Databases

BCS1510

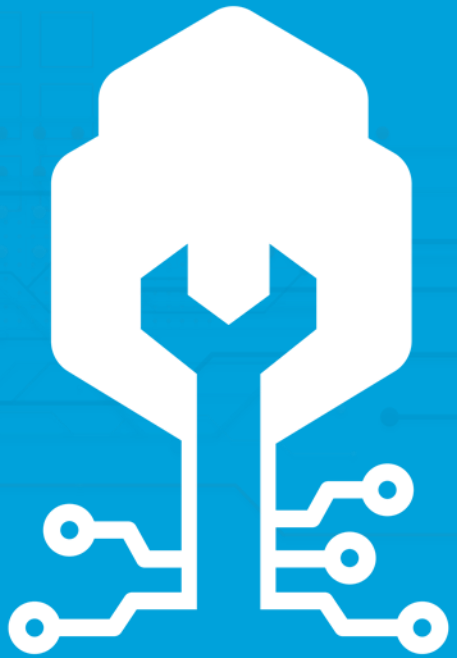
Dr. Katharina Schneider &
Dr. Tony Garnock-Jones*

🗓️ Week 1 - Lecture 2

📍 EPD150 MSM Conference Hall

Adapted with gratitude from the original lectures by Dr. Ashish Sai

*CC both of us when emailing please!



Maastricht University

What have we looked at so far?

- Databases:
  - Collection of logically related data.
  - Two formal definitions

- Database Management System:
  - Program used for defining and maintaining this database

- DBMS Evolution
  - From file systems to blockchains and big data.

Learning Objectives

- By the end of this lecture, you should be able to:
  1. Understand what a data model is
     - Explain the difference between a conceptual and a physical data model
  2. Understand what a relational model is
     - Explain the terms relation schema, (relational) database schema
     - Understand and apply the core operations in relational algebra
  3. Use SQL as Data Definition Language (DDL)
     - Use the CREATE command to create relations
     - Use the ALTER command to modify relations
     - Use the DROP command to delete relations
     - Use the PRIMARY KEY constraint
     - Use the UNIQUE constraint
     - Use the different data types

Data Models

Data Models

- A database schema is written using a DDL (Data Definition Language)
  - In fact, in the DDL of a particular DBMS

- Too low level to describe the data requirements in an understandable way

- We require a higher-level description of the schema – Conceptual Data Model

What is a Data Model?

- A data model is a notation for describing data or information

- The description consists of three parts:
  - Structure of the data
  - Operations on the data
  - Constraints on the data

