Relationale Datenbanken

Dinosaurier oder Dauerbrenner?

Folien

Alexander Seminjakiw

- 2016: M. Sc. Physik
- 2017: Entwickler bei evopro AG
- 2022: Architekt und Lead-Developer bei evopro AG
- www.linkedin.com/in/alexander-seminjakiw-b66935278
- Aufgaben:
  - Bildverarbeitung
  - Backend & Verteilte Systeme
  - Desktop Apps
  - Produktionsanlagennahe Software

Werkzeuge

Werkzeuge

Werkzeuge

Werkzeuge

Java

C#

C++

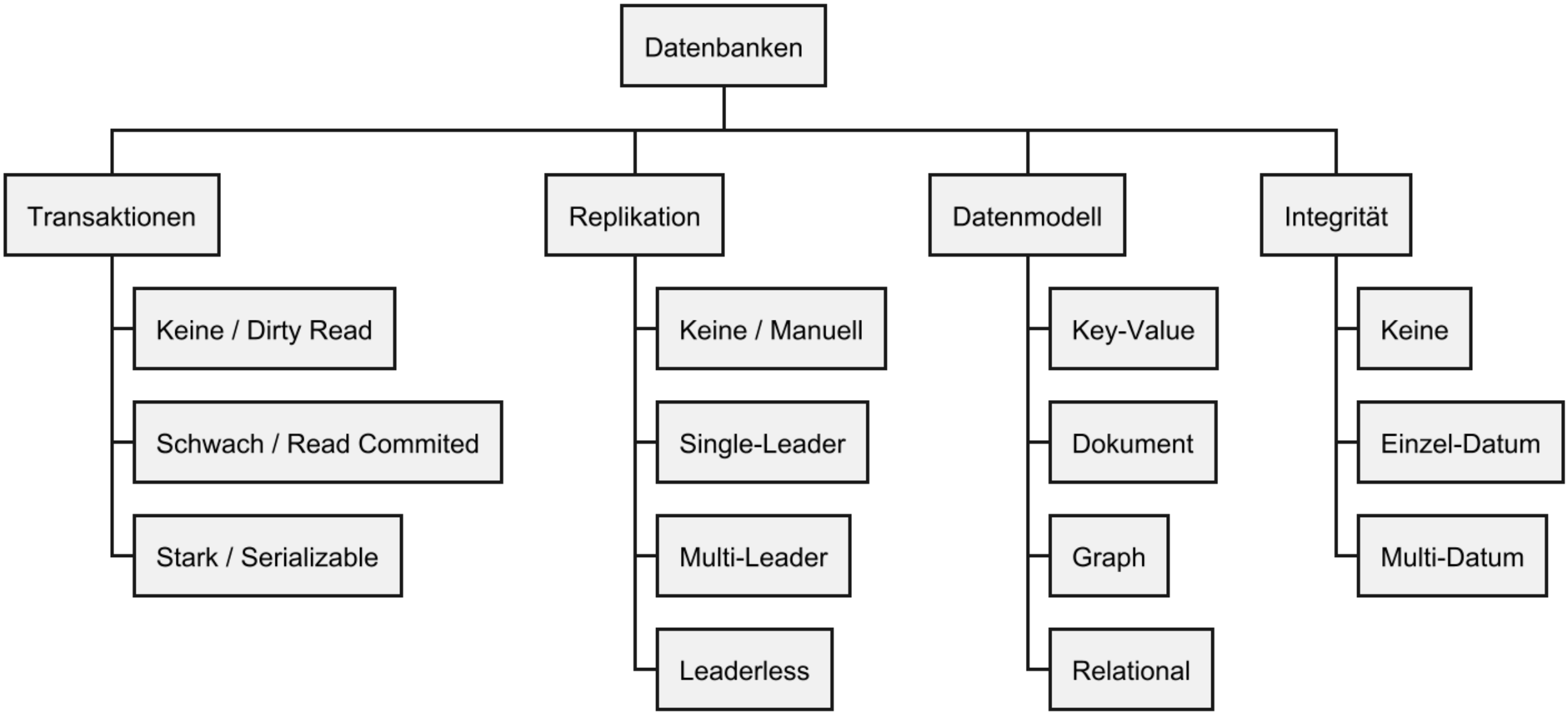
Werkzeuge

MS SQL

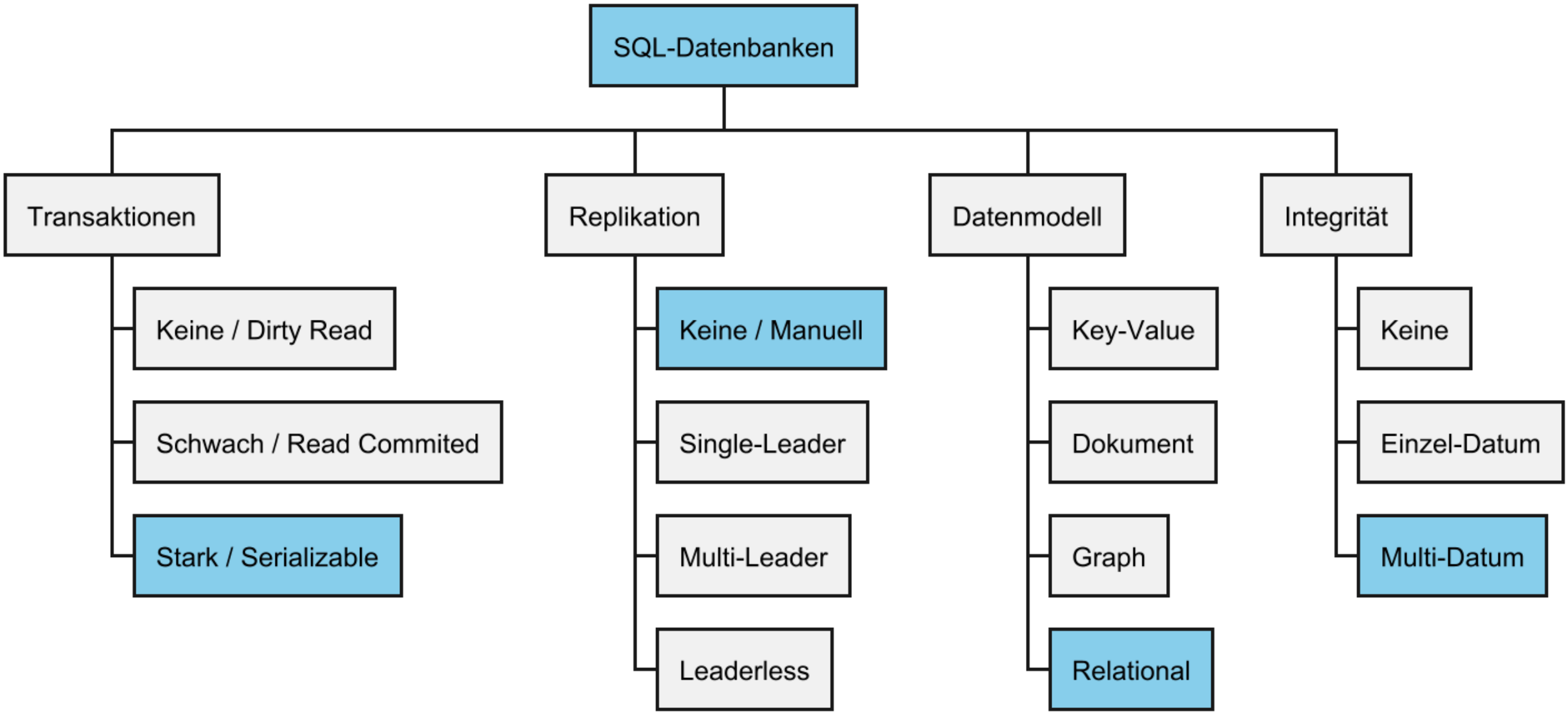
PostgreSQL

Oracle

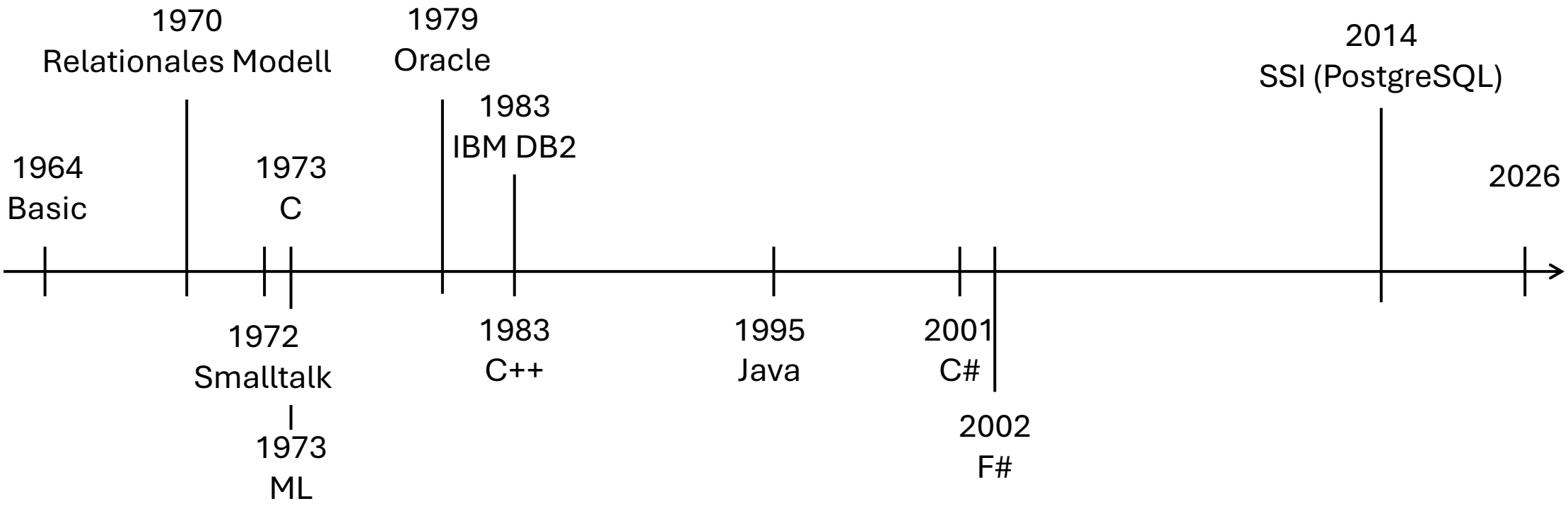
Das "Ich bau mir eine Datenbank"-Spiel
Wähle 1x aus jedem Ast



Das "Ich bau mir eine Datenbank"-Spiel
Wähle 1x aus jedem Ast



Geschichte


1964
Basic
1970
Relationales Modell
1973
C
1972
Smalltalk
1973
ML
1979
Oracle
1983
IBM DB2
1983
C++
1995
Java
2001
C#
2002
F#
2014
SSI (PostgreSQL)
2026

Lösung für was?

> *Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.*
> *Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.*

