



Home

About

Content

Others

Capstone Project

Nico Baumann,
Marco Weber,
Lavinia Lauer

–

Datenbanken und DB-
Programmierung

[Home](#)[About](#)[Content](#)[Others](#)

Zielsetzung

Aufgabe des Projekts war es, ein minimales Analysesystem zu entwickeln, das drei konkrete, datengetriebene Unternehmensfragen beantwortet.



Capstone Projekt

[Home](#)[About](#)[Content](#)[Others](#)

Datenbank

Die Datenbank besteht aus 8 Tabellen, die die benötigten Unternehmensdaten strukturiert speichern und miteinander verknüpfen.



[Home](#)[About](#)[Content](#)[Others](#)

Stakeholder

Geschäftsführung

„In welcher Branche haben wir Kunden mit sehr hoher bzw. sehr niedriger Projektanzahl, und wo besteht Potenzial, bestehende Kunden mit wenigen Projekten auszubauen?“



Entscheidung, die die Frage unterstützen soll:

- Priorisierung von Upselling- oder Cross-Selling-Maßnahmen bei Kunden mit wenigen Projekten
- Fokus auf Branchen, in denen der Abstand zwischen Top- und Low-Client besonders groß ist

Zeitfenster:

- Historische Daten: alle abgeschlossenen und laufenden Projekte bis zum aktuellen Stichtag
- Optional: Filter nach Jahr oder Quartal für Trendanalysen

Messgröße(n):

- Anzahl der Projekte pro Kunde (project_count)
- Branchenzugehörigkeit der Kunden (industry)
- Differenz zwischen Top- und Low-Client pro Branche

benötigte Tabellen

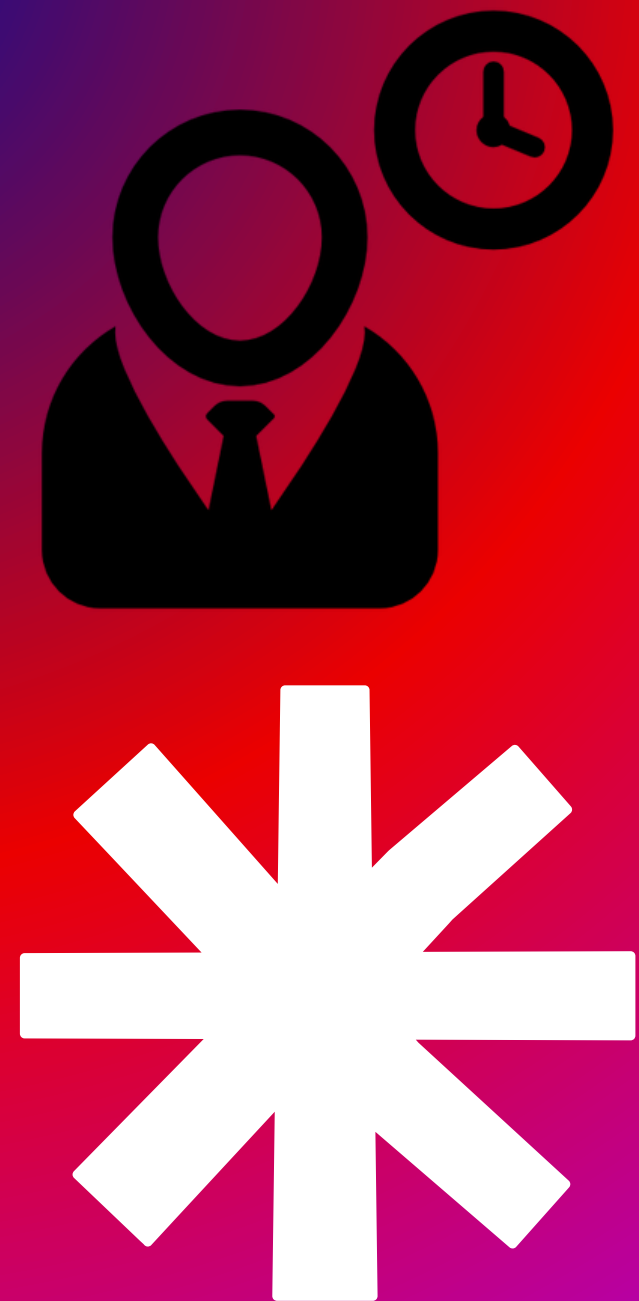
Tabelle	Felder	Zweck
clients	client_id, name, industry	Kundenstammdaten, Branchenzuordnung, Anzeige des Kunden
projects	project_id, client_id, status	Anzahl Projekte pro Kunde, Status für Filterung



