GFS HAFAS-RS

Daten abfragen bei der Deutschen Bahn

Adrian Struwe, 11.01.2023

Inhaltsverzeichnis

Einführung

Inhalt

- Zielsetzung des Projekts
- HaCon Fahrplan-Auskunfts-System
- Programmiersprache Rust
- Datenbanksystem Postgresql

Einführung

Zielsetzung des Projekts

- C36C3 - David Kriesel
- Fahrplandaten der Deutschen Bahn extrahieren und graphisch darstellen
- Verspätungsdaten sammeln
- Schlüsse ziehen

Einführung

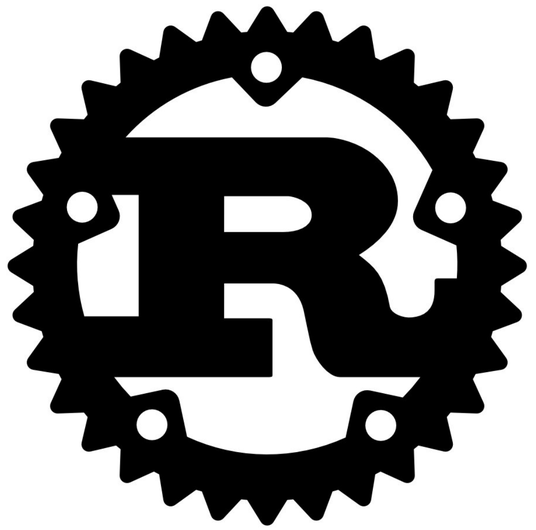
HAFAS

- **Ha**Con **F**ahrplan-**A**uskunfts-**S**ystem
- HaCon - Hannover Consulting Tochterfirma von Siemens
- Programmierschnitsttelle (API)
- Verwendet von DB, SBB, ÖBB, SNCF (Niederlande), PKP (Polen)

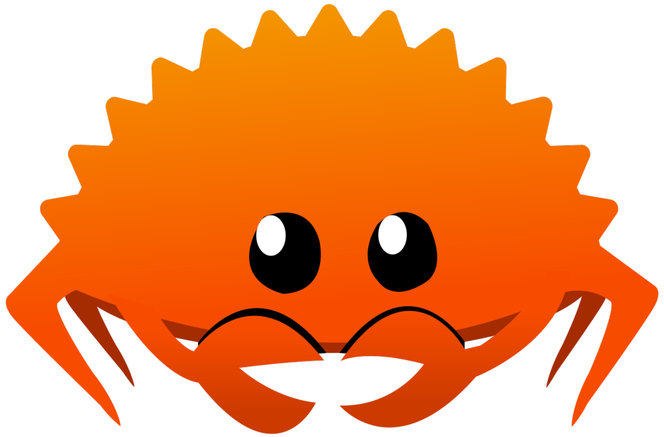
Einführung

Programmiersprache Rust

- Systemunabhängig
- Speichersicher ohne automatische Speicherbereinigung
- Schnell
- Sinnvolle Fehlermeldungen