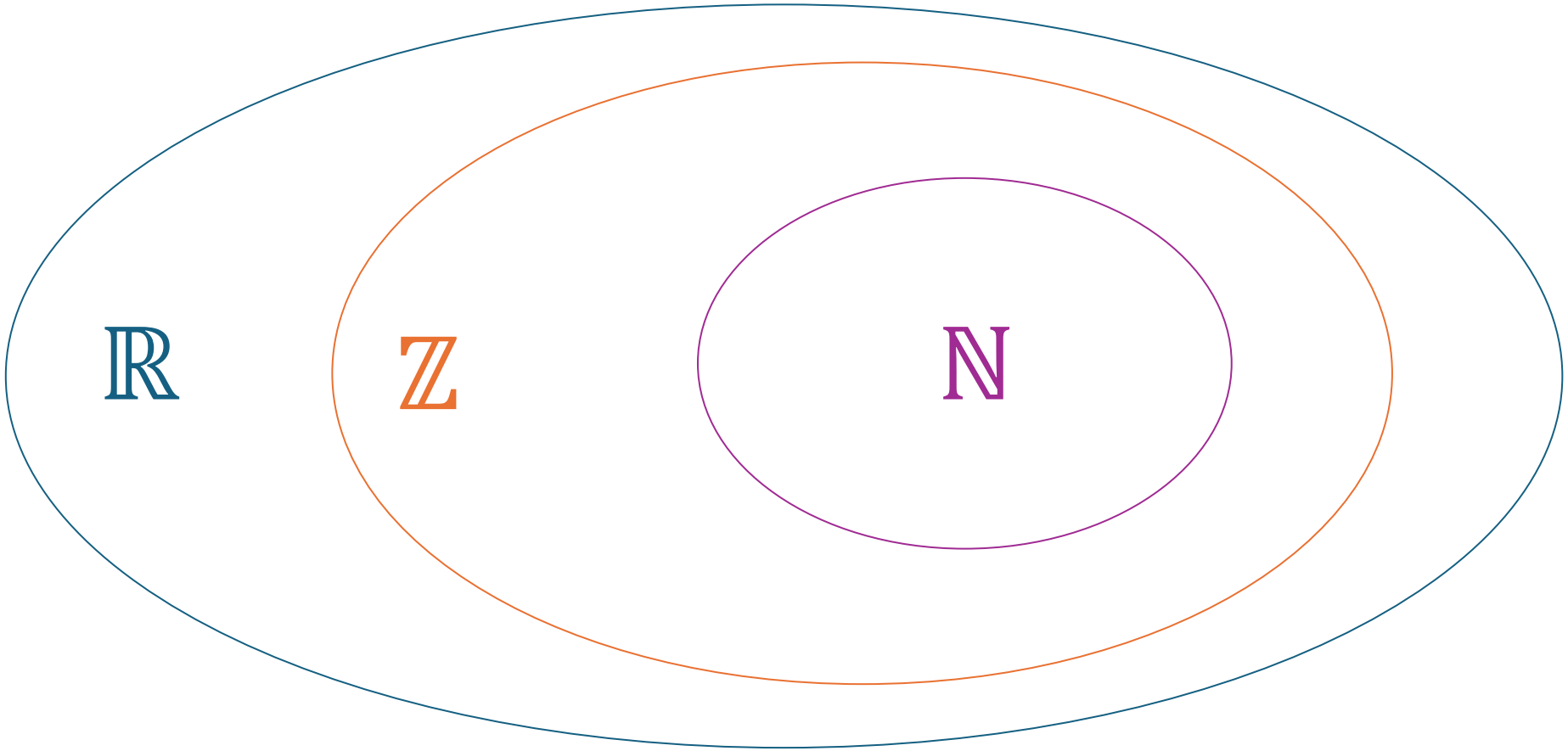
Abstract von “A relational model of data for large shared data banks”
E.F. Codd, 1. Juli 1970

Warum ist Struktur so wichtig?

- `052knai0111D160129Erec727711ap208ne55S001061016`
- `100SpenserE210619802615O160PiankaD1110117727250`
- `100 Spenser E21 06-19-80 26150 160 Pianka D11 10-11-77 27250`
- `100 Spenser E21 06-19-80 26150`
  `160 Pianka  D11 10-11-77 27250`

- “BUSINESS MODELING FOR DATABASE DESIGN: Formalizing the Informal (PRACTICAL DATABASE FOUNDATIONS Book 1)”, Fabian Pascal, 2015