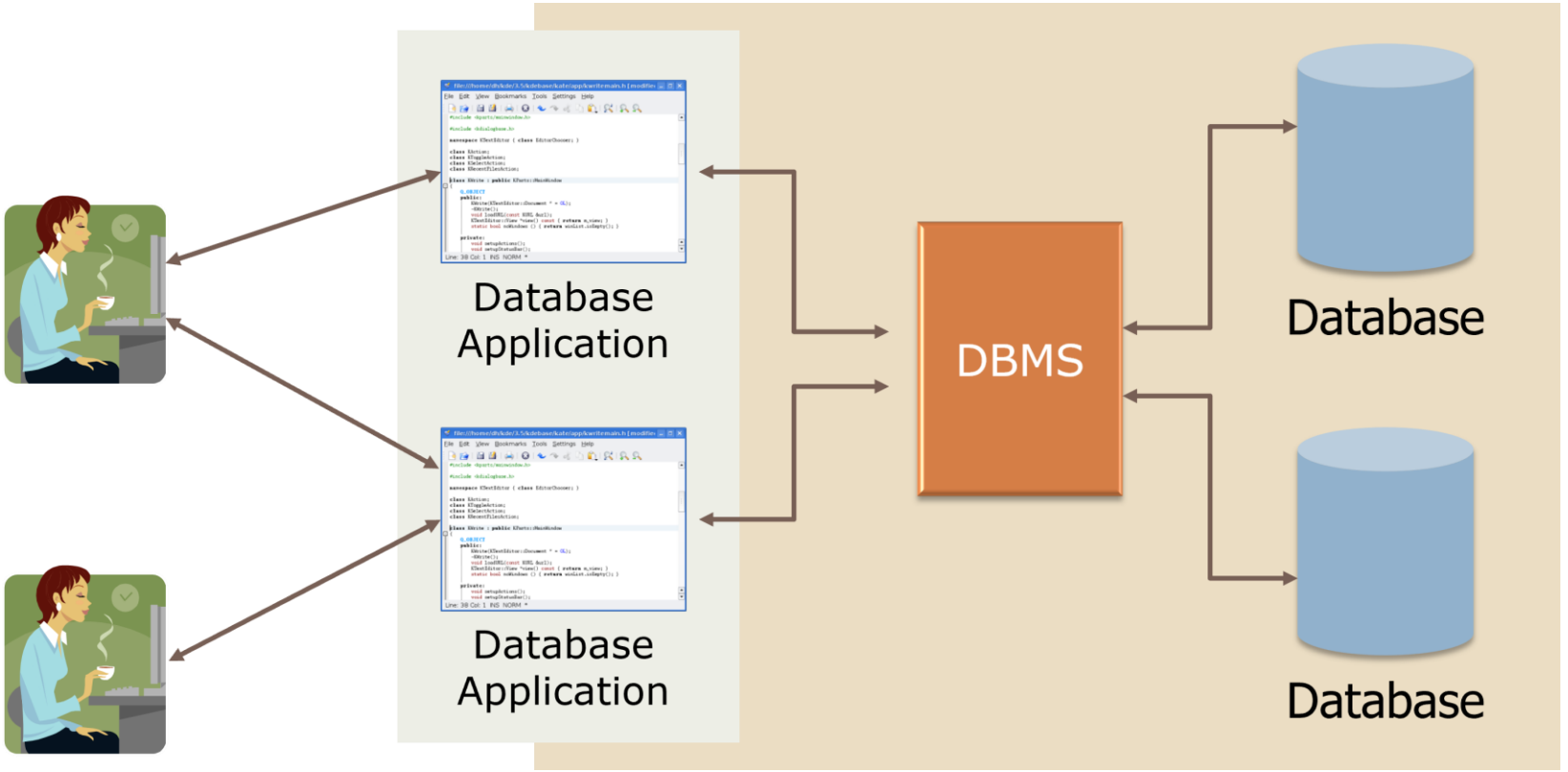
Physical Data Models

- Data structures used to implement data in the computer are referred to as physical data model
- Define how data is actually stored, accessed and managed in a specific DBMS
- Low-level detailed
- Used by Database admins and developers