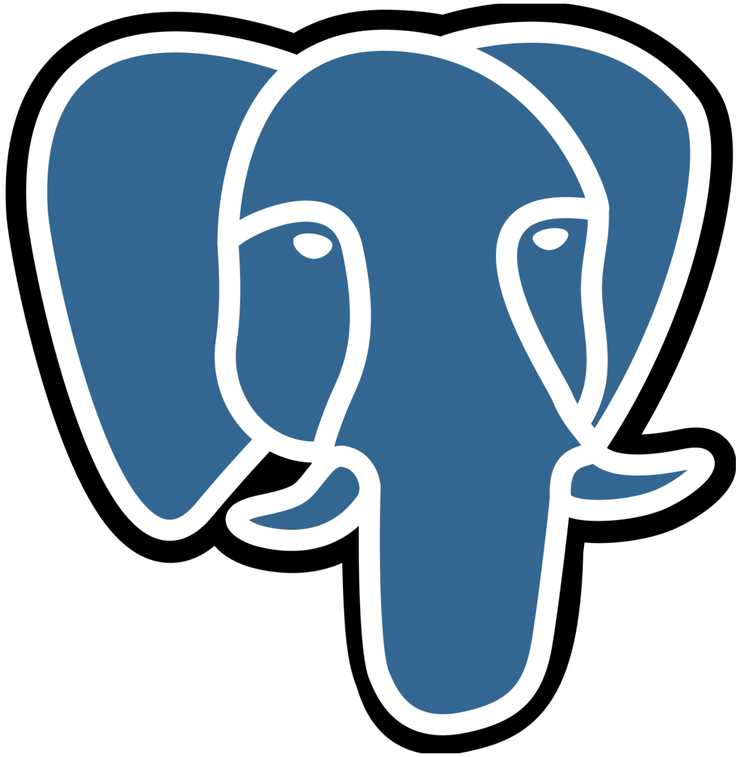
The Rust Programming Language



Einführung

Datenbanksystem Postgresql

- Quelloffen (Open Source)
- Objektrelationale Datenbank
- Zuverlässig
- SQL Standard konform



```
SELECT name FROM customers
WHERE country = ‘Germany‘;
```

OOP in Rust

Inhalt

- Klassendefinition
  - Attribute
  - Methoden (UP‘s)
- Vererbung
- (Bonus) Generische Typen

```rust
use sqlx::connection::PgConnection;

pub struct Client {
    /// A reqwest client used to make web requests.
    reqwest: reqwest::Client,
    /// A database connection used to insert data
    conn: PgConnection,
}

impl Client {
    pub async fn make_request(&mut self, url: &str) {
        self.reqwest
            .post(url)
            .send()
            .await
            .unwrap();
    }
}

impl From<PgConnection> for Client {
    fn from(value: PgConnection) -> Client {
        Client {
            reqwest: reqwest::Client::default(),
            conn: value,
        }
    }
}
```

OOP in Rust

Klassendefinition - Attribute

Java

```java
import sqlx.connection.PgConnection;
import reqwest.Client;

public class Client {
    private Client reqwest;
    private PgConnection conn;
}
```

Rust

```rust
use sqlx::connection::PgConnection;

pub struct Client {
    reqwest: reqwest::Client,
    conn: PgConnection,
}

```

OOP in Rust

Klassendefinition - Methoden (Unterprogramme)

Java

```java
import sqlx.connection.PgConnection;
import reqwest.Client;

public class Client {
    private Client reqwest;
    private PgConnection conn;

    public String make_request(String url) {
        String response = this.reqwest
            .post(url)
            .send()
            .text;

        return text;
    }
}
```

Rust

```rust
use sqlx::connection::PgConnection;

pub struct Client {
    reqwest: reqwest::Client,
    conn: PgConnection,
}

impl Client {
    pub fn make_request(&self, url: String) -> String {
        let response: String = self.reqwest
            .post(&url)
            .send()
            .text;

        return text;
    }
}
```

OOP in Rust

Klassendefinition - Vererbung

Java

```java
import sqlx.connection.PgConnection;
import reqwest.Client;

public class Client implements From<PgConnection> {
    private Client reqwest;
    private PgConnection conn;

    public Client (PgConnection conn) {
        this.reqwest = new reqwest.Client();
        this.conn = conn;
    };

    public Client from(PgConnection value) {
        return new Client(value);
    }
}
```