Mengentheorie



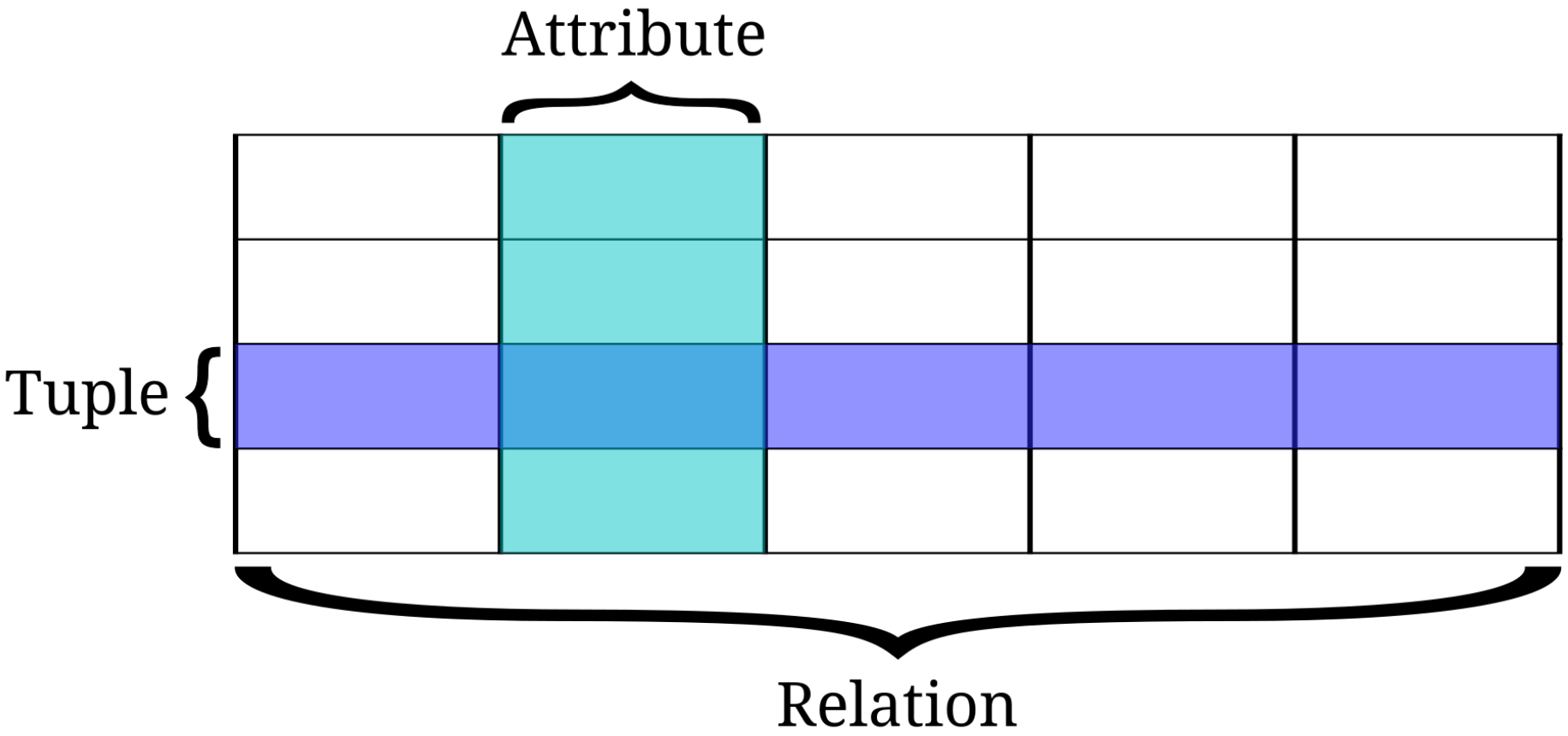
Logik

- ```
  100 Spenser E21 06-19-80 26150
  160 Pianka  D11 10-11-77 27250
  ```

- Ein Mitarbeiter wird identifiziert durch Nummer (EMP#), hat den Namen (ENAME), arbeitet in der Abteilung mit der Nummer (DEPT#), wurde eingestellt zum Datum (HIREDATE) und verdient jährlich (SALARY) €.

- ```
  EMP# ENAME   DEPT# HIREDATE SALARY
  100  Spenser E21   06-19-80 26150
  160  Pianka  D11   10-11-77 27250
  ```

Relationales Modell



https://commons.wikimedia.org/wiki/File:Relational_database_terms.svg

Normalformen: insert anomaly

Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201
424	Dr. Newsome	29-Mar-2007	?

https://commons.wikimedia.org/wiki/File:Insertion_anomaly.svg

Normalformen: Update anomaly

Employees' Skills

Employee ID	Employee Address	Skill
426	87 Sycamore Grove	Typing
426	87 Sycamore Grove	Shorthand
519	94 Chestnut Street	Public Speaking
519	96 Walnut Avenue	Carpentry

https://commons.wikimedia.org/wiki/File:Update_anomaly.svg

Normalformen: delete anomaly

Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201

DELETE

https://commons.wikimedia.org/wiki/File:Deletion_anomaly.svg

Normalformen

- 0 Normalform:
  - nicht normalisiert, Redundanzen
- 3 Normalform:
  - fast keine Redundanzen mehr in einer “Tabelle” (Funktionale Abhängigkeiten)
- 3,5 / Boyce-Codd Normalform:
  - Keine Redundanzen mehr in einer “Tabelle”
  - Normalerweise ausreichend
- 5 Normalform:
  - Keine Redundanzen mehr zwischen mehreren “Tabellen” / (Join Abhängigkeiten). Weitere Zerlegung in kleinere “Tabellen” führt zu Datenverlust.

