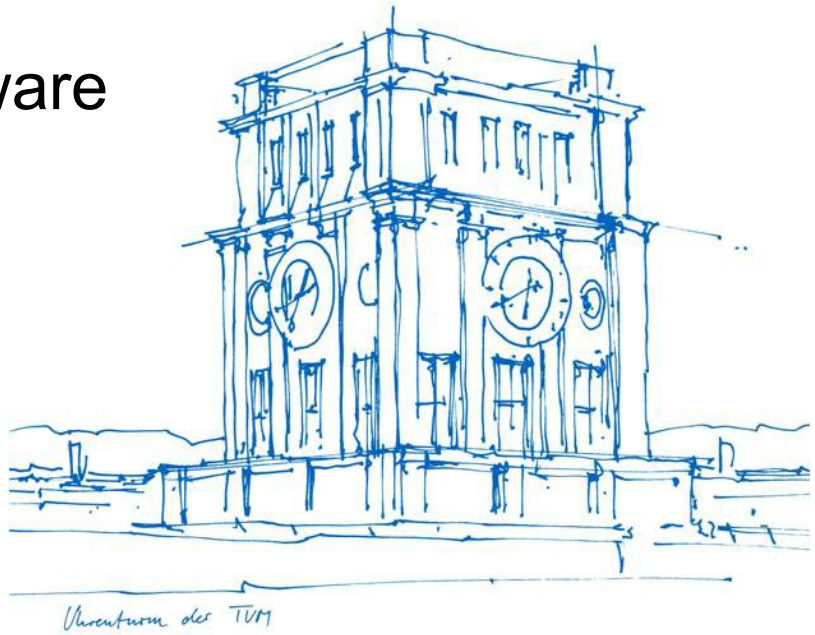
Computational Databases: Inspirations from Statistical Software

Linnea Passing, linnea.passing@tum.de

Technical University of Munich



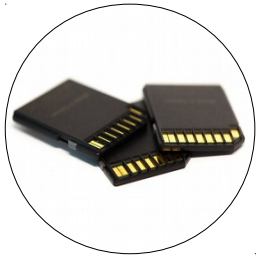
Data Science Meets Databases

Data Cleansing

- Pipelines
- Fuzzy joins
- Data input errors
- e.g., Alteryx

Statistics

- Pre-tests
- Handling missing data
- e.g., SPSS, R



Big Data, BI

- OLAP cubes
- Joins and aggregation
- Roll-up and drill-down
- e.g., Spark, DBS

Data Mining

- Classification
- Clustering
- Forecasting
- Regression
- e.g., R, python

ML, AI

- Neural networks
- Linear algebra
- e.g., TensorFlow

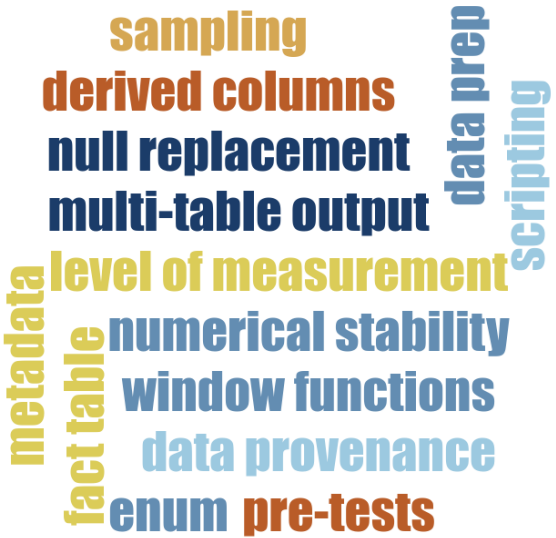
Inspirations from Statistical Software

Similarities

- Connecting to flat files, RDBMS, Big Data environments
- Fixed-point numerics as main datatype
- Extensions for e.g. geospatial and streaming data

Differences

- New queries (e.g. statistical properties)
- Higher quality requirements (e.g. numerical stability)
- New workflows (e.g. metadata output)
- Richer semantics (e.g. handling missing values)



OPTIMIZING COMMON QUERIES

QUALITY IMPROVEMENTS

WORKFLOWS

SELF-CONTAINED DATABASE

Schema, Joins, Enums

Inspirations from statistical software

One big fact table, user-defined enum datatypes[1]

Derived columns are added to the fact table

State-of-the-art in database systems

Normalized schema with fact and dimension tables (avoid anomalies, keep database small, …)

Derived columns can be added to (materialized) views (then joins are required to query all columns) or via

```
ALTER TABLE ADD COLUMN
```

Take-away

Column stores: light-weight `ALTER TABLE ADD COLUMN`, maybe even for non-materialized columns?