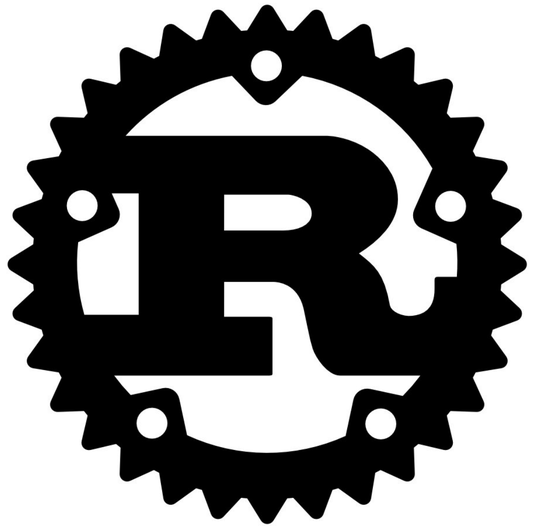
Rust

```rust
use sqlx::connection::PgConnection;

pub struct Client {
    reqwest: reqwest::Client,
    conn: PgConnection,
}

impl From<PgConnection> for Client {
    fn from(value: PgConnection) -> Client {
        Client {
            reqwest: reqwest::Client::default(),
            conn: value,
        }
    }
}
```

OOP in Rust

Unterschiede zu Java

- Vererbung
  - Keine klassische Vererbung
  - Vererbung durch Traits, bzw. Java Interfaces
- Funktionsdefinition außerhalb der Klasse