Warum SQL so scheiön ist

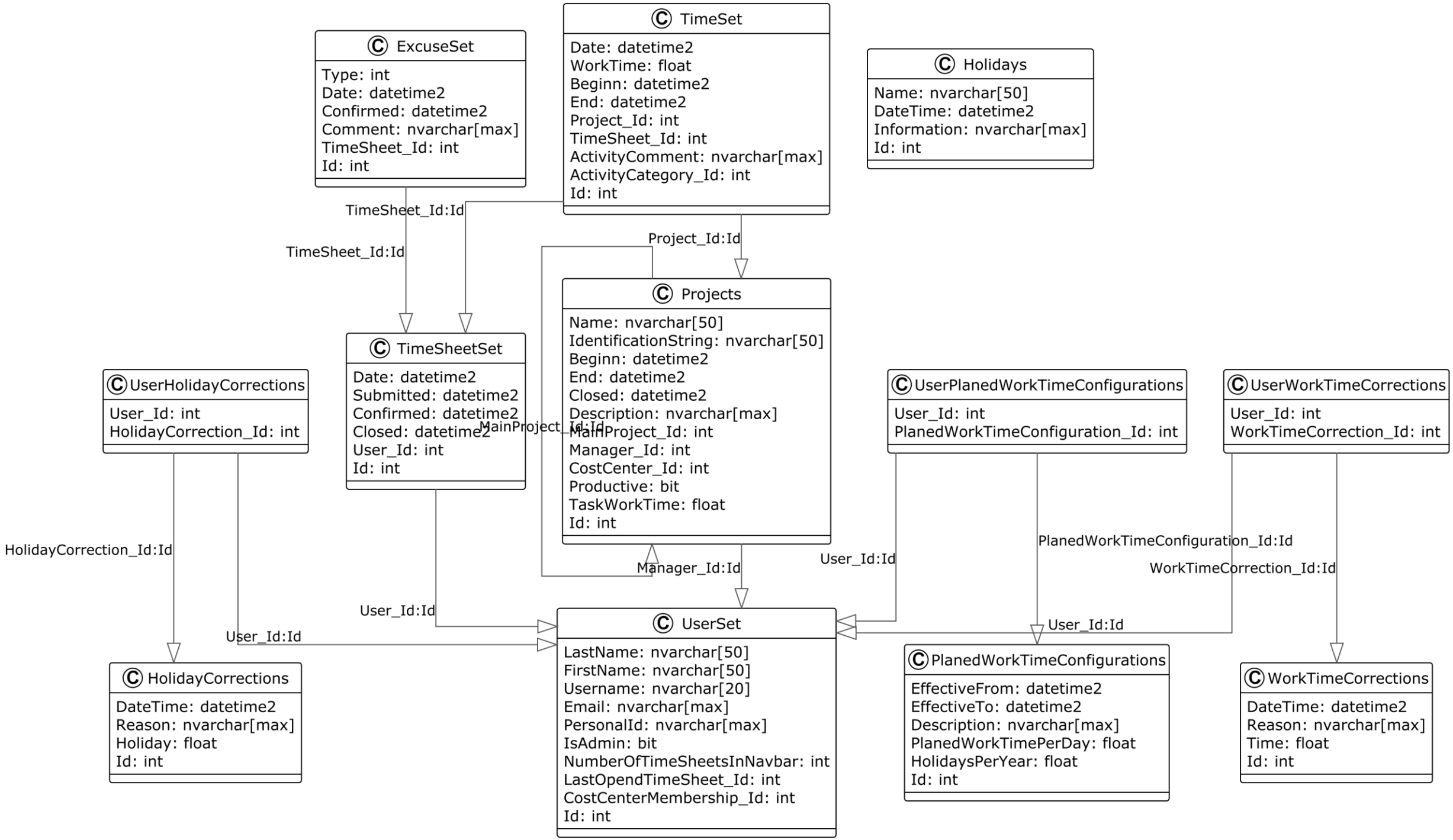
- Katastrophale Syntax: `select … from … where`
- Keine Trennung von logischem und physischem Schema
- WAHR, FALSCH, NULL → Dreiwertlogik