[1] e.g. education: {(0, none), (1, highschool diploma), (2, college degree), (3, PhD degree), ...}

Cumulative and total values

Inspirations from statistical software
E.g. variance explained using 1, 2, 3, … factors
E.g. number of participants per age group **and total**

State-of-the-art in database systems
Running sums (cumulative values) supported via window functions
No native support for analyses on multiple levels in one query, one has to use multiple `GROUP BY` expressions and `UNION ALL` or non-standardized `ROLLUP/CUBE`

Take-away
Combination of rollup/cube (implemented as part of `GROUP BY`) and windows required

Common Statistical Properties/Pretests

Inspirations from statistical software

Often required for statistical methods

E.g. homoscedasticity[1] for Pearson correlation coefficient, normal distribution for T-test

State-of-the-art in database systems

Small materialized aggregates for common aggregates (`SUM`, `MIN`, ...)

Take-away

Small materialized aggregates for common statistical properties/pre-tests, also spanning multiple columns

e.g. standard deviation of column x; covariance of columns x,y

[1] variance around regression line is the same for all values, not given in e.g. weather forecast data

OPTIMIZING COMMON QUERIES

QUALITY IMPROVEMENTS

WORKFLOWS

SELF-CONTAINED DATABASE

High-quality Sampling Techniques

Inspirations from statistical software

For running expensive statistical simulations on representative subsets

To adjust weights for underrepresented participant groups in a survey

State-of-the-art in database systems

Often only low-quality sampling using `LIMIT` or `WHERE (ROWID%X)=0` or expensive `ORDER BY RAND()` sampling is supported

Take-away

Providing high-quality sampling techniques that go with the query execution flow of databases

Numerical Stability

Inspirations from statistical software
Often deals with small (e.g. z-transformed) numbers
Stable two-pass implementation

State-of-the-art in database systems
AVG rewritten to SUM/COUNT
Naive one-pass implementation of covariance

Take-away
Stable implementations require re-streaming or materialization of input data which is expensive

OPTIMIZING COMMON QUERIES

QUALITY IMPROVEMENTS

WORKFLOWS

SELF-CONTAINED DATABASE

Data Provenance and Scripts

Inspirations from statistical software

A single GUI action often corresponds to multiple queries, e.g. adding statistical pre- and post-tests

Running algorithms often to overcome local minima

Interactive data preparation

State-of-the-art in database systems

Transactional guarantees allow executing pre-tests and actual actions on exactly the same data

Take-away

Capture and replay of queries

Provenance metadata attached to tables and views

Multiple Output Tables

Inspirations from statistical software

One action/query often results in multiple output tables, e.g. containing additional statistics, results of pre- and post-tests, comments about result quality, footnotes

State-of-the-art in database systems

The output of an operator/query can only be one relation. Thus while many temporary tables can be created using common table expressions, a query can only materialize/output **one** table

Take-away

Being able to output multiple relations?

OPTIMIZING COMMON QUERIES
QUALITY IMPROVEMENTS
WORKFLOWS
SELF-CONTAINED DATABASE

Handling Missing Values / Replacing `NULL` values

Inspirations from statistical software

Replacement value for missing data is stored as part of the schema and applied throughout all functions that use the column

State-of-the-art in database systems

Replacement values for missing data can only be given per query, using `COALESCE` clauses, not as part of the schema

Take-away

Achieving self-contained databases, where less business logic or domain knowledge is stored solely in applications

Restricting Computations / Levels of measurement

Inspirations from statistical software

nominal (enum, e.g. gender), **ordinal** (ordered enum, e.g. educational degree), **interval** (number with arbitrary zero-point, e.g. degree Celsius), **ratio** (number with meaningful zero-point, e.g. quantity)

Operations restricted to what the *level* allows (e.g. no mean for ordinal data, no ratio for interval data)

State-of-the-art in database systems

String, bool, numbers, (sometimes enum), and more

Operations restricted to what is *mathematically* possible

Take-away

Enrich table metadata to achieve a more self-contained database

Can speed up computations, e.g. `GROUP BY` on a known-to-be limited number of groups

Summary and Outlook

Enabling new types of data scientists to do new types of analytics in computational databases

Focus on statistics also benefits other types of data scientists and workloads

Further inspirations for database systems