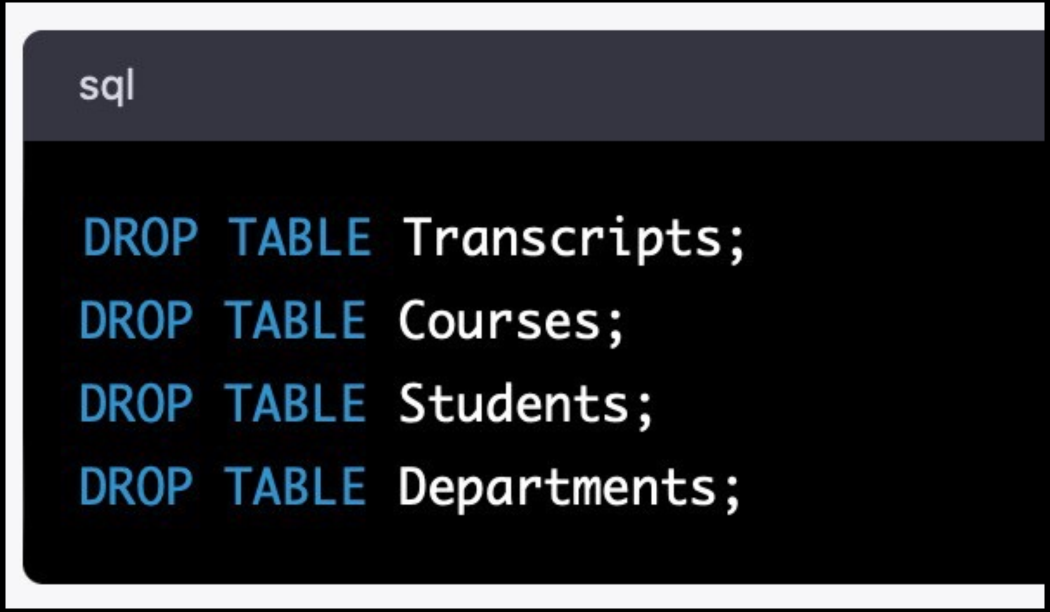
CId	SId	Grade	Comment
🟥	🟥	🟥	🟩

🟥 Cannot be null

🟩 Can be null

SQL -- DDL

Drop All Created Relations



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

Takeaways

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 actions
- (*we will dive into it next week*)

What is next?

Next week: SQL DML

- SQL DML basics, joins, aggregates, grouping, …

Install PostgreSQL

- … if you have not already done so
- Problems: get help during the exercise session!

Exercises today

1. Scripts to start playing on LearnIT (DB install script and queries)
2. Create a sample “Coffee” database
   - Use commands from slides provided on LearnIT
   - Write some INSERT and SELECT statements
   - Play with constraints
3. Consider databases without a DBMS