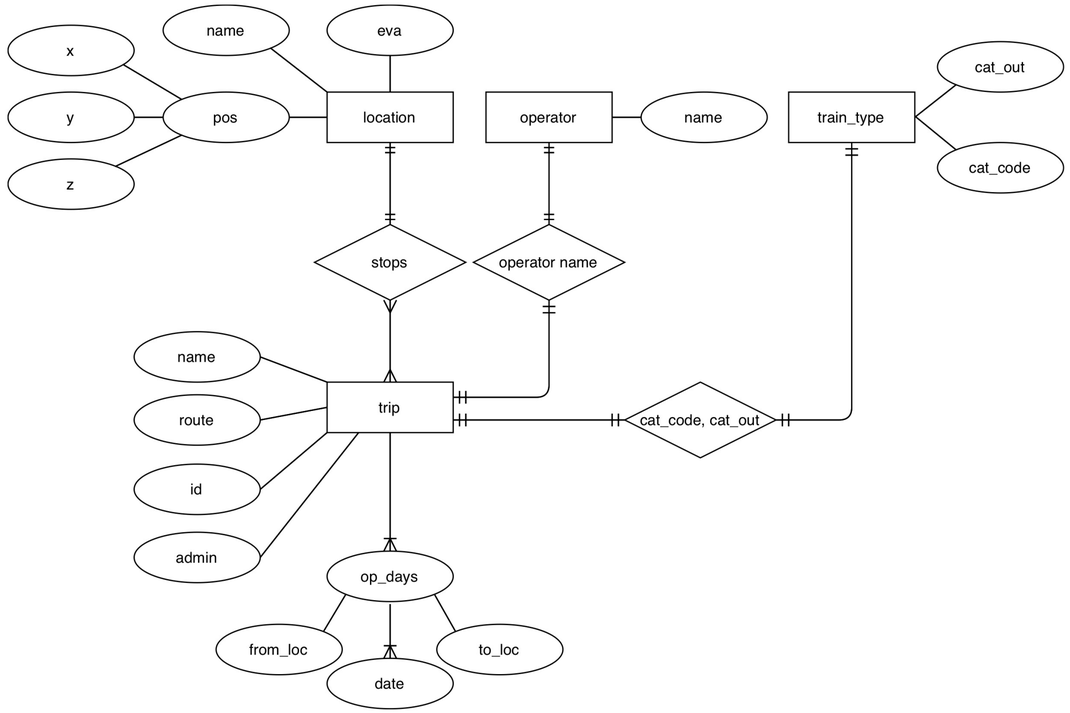


The Rust Programming Language

Chen Notation

Inhalt

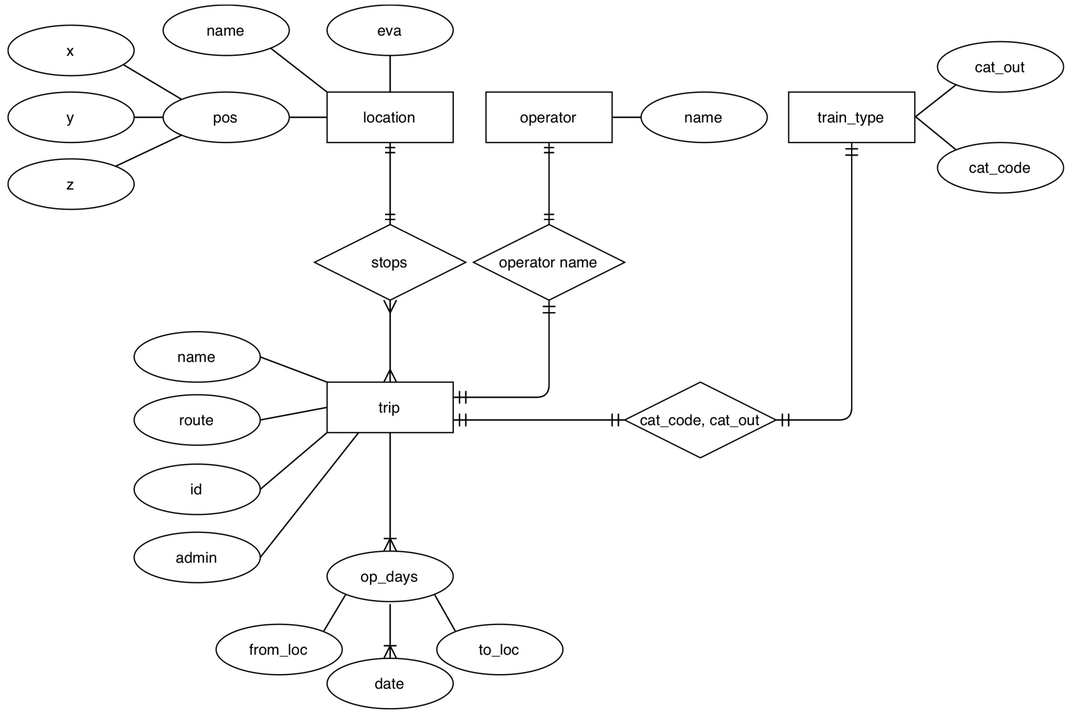
- Was ist das?
- Typen
- Attribute
- Beziehungen



Chen Notation

Was ist das?

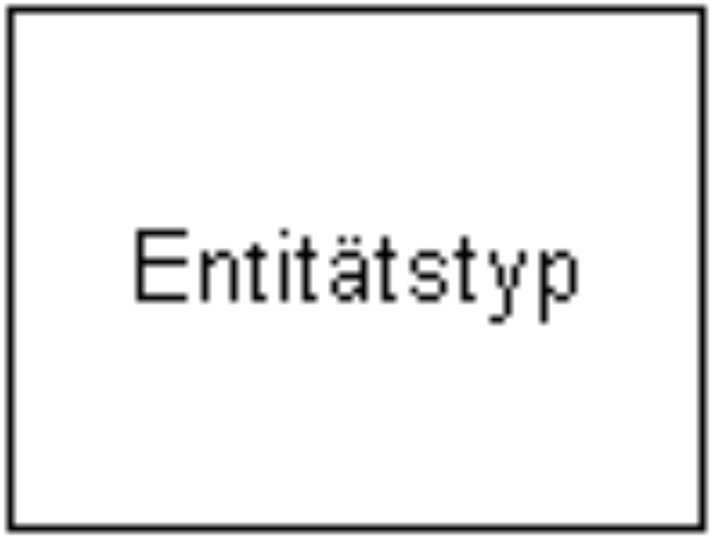
- Diagramm
- Beschreibt Objektmodelle
- Benannt nach Peter Chen
- Teil des Lehrplans Informatik BW