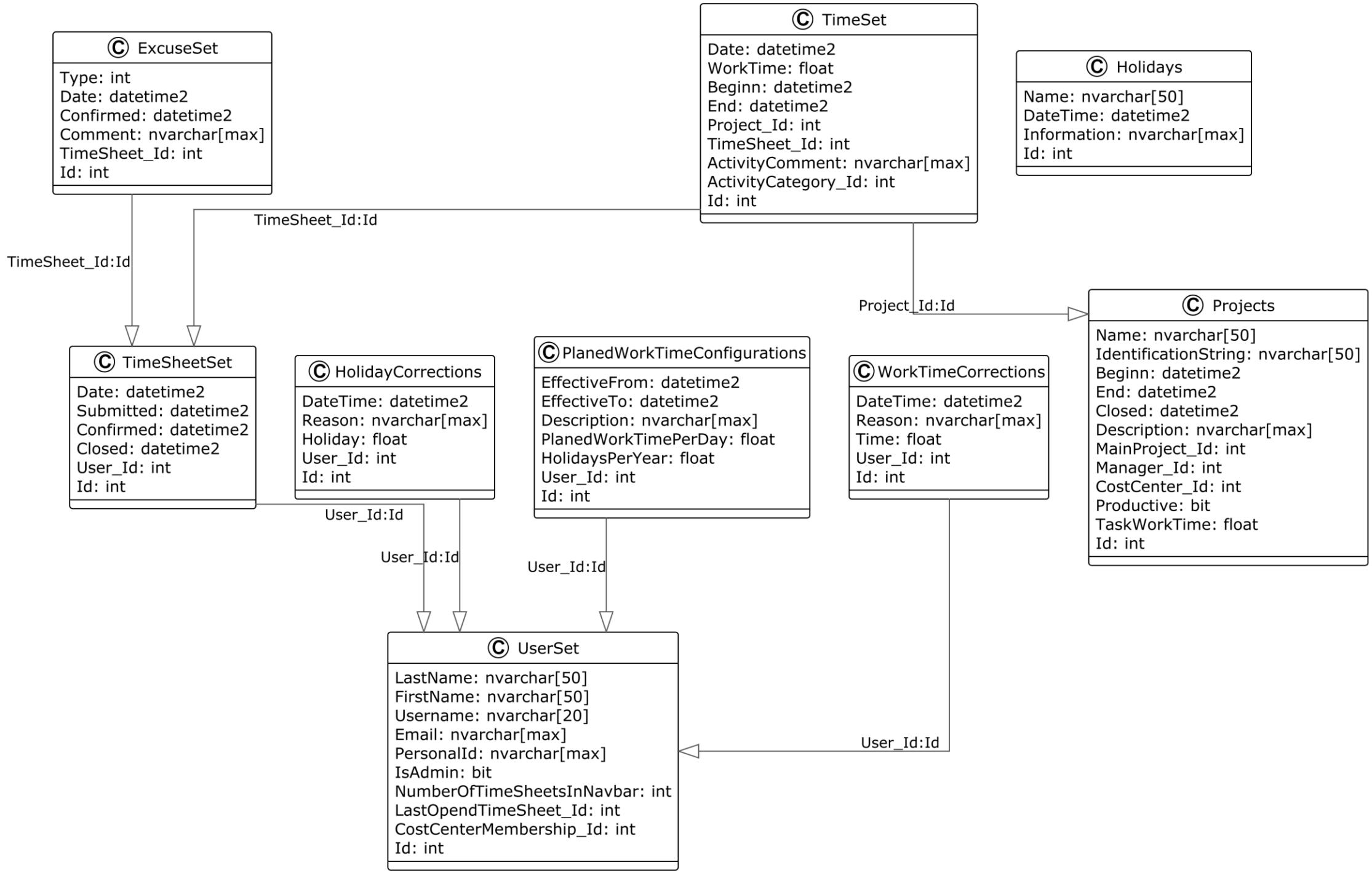
```
SELECT * FROM MyTable WHERE MyColumn != NULL (0 Results)
SELECT * FROM MyTable WHERE MyColumn <> NULL (0 Results)
SELECT * FROM MyTable WHERE MyColumn IS NOT NULL (568 Results)
```

- SQL erlaubt doppelte Einträge
- SQL erlaubt keine Constraints über mehrere Einträge oder Spalten
  - CREATE ASSERTION …