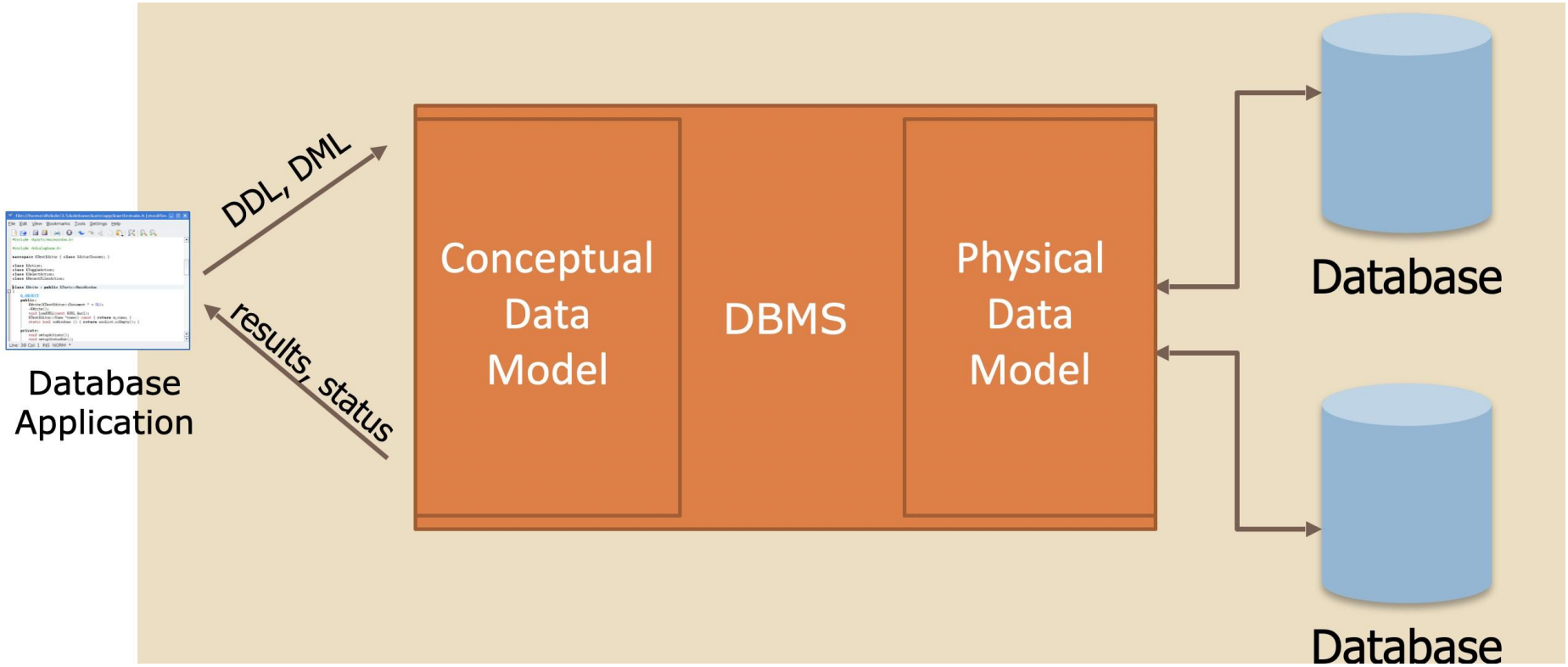
Conceptual Data Models

- In database world, data models are at a higher level than data structure and are referred to as conceptual model
- Focusing on business requirements and data relationships without considering technical implementation
- Used by Business stakeholders, analysts, designers

Remember


Database Application
Database Application
DBMS
Database
Database

Conceptual and Physical Data Models



What is a Data Model?

- A data model is a notation for describing data or information

- The description consists of three parts:
  - Structure of the data
  - Operations on the data
  - Constraints on the data

Operations on the data

- In programming: operations can be anything that can be programmed (e.g. program to replace every 10^{th} occurrence of ‘a’ with ‘z’)

- In database data models: operations limited to a set of queries (operations that retrieve information) and modifications (operations that change the database)

What is a Data Model?

- A data model is a notation for describing data or information

- The description consists of three parts:
  - Structure of the data
  - Operations on the data
  - Constraints on the data

Constraints on the data

- A way of describing limitations on data by the database data model (e.g. a person cannot have a ‘-2’ as an age)

Different Types of Data Models

- Conceptual
  - Entity-relationship
  - Relational
  - Hierarchical
  - Network
  - Graph
  - Key-value
  - Object-based
  - Document-based
    - XML
    - JSON
- Physical
  - Unifying model
  - Frame model

Relational Model

Why Relations?

- Very simple model
- Often matches how we think about data
- Relational Model is the data model that underlies SQL, the most popular database language today
- Relations (plus the operations with relations) form an algebra (relational algebra – well studied by database theorists)

Relational Model

- Gives us a single way to represent data: as a two-dimensional table called a relation

Title	Year	Length	Genre
Batman Begins	2005	220	Fantasy
Gone with the wind	1939	231	Drama
Star Wars	1977	124	sciFi

Movies

Relational Model

- Attributes – Columns: Title, Year, Length, Genre
- Tuples – Rows: Batman begins, Star wars, …
- Relation Name - Movie

Title	Year	Length	Genre
Batman Begins	2005	220	Fantasy
Gone with the wind	1939	231	Drama
Star Wars	1977	124	sciFi

Movies

Relation Schema

- Name of a relation and the set of attributes

Movies(title, year, length, genre)

OR

Movies(title:String, year:integer, length:integer, genre:String)

Database Schema

- Relational Model: a database consist of one or more relations

- Set of relation schemas (each with distinct name) is called a relational database schema (or just a database schema)

- Database = collection of relations

Remember: Why Relations?

- Very simple model
- Often matches how we think about data
- Relational Model is the data model that underlies SQL, the most popular database language today
- Relations (plus the operations with relations) form an algebra (relational algebra – well studied by database theorists)

What is an “Algebra”?

