

Data Modelling & Databases

Janis Hutz
<https://janishutz.com>

June 22, 2026



“A curry can be written in many different ways”

- Prof. Dr. Gustavo Alonso, 2026

FS2026, ETHZ

Summary of the Lecture Slides

<https://systems.ethz.ch/education/courses/2026-spring/dmdb.html>

Contents

1	Introduction	3
1.1	IMS	3
1.2	Network Model	3
1.3	Relational Model	3
2	SQL	4
3	Relational Logic & Algebra	5
	Glossary	6

1 Introduction

A note on the quote on the title page: He did not *actually* mean “curry”, but of course “query”, but he just pronounced it that way.

This summary assumes some basic knowledge about databases. All words written in **bold** should be clickable and should link you to the corresponding term in the glossary. If I did not forget a term that is, of course.

1.1 IMS

Early databases were not relational and they also required the user of the system to know *how the data was stored*. This of course is not ideal, as any update to the data structure also required an update to the code interacting with it. In addition, manual query optimization has to be done.

Non-relational databases commonly have data redundancy, due to non-hierarchical applications being forced into a tree-based model. This causes issues when updating the data, as one needs to update the data in all places where a copy of the data exists.

1.2 Network Model

In a network model, one record is fetched at a time, so it essentially traverses the network. However, we again do not have data independence, thus the application again needs to change when the physical data representation changes. In addition, manual query optimization has to happen again, thus, any change to the physical data representation may cause the need to redo the optimization.

1.3 Relational Model

This is the focus of this course

2 SQL

SQL is a declarative programming language, used to describe and operate on databases.

Keyword	Description
SELECT val, val2	returns the columns val and val2
FROM table d, table2 c	selects the target tables. Can use range variables (d, c here)
WHERE	Filter results. Can also target non-acquired columns
DISTINCT	Returns elements without duplicates
IN	Checks if the element is present in subquery
UNION	Set union (also removes duplicates)
INTERSECT	Set intersection
EXCEPT	Set difference
JOIN	Join tables, always requires a ON clause
ON	Specify on which column with what condition to join the tables
DATE	Used to do comparisons against fixed Dates (followed by Datestring)
AS	Rename a column
AVG(column)	Computes the average of the specified column
SUM(column)	Computes the sum of the specified column

We can directly add new columns to our result using statements like (also note that we do not actually return salary and employment columns)

```
SELECT name, salary * employment / 100 AS RealSalary FROM employees;
```

When applying GROUP BY, COUNT and the like operate on each group, not entire dataset.

NATURAL JOIN doesn't require a ON clause (it looks for columns with same name)

After GROUP BY we can't use WHERE, instead we can use HAVING, which operates on groups instead of on rows.

We can compare against a query with a single result column and row using a WHERE clause.

The CROSS JOIN is the both-sided extension to LEFT JOIN and RIGHT JOIN

We can compute the number of all passed grades for a person using something like

```
SUM(CASE Grade >= 1 THEN 1 ELSE 0 END) AS PassedSubjectsCount
```

3 Relational Logic & Algebra

Glossary

declarative Describes *what* to achieve, but not how.

imperative Describes how to achieve something.

record .

SQL Structured Query Language.