Chen Notation

Typen

- Wie Klassen
- Als Rechteck dargestellt



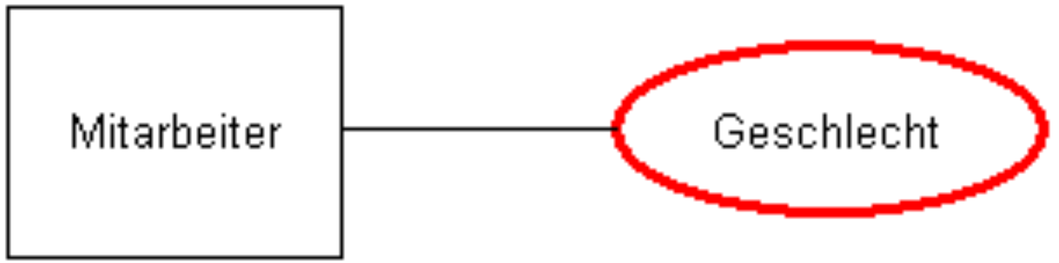
Attribute

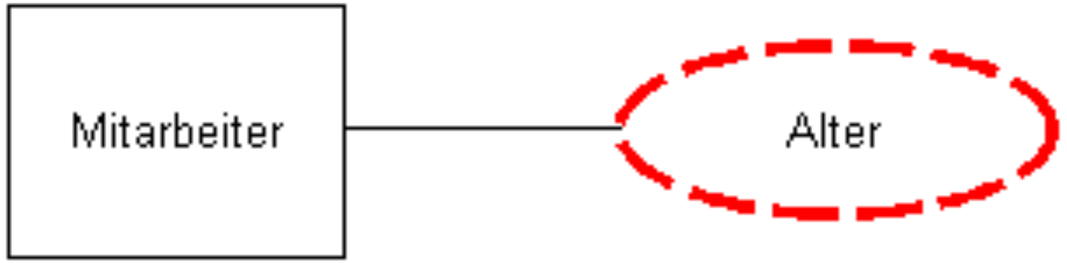
- Attribut einer Klasse
- Als Oval dargestellt

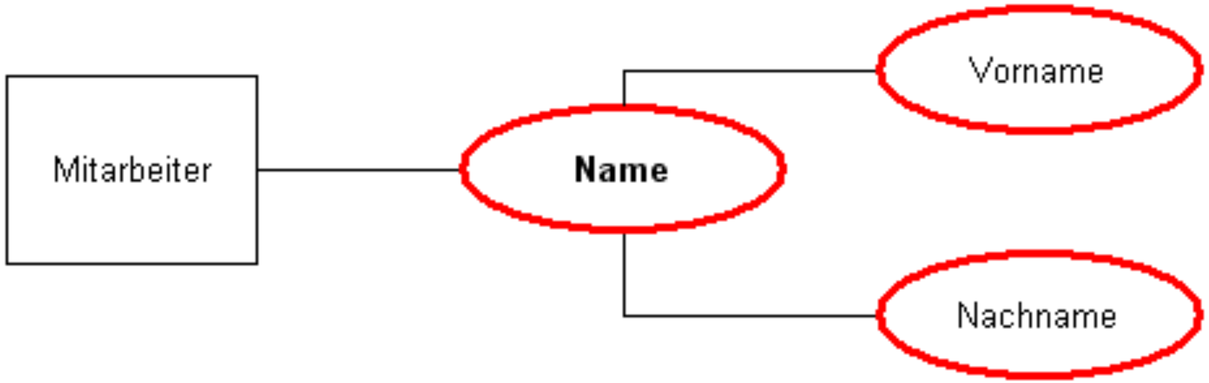
Chen Notation

Klassen und Attribute

- Klassen bestehen aus Attributen
- Abgeleitete Attribute werden gestrichelt dargestellt
- Attribute können aus mehreren Attributen zusammengesetzt sein