Erkenntniss

In jeder Branche gibt es Kunden mit deutlich weniger Projekten. Ziel: prüfen, ob diesen Kunden zusätzliche Projekte angeboten werden können

Industry	Max Project (Count)	Min Project (Count)
Aerospace	QuantumPeak Software (17)	Apexion Labs (4)
Agriculture	Starwave Innovations (16)	QuantumPeak Software (3)
Automotive	NovaCore Solutions (16)	Vertexon Industries (5)
Banking	CloudForge Technologies (15)	LuminaWorks (3)
Biotechnology	QuantumPeak Software (16)	Silverline Systems (4)
Chemical Industry	Everbright Analytics (17)	Greenridge Manufacturing (3)
Construction	Greenridge Manufacturing (18)	Everbright Analytics (5)
Consulting	Apexion Labs (18)	Starwave Innovations (4)
Consumer Electronics	LuminaWorks (16)	CloudForge Technologies (4)
E-Commerce	Greenridge Manufacturing (19)	Starwave Innovations (5)
Education	Greenridge Manufacturing (18)	Everbright Analytics (5)
Energy	Everbright Analytics (19)	Silverline Systems (3)
Entertainment	QuantumPeak Software (18)	Silverline Systems (3)
Environmental Services	NovaCore Solutions (21)	Vertexon Industries (5)
Fashion	Apexion Labs (17)	NovaCore Solutions (3)
Financial Services	Starwave Innovations (19)	LuminaWorks (4)
Food and Beverage	Everbright Analytics (17)	Bluecrest Dynamics (8)
Healthcare	NovaCore Solutions (24)	Everbright Analytics (6)
Hospitality	CloudForge Technologies (15)	Apexion Labs (6)
Information Technology	QuantumPeak Software (15)	CloudForge Technologies (4)
Insurance	Greenridge Manufacturing (18)	NovaCore Solutions (4)
IT	BlueWave AG (23)	
Logistics	Greenridge Manufacturing (14)	Starwave Innovations (4)
Manufacturing	Starwave Innovations (21)	LuminaWorks (4)
Marketing	LuminaWorks (20)	Greenridge Manufacturing (4)
Media	Everbright Analytics (19)	Apexion Labs (3)
Mining	QuantumPeak Software (22)	CloudForge Technologies (3)
Nonprofit	Starwave Innovations (24)	Greenridge Manufacturing (4)
Oil and Gas	NovaCore Solutions (16)	Silverline Systems (5)
Pharmaceuticals	CloudForge Technologies (19)	CloudForge Technologies (4)
Public Sector	NovaCore Solutions (14)	LuminaWorks (7)
Real Estate	Vertexon Industries (17)	Silverline Systems (4)
Renewable Energy	NovaCore Solutions (16)	CloudForge Technologies (5)
Retail	Vertexon Industries (17)	Starwave Innovations (3)
Robotics	QuantumPeak Software (19)	Silverline Systems (2)
Software Development	QuantumPeak Software (21)	QuantumPeak Software (4)
Sports	Silverline Systems (18)	Starwave Innovations (4)
Telecommunications	LuminaWorks (15)	CloudForge Technologies (5)
Tourism	Apexion Labs (17)	LuminaWorks (5)

[Home](#)[About](#)[Content](#)[Others](#)

Stakeholder Projektmanager

„Gibt es Teams, in denen einzelne Mitarbeiter deutlich weniger Tasks haben als andere? Welche Teams haben insgesamt besonders wenige Tasks?“



Entscheidung, die die Frage unterstützen soll:

- Priorisierung von Kapazitäts- oder Aufgabenmanagement-Maßnahmen
- Fokus auf Teams mit großer Differenz zwischen Top- und Low-Performer

Zeitfenster:

- Alle laufenden Aufgaben (status = 'Running') zum aktuellen Stichtag

Messgröße(n):

- Anzahl laufender Aufgaben pro Mitarbeiter (task_count)
- Teamzugehörigkeit (team_id)
- Durchschnittliche, minimale und maximale Aufgaben pro Mitarbeiter im Team
- Anzahl Mitarbeiter pro Team

benötigte Tabellen

Tabelle	Felder	Zweck
employees	employee_id, team_id, role	Enthält Informationen über die Mitarbeiter und ihre Zugehörigkeit zu Teams
tasks	task_id, assignee_id, status, created_at	Enthält Informationen zu Aufgaben und welchem Mitarbeiter sie zugewiesen sind
teams	team_id, name	Enthält Informationen über die Teams