https://stackoverflow.com/questions/5658457/not-equal-operator-on-null

Beispiel: Sollarbeitszeitberechnung


ExcuseSet
Type: int
Date: datetime2
Confirmed: datetime2
Comment: nvarchar[max]
TimeSheet_Id: int
Id: int
TimeSet
Date: datetime2
WorkTime: float
Beginn: datetime2
End: datetime2
Project_Id: int
TimeSheet_Id: int
ActivityComment: nvarchar[max]
ActivityCategory_Id: int
Id: int
Holidays
Name: nvarchar[50]
DateTime: datetime2
Information: nvarchar[max]
Id: int
TimeSheet_Id:Id
TimeSheet_Id:Id
Project_Id:Id
Projects
Name: nvarchar[50]
IdentificationString: nvarchar[50]
Beginn: datetime2
End: datetime2
Closed: datetime2
Description: nvarchar[max]
MainProject_Id: int
Manager_Id: int
CostCenter_Id: int
Productive: bit
TaskWorkTime: float
Id: int
TimeSheetSet
Date: datetime2
Submitted: datetime2
Confirmed: datetime2
Closed: datetime2
User_Id: int
Id: int
MainProject_Id:Id
UserHolidayCorrections
User_Id: int
HolidayCorrection_Id: int
UserPlanedWorkTimeConfigurations
User_Id: int
PlanedWorkTimeConfiguration_Id: int
UserWorkTimeCorrections
User_Id: int
WorkTimeCorrection_Id: int
HolidayCorrection_Id:Id
Manager_Id:Id
User_Id:Id
PlanedWorkTimeConfiguration_Id:Id
WorkTimeCorrection_Id:Id
User_Id:Id
User_Id:Id
User_Id:Id
UserSet
LastName: nvarchar[50]
FirstName: nvarchar[50]
Username: nvarchar[20]
Email: nvarchar[max]
PersonalId: nvarchar[max]
IsAdmin: bit
NumberOfTimeSheetsInNavbar: int
LastOpendTimeSheet_Id: int
CostCenterMembership_Id: int
Id: int
HolidayCorrections
DateTime: datetime2
Reason: nvarchar[max]
Holiday: float
Id: int
PlanedWorkTimeConfigurations
EffectiveFrom: datetime2
EffectiveTo: datetime2
Description: nvarchar[max]
PlanedWorkTimePerDay: float
HolidaysPerYear: float
Id: int
WorkTimeCorrections
DateTime: datetime2
Reason: nvarchar[max]
Time: float
Id: int


ExcuseSet
Type: int
Date: datetime2
Confirmed: datetime2
Comment: nvarchar[max]
TimeSheet_Id: int
Id: int
TimeSet
Date: datetime2
WorkTime: float
Beginn: datetime2
End: datetime2
Project_Id: int
TimeSheet_Id: int
ActivityComment: nvarchar[max]
ActivityCategory_Id: int
Id: int
Holidays
Name: nvarchar[50]
DateTime: datetime2
Information: nvarchar[max]
Id: int
TimeSheet_Id:Id
TimeSheet_Id:Id
Project_Id:Id
TimeSheetSet
Date: datetime2
Submitted: datetime2
Confirmed: datetime2
Closed: datetime2
User_Id: int
Id: int
HolidayCorrections
DateTime: datetime2
Reason: nvarchar[max]
Holiday: float
User_Id: int
Id: int
PlanedWorkTimeConfigurations
EffectiveFrom: datetime2
EffectiveTo: datetime2
Description: nvarchar[max]
PlanedWorkTimePerDay: float
HolidaysPerYear: float
User_Id: int
Id: int
WorkTimeCorrections
DateTime: datetime2
Reason: nvarchar[max]
Time: float
User_Id: int
Id: int
Projects
Name: nvarchar[50]
IdentificationString: nvarchar[50]
Beginn: datetime2
End: datetime2
Closed: datetime2
Description: nvarchar[max]
MainProject_Id: int
Manager_Id: int
CostCenter_Id: int
Productive: bit
TaskWorkTime: float
Id: int
User_Id:Id
User_Id:Id
User_Id:Id
User_Id:Id
UserSet
LastName: nvarchar[50]
FirstName: nvarchar[50]
Username: nvarchar[20]
Email: nvarchar[max]
PersonalId: nvarchar[max]
IsAdmin: bit
NumberOfTimeSheetsInNavbar: int
LastOpendTimeSheet_Id: int
CostCenterMembership_Id: int
Id: int

Literatur

- Data-Oriented Design, Richard Fabian, 2018
- Designing Data-Intensive Applications, Martin Kleppmann, 2017
- An Introduction to Relational Database Theory, Darwin Hugh, 2014
  - SQL: A Comparative Survey, Darwin Hugh, 2017
- Applied Mathematics for Database Professionals, Lex de Haan, Toon Koppelaars, 2007

Werkzeuge