- Mathematical system consisting of
  - **Operands**: variables or values from which new values can be constructed
  - **Operators**: symbols denoting procedures that construct new values from given values

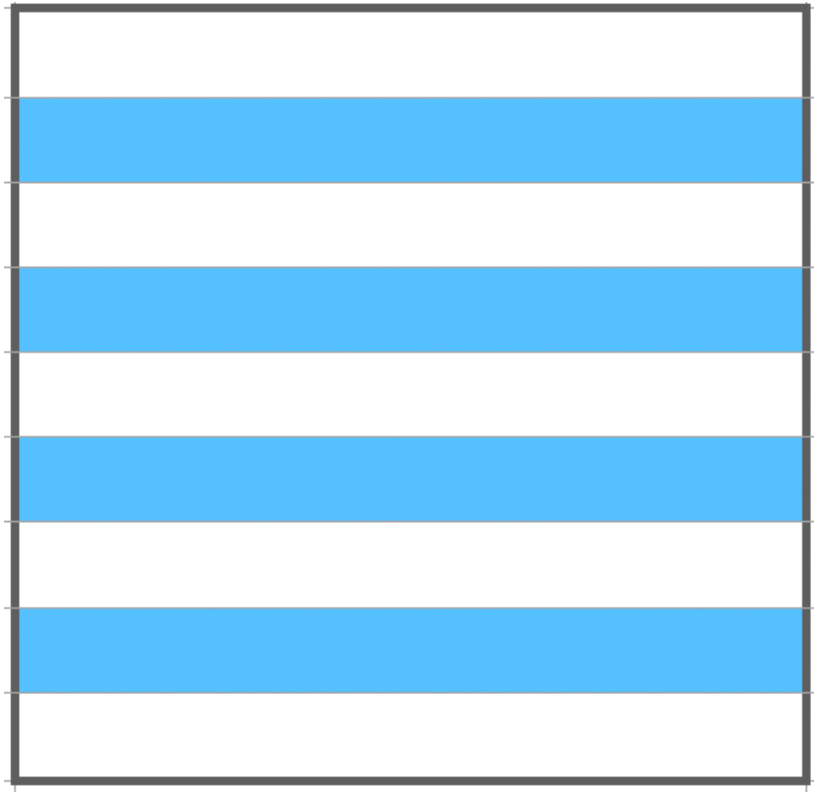
What is Relational Algebra?

- Algebra whose *operands are relations or variables that represent relations*
- Operators are designed to do the most common things that we need to do with relations in a database
- Can be used as a query language for relations

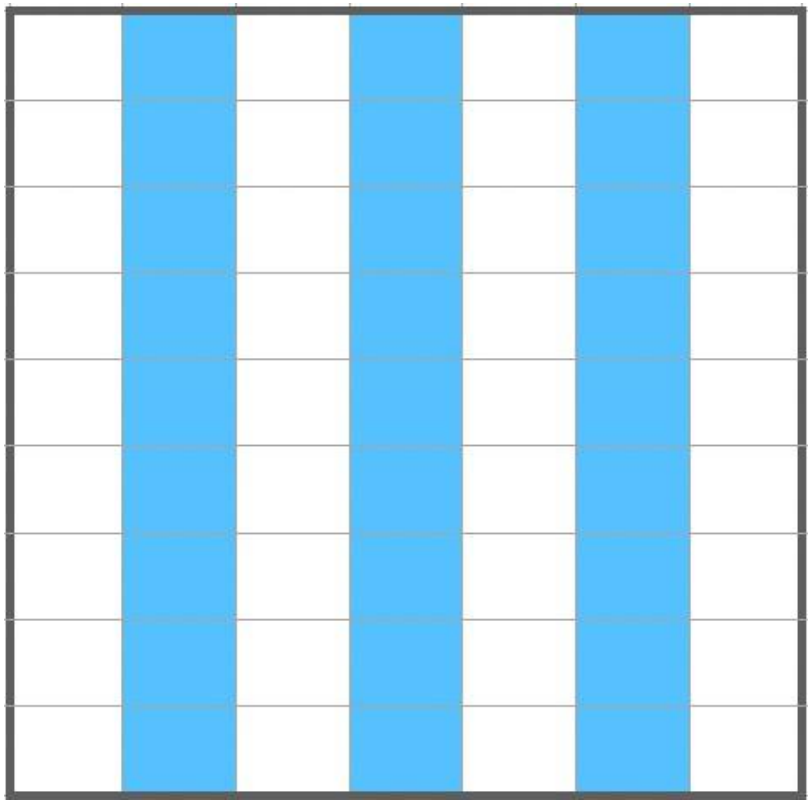
Core Relational Algebra Operators

- Selection: picking certain rows
- Projection: picking certain columns
- Product: Cartesian product of two relations (considered as sets of tuples)
- Union, intersection, and difference
  - Usual set operations, but both operands must have the same relation schema
- Renaming of relations and attributes