Erkenntniss

Teams mit vielen Aufgaben pro Mitarbeiter sind oft auch die größten Teams. Die Unterschiede zwischen Teams sind teilweise sehr groß – einige Teams haben mehr als doppelt so viele durchschnittliche Aufgaben wie andere. Die Grafik zeigt die Top 10 mit den höchsten und die 10 mit den niedrigsten Durchschnittswerten. Ziel: Teamgrößen und Aufgabenverteilung gezielt optimieren

team_id	team_name	avg_task_per_employee	min_task_per_employee	max_task_per_employee	num_employees_per_team
12	Team cause	5,294117647	1	11	34
6	Team himself	5,277777778	2	11	36
9	Team put	5,166666667	1	11	36
2	Marketing	5,097560976	1	14	41
107	Team quality	4,944444444	2	10	18
3	Team tax	4,921052632	2	14	38
4	Team skin	4,757575758	1	11	33
7	Team poor	4,714285714	1	10	42
10	Team try	4,694444444	1	10	36
8	Team I	4,676470588	1	11	34
435	Team beat	2,181818182	1	5	11
287	Team become	2,333333333	1	6	18
346	Team month	2,380952381	1	6	21
517	Team job	2,388888889	1	5	18
275	Team catch	2,428571429	1	4	14
168	Team pick	2,466666667	1	6	15
75	Team as	2,473684211	1	5	19
36	Team like	2,5	1	5	10
724	Team attention	2,5	1	7	14
910	Team southern	2,555555556	1	5	18

[Home](#)[About](#)[Content](#)[Others](#)

Stakeholder

HR / Recruiting

„Gibt es Mitarbeiter oder Rollen, die in den letzten 3 Monaten deutlich mehr Stunden gearbeitet haben als der Teamdurchschnitt, um Überlastung und potenzielle Burnout-Risiken zu erkennen?“



Entscheidung, die die Frage unterstützen soll:

- Faire Arbeitslast und Überstunden-Management
- Fokus auf Mitarbeiter oder Rollen mit deutlich überdurchschnittlicher Arbeitszeit

Zeitfenster:

- Rolling 3 Monate
- Optional: Filter nach Team oder Rolle für Trendanalysen

Messgröße(n):

- Summe der Stunden pro Mitarbeiter (time_logs)
- Durchschnittliche Stunden pro Team (employees.team_id)
- Rolle des Mitarbeiters (employees.role)

benötigte Tabellen

Tabelle	Felder	Zweck
employees	employee_id, team_id, role	Informationen über Mitarbeiter und ihre Teamzugehörigkeit
tasks	task_id, assignee_id, status, created_at	Arbeitsstunden der Mitarbeiter pro Projekt
teams	team_id, name	Informationen über Teams
time_logs	employee_id, project_id, date, hours	Aufgaben der Mitarbeiter



Erkenntniss

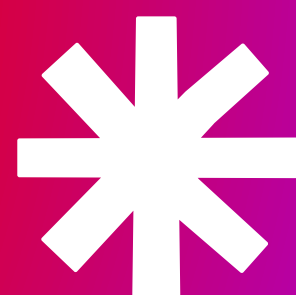
Die Analyse zeigt, dass einige Mitarbeitende deutlich mehr Arbeitsstunden leisten als andere Personen in derselben Rolle. Besonders auffällig ist, dass Software Developer sowie Mitarbeitende aus Team 7 weit über dem durchschnittlichen Stundenwert ihrer jeweiligen Rolle liegen. Diese Überlastung kann ein Hinweis darauf sein, dass bestimmte Rollen, insbesondere im Bereich Softwareentwicklung, strukturell stärker belastet sind. Daher sollte geprüft werden, ob Neueinstellungen sinnvoll wären oder ob eine Umverteilung von Aufgaben Entlastung schaffen kann.