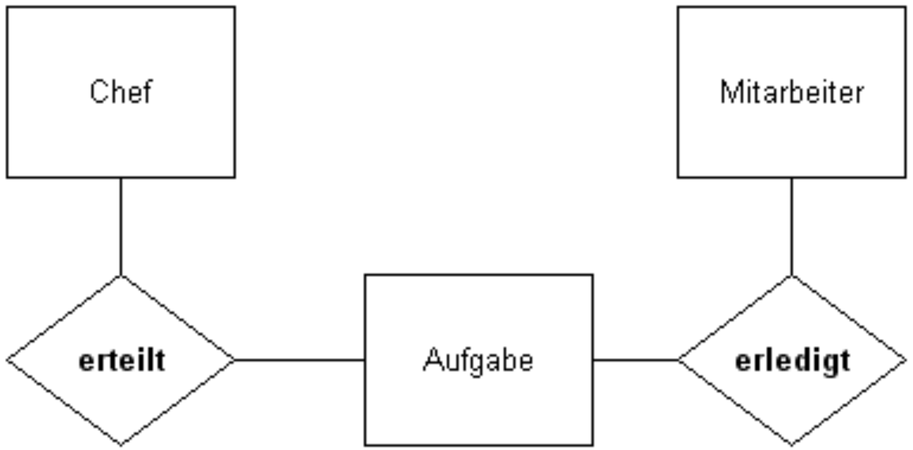


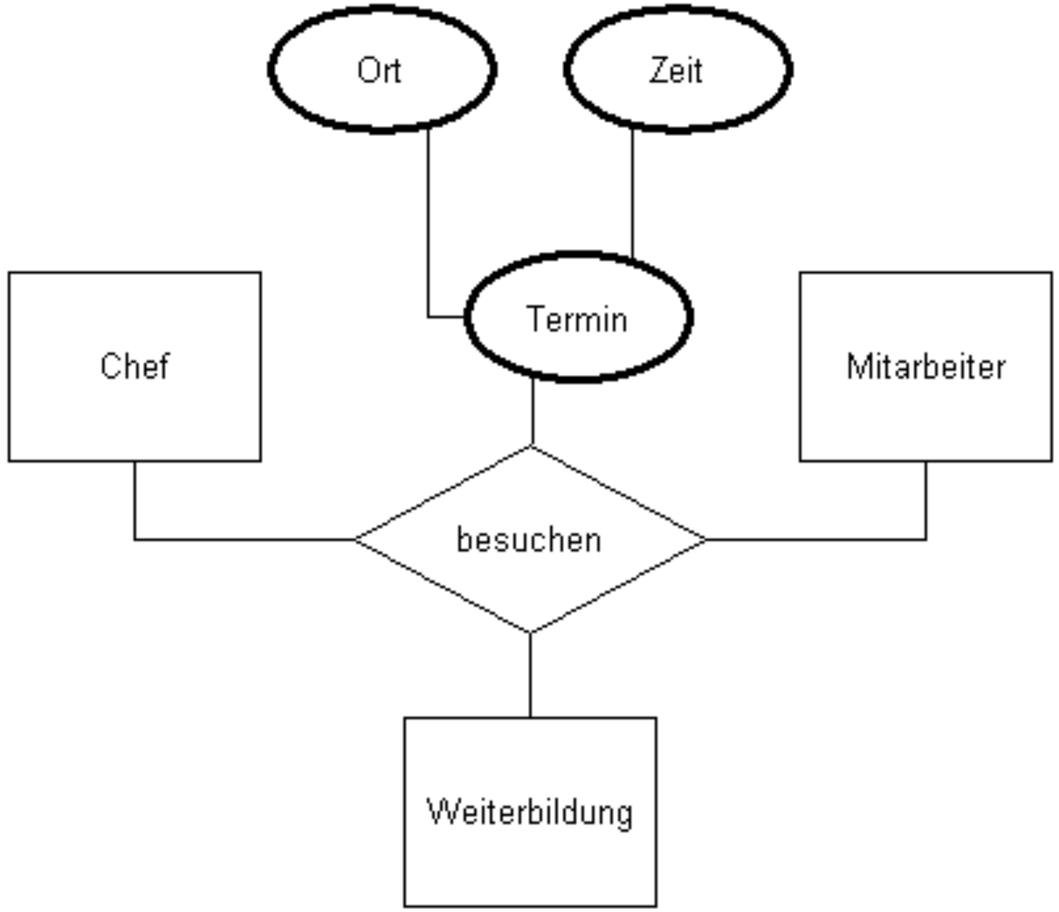