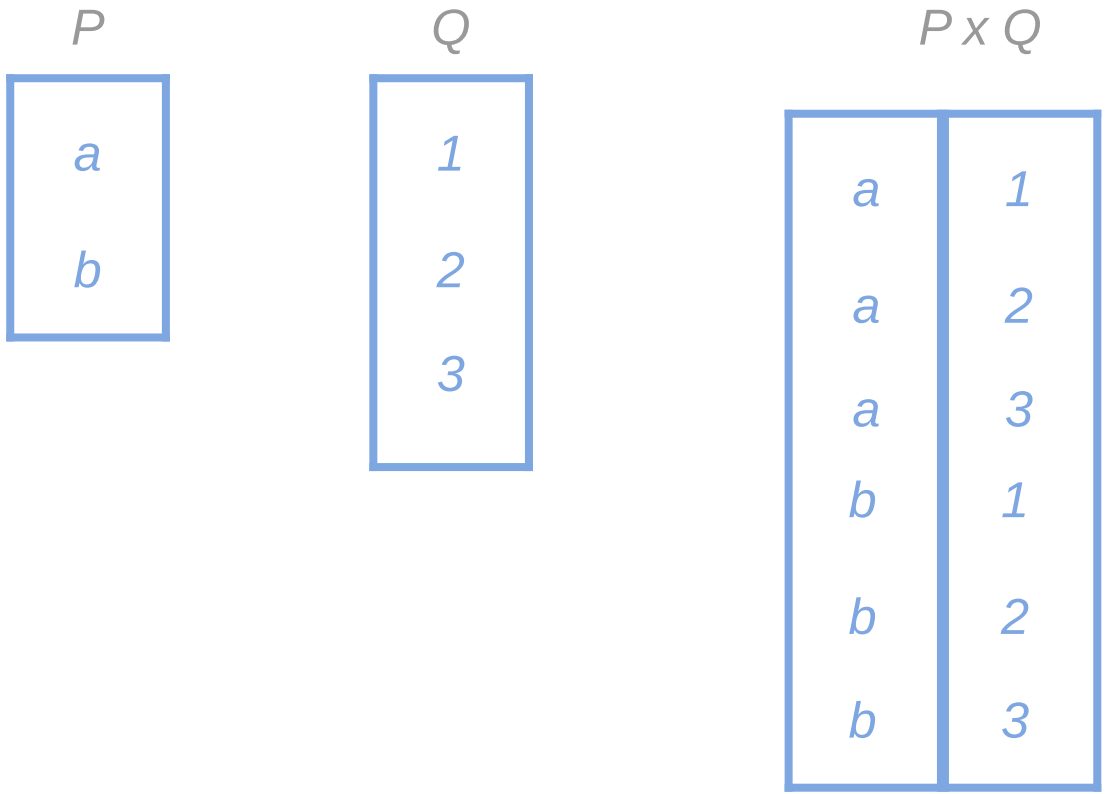
Selection



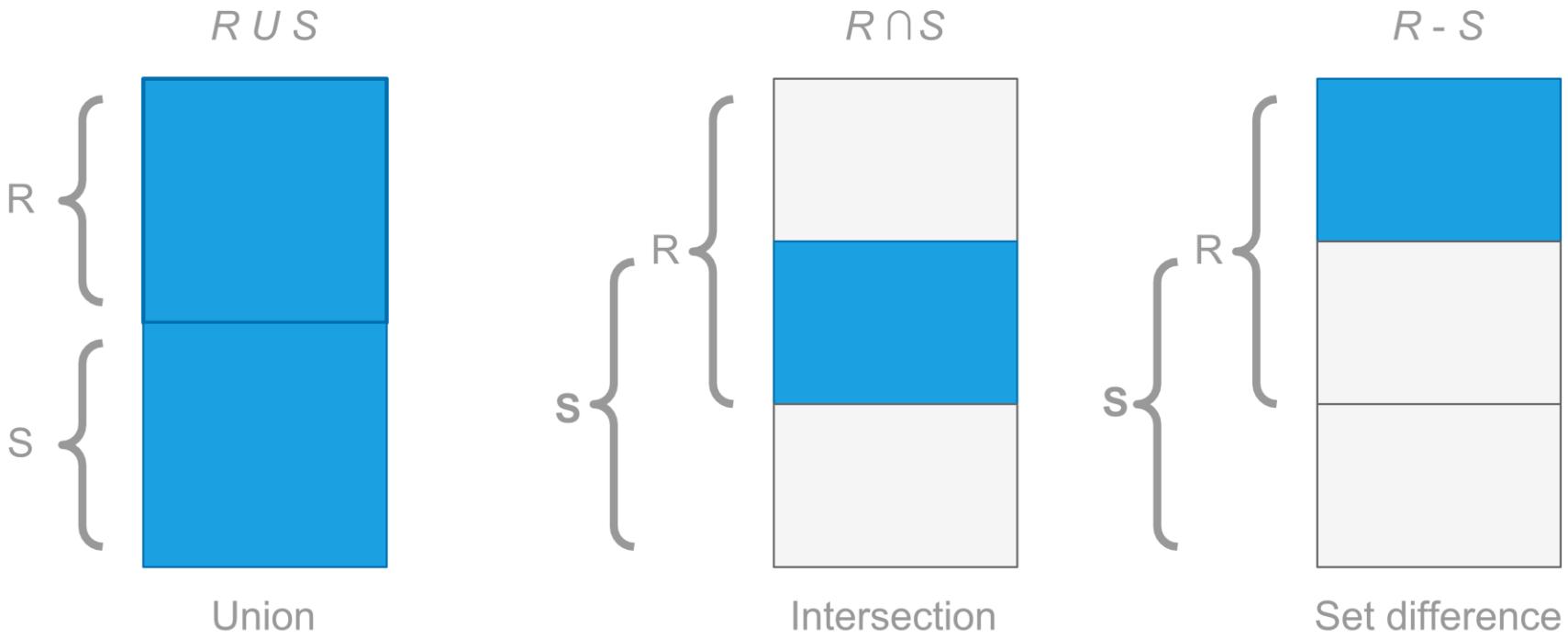
Projection



Product



Union, Intersection, and Difference



Make sure to attend the lab

- We did not cover any notation to use in Relational Algebra in this class

- This will be done in the lab on Friday

Remember: Why Relations?

- Very simple model
- Often matches how we think about data
- Relational Model is the data model that underlies SQL, the most popular database language today
- Relations (plus the operations with relations) form an algebra (relational algebra – well studied by database theorists)

Short Break

- **10** minutes

SQL (Part 1)

Pronunciation of SQL

- Option 1: "Ess-Queue-Ell" (S-Q-L)
- Option 2: "Sequel“
- Both are correct

General Information

- SQL is a very high-level language
  - Say “what to do rather” than “how to do it”
  - Avoid a lot of data-manipulation details needed in procedural languages like Java or C++

- DBMS figures out “best” way to execute query (query optimization)

- Latest version of the standard: SO/IEC 9075-1:2023 https://www.iso.org/standard/76583.html

Our Running Example: PC Shop

All (or most) upcoming SQL queries will be based on the following database schema:

- Products(maker, model, type)
  - PCs(model, speed, ram, hd , price)
  - Laptops(model, speed, ram, hd , screen, price)
  - Printers(model, color , type, price)
  - Customers(customer_id, firstname , lastname , city, address, email)
  - Sales(customer_id , model, quantity, day , paid, type_of_payment)

Our Running Example: PC Shop – Primary Key

Underline indicates primary key attributes (no tuples (i.e. data rows) of the relation may have the same values for the attributes)

- Products(maker, model, type)
  - PCs(model, speed, ram, hd, price)
  - Laptops(model, speed, ram, hd, screen, price)
  - Printers(model, color , type, price)
  - Customers(customer_id, firstname, lastname, city, address, email)
  - Sales(customer_id, model, quantity, day , paid, type_of_payment)