employee_id	team_id	role	total_hours	avg_hours_per_role	deviation_from_role_avg
56	7	Software-Developer	71	26,25	44,75
47	7	Human-Resources	72	30,11	41,89
8519	924	Project-Leader	66	24,14	41,86
64	3	Manager	75	34,78	40,22
9435	791	Software-Developer	65	25	40
92	9	Manager	69	29,14	39,86
3836	509	Software-Developer	63	23,5	39,5
188	8	Human-Resources	67	28	39
3087	119	Project-Leader	61	22,67	38,33
36	2	Software-Developer	62	24	38
171	2	Manager	61	23,89	37,11
8339	374	Manager	58	21	37
5661	502	Software-Developer	55	18,83	36,17
20183	365	Software-Developer	59	23,25	35,75
14478	871	Human-Resources	57	21,43	35,57
96	7	Manager	66	31,5	34,5
32	5	Secretary	71	36,63	34,38
12621	217	Software-Developer	62	27,71	34,29
8388	285	Secretary	54	20,14	33,86
38	10	Secretary	72	38,17	33,83

[Home](#)[About](#)[Content](#)[Others](#)

SQL-Abfrage

Unoptimiert



```
WITH max_date AS (  
    SELECT MAX(date) AS latest_date  
    FROM time_logs  
),  
employee_hours AS (  
    SELECT  
        e.employee_id,  
        e.team_id,  
        e.role,  
        SUM(tl.hours) AS total_hours  
    FROM time_logs tl  
    JOIN employees e  
        ON tl.employee_id = e.employee_id  
    CROSS JOIN max_date md  
    WHERE tl.date >= md.latest_date - INTERVAL '3 months'  
    GROUP BY e.employee_id, e.team_id, e.role  
),  
team_avg AS (  
    SELECT  
        team_id,  
        role,  
        AVG(total_hours) AS avg_hours_per_role  
    FROM employee_hours  
    GROUP BY team_id, role  
)  
SELECT  
    eh.employee_id,  
    eh.team_id,  
    eh.role,  
    eh.total_hours,  
    ta.avg_hours_per_role,  
    eh.total_hours - ta.avg_hours_per_role AS deviation_from_role_avg  
FROM employee_hours eh  
JOIN team_avg ta  
    ON eh.team_id = ta.team_id  
    AND eh.role = ta.role  
WHERE eh.total_hours > ta.avg_hours_per_role  
ORDER BY deviation_from_role_avg DESC;
```

Datenvolumen:

- time_logs ~500K
- employees ~20k
- letzte 3 Monate ~85k

Performance:

- unoptimiert 134ms

Verbesserung:

- Indexes auf date und employee_id
- Direkter Max-Date-Filter

[Home](#)[About](#)[Content](#)[Others](#)

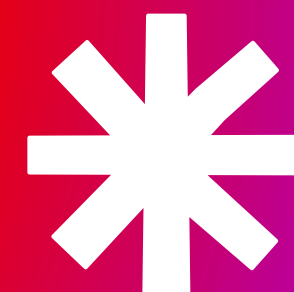
```
CREATE INDEX idx_time_logs_date ON  
time_logs(date);
```

```
CREATE INDEX idx_time_logs_employee_id ON  
time_logs(employee_id);
```

```
WHERE tl.date >= (SELECT MAX(date) FROM  
time_logs) - INTERVAL '3 months'
```

SQL-Abfrage

Optimiert



Performance:

- optimiert 85ms

Index auf date wird genutzt

- Vorher: 505.000 Rows, Seq Scan
- Jetzt: 85.422 relevante Rows, Bitmap Heap Scan → weniger I/O

MAX(date) nutzt Index Only Scan

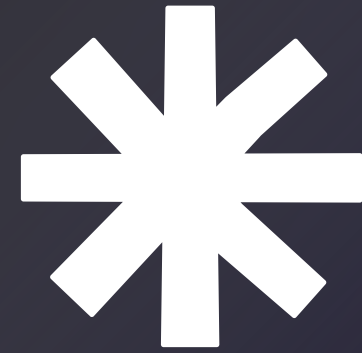
- Heap Fetches: 0 → extrem schnell (0,014 ms)
- CTE muss nicht mehr ganze Tabelle scannen

Hash Join reduziert Datenmenge

- Vorher: Full Scan + Join Filter
- Nachher: Nur 85k gefilterte Rows gejoint

[Home](#)[About](#)[Content](#)[Others](#)

FAZIT



Insgesamt empfiehlt sich ein kombiniertes Vorgehen aus Kundenpotenzialanalyse, strukturierter Teamoptimierung und gezieltem Kapazitätsaufbau, um sowohl Effizienz als auch Mitarbeitendenzufriedenheit nachhaltig zu steigern.





[Home](#)

[About](#)

[Content](#)

[Others](#)

Thank You

-