PCs

model	speed	ram	hd	price
1001	2.66	1024	250	2114.00
1002	2.10	512	250	995.00
1003	1.42	512	80	478.00
1004	2.80	1024	250	649.00
1005	3.20	512	250	630.00
1006	3.20	1024	320	1049.00
1007	2.20	1024	200	510.00
1008	2.20	2048	250	770.00
1009	2.00	1024	250	650.00
1010	2.80	2048	300	770.00
1011	1.86	2048	160	959.00
1012	2.80	1024	160	649.00
1013	3.06	512	80	529.00

Printers

model	color	type	price
3001	TRUE	ink-jet	99.00
3002	FALSE	laser	239.00
3003	TRUE	laser	899.00
3004	TRUE	ink-jet	120.00
3005	FALSE	laser	120.00
3006	TRUE	ink-jet	100.00
3007	TRUE	laser	200.00

Sales

customer_id	model	quantity	day	paid	type_of_payment
9999999999	1007	1	2013-12-20	459.00	visa credit
1234567890	1001	1	2013-12-20	1902.60	mastercard credit
9999999999	3007	2	2013-12-20	360.00	visa credit
9876543210	1007	3	2013-12-19	1530.00	visa debit
1122334455	2010	1	2013-12-19	2300.00	mastercard credit
1231231231	2002	2	2013-12-19	1898.00	visa credit
1122334455	3001	1	2013-12-18	99.00	cash
1231231231	3002	1	2013-12-18	239.00	cash
9876543210	1007	1	2013-12-17	510.00	visa debit
9876543210	2002	1	2013-12-17	949.00	visa debit

Customers

customer_id	firstname	lastname	city	address	email
9999999999	Norah	Jones	Limerick	2 Thomas St.	nj@yahoo.com
1234567890	Paul	Murphy	Cork	20 O'Connell St.	NULL
1122334455	Ann	O'Brien	Dublin	1 Jervis St.	aob@ul.ie
1231231231	John	Doe	Limerick	NULL	NULL
9876543210	Jack	Murphy	Galway	101 O'Connell St.	jm@ul.ie

Products

maker	model	type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
D	1008	pc
D	1009	pc
D	1010	pc
D	3004	printer
D	3005	printer

maker	model	type
E	1011	pc
E	1012	pc
E	1013	pc
E	2001	laptop
E	2002	laptop
E	2003	laptop
E	3001	printer
E	3002	printer
E	3003	printer
F	2008	laptop
F	2009	laptop
G	2010	laptop
H	3006	printer
H	3007	printer

Laptops

model	speed	ram	hd	screen	price
2001	2.00	2048	240	20.1	3673.00
2002	1.73	1024	80	17.0	949.00
2003	1.80	512	60	15.4	549.00
2004	2.00	512	60	13.3	1150.00
2005	2.16	1024	120	17.0	2500.00
2006	2.00	2048	80	15.4	1700.00
2007	1.83	1024	120	13.3	1429.00
2008	1.60	1024	100	15.4	900.00
2009	1.60	512	80	14.1	680.00
2010	2.00	2048	160	15.4	2300.00

SQL as Data Definition Language (DDL)

- SQL is primarily a query language, for getting information from a database

- But SQL also includes a data-definition component for describing database schemas

Remember

- Relation schema = relation name and attribute list + Optionally: types of attributes
- Database schema = set of all relation schemas in the database.
- Database = collection of relations.

SQL: Create

Creating (Declaring) a Relation

- Simplest form is:
- Most basic element:
  - Name of an attribute and its type

```
CREATE TABLE <name> (
<list of elements >
);
```

Example:

- QL names/identifiers must be:
  - No longer than 128 characters
  - Must start with a letter
  - Cannot contain spaces

```
CREATE TABLE Products (
      maker CHAR (1),
      model CHAR (4),
      type VARCHAR (8)
);
```

ISO SQL Data Types (often used)

- Boolean: BOOLEAN
- Character:
  - CHAR(n) = fixed-length string of n characters
  - VARCHAR(n) = variable-length string of up to n characters
- Numeric:
  - NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE PRECISION
- Datetime:
  - DATE, TIME, TIMESTAMP

ISO SQL Data Types

- Standard data types: SQL Data Types

- Data types in various DBMSs:
  - http://www.w3schools.com/sql/sql_datatypes.asp

Running Example: PC Shop

```
CREATE TABLE Products (
      maker CHAR (1),
      model CHAR (4),
      type VARCHAR (8)
);
```

Products

maker	model	type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop

Running Example: PC Shop

```
CREATE TABLE Customers (
        customer_id   CHAR (10),
        firstname     VARCHAR (32),
        lastname      VARCHAR (32),
        city      VARCHAR (32),
        address     VARCHAR (128),
        email     VARCHAR (128)
);
```

Customers

customer_id	firstname	lastname	city	address	email
9999999999	Norah	Jones	Limerick	2 Thomas St.	nj@yahoo.com
1234567890	Paul	Murphy	Cork	20 O'Connell St.	NULL
1122334455	Ann	O'Brien	Dublin	1 Jervis St.	aob@ul.ie
1231231231	John	Doe	Limerick	NULL	NULL
9876543210	Jack	Murphy	Galway	101 O'Connell St.	jm@ul.ie

Running Example: PC Shop (continued)

```
CREATE TABLE Sales (
        customer_id    CHAR (10),
        model    CHAR (4),
        quantity    INTEGER,
        day    DATE,
        paid    REAL,
        type_of_payment    VARCHAR (32)
);
```

Sales

customer_id	model	quantity	day	paid	type_of_payment
9999999999	1007	1	2013-12-20	459.00	visa credit
1234567890	1001	1	2013-12-20	1902.60	mastercard credit
9999999999	3007	2	2013-12-20	360.00	visa credit
9876543210	1007	3	2013-12-19	1530.00	visa debit

SQL Values

- Integers and reals (i.e. floating point numbers) are represented as you would expect

- Strings are too, except they require single quotes
  - Two single quotes = real quote, e.g. ‘O’’Brien’

- Any value can be NULL

SQL Values (continued)

- Dates and Times
  - DATE
    - form of a date value `yyyy -mm- dd`
  - TIME
    - form of a time value `hh:mm:ss`
  - TIMESTAMP
    - Date and time together `yyyy -mm- dd  hh:mm:ss`

Declaring Keys

- An attribute or list of attributes may be declared PRIMARY KEY or UNIQUE

- Either says that no two tuples (i.e. data rows) of the relation may have the same values for the attribute(s) on the list.

- There are a few distinctions to be mentioned later

Declaring Single-Attribute Keys

- Place PRIMARY KEY or UNIQUE after the type in the declaration of the attribute

- Example:

```
CREATE TABLE Products (
      maker CHAR (1),
      model CHAR (4) PRIMARY KEY,
      type VARCHAR (8)
);
```

Declaring Single-Attribute Keys (continued)

- Example of two alternative keys:

```
CREATE TABLE Customers (
        customer_id  CHAR (10) PRIMARY KEY,
        firstname    VARCHAR (32),
        lastname    VARCHAR (32),
        city    VARCHAR (32),
        address    VARCHAR (128),
        email    VARCHAR (128) UNIQUE
);
```

Declaring Multiattribute Keys

- A key declaration can also be another element in the list of elements of a `CREATE TABLE` statement

- This form is essential if the key consists of more than one attribute.
  - May be used even for one-attribute keys.

Declaring Multiattribute Keys - Example

- We assume that for each model X purchased by customer Y there is one entry per day in table Sales. Thus:

```
CREATE TABLE Sales (
        customer_id    CHAR (10),
        model    CHAR (4),
        quantity    INTEGER,
        day    DATE,
        paid    REAL,
        type_of_payment    VARCHAR (32),
        PRIMARY KEY ( customer_id, model, day )
);
```

PRIMARY KEY vs. UNIQUE

- There can be only one PRIMARY KEY for a relation, but several UNIQUE attributes

  - No attribute of a PRIMARY KEY can ever be **NULL** in any tuple.

  - But attributes declared UNIQUE may have NULL values, and there may be several tuples with NULL.

Altering and Deleting Relations

- To alter the schema of a relation:
  - Add a column

```
ALTER TABLE Customers
ADD date_of_birth DATE;
```

  - Drop a column

```
ALTER TABLE Customers
DROP COLUMN date_of_birth;
```

- To delete a relation

```
DROP TABLE Customers;
```

Contact details

- If you have any issues throughout the course, contact us via email:
- Tony: tony.garnock-jones@maastrichtuniversity.nl
- Katharina: k.schneider@maastrichtuniversity.nl
- Or send a message on Discord
- (https://discord.gg/Be2KSF8QG6)

Questions?

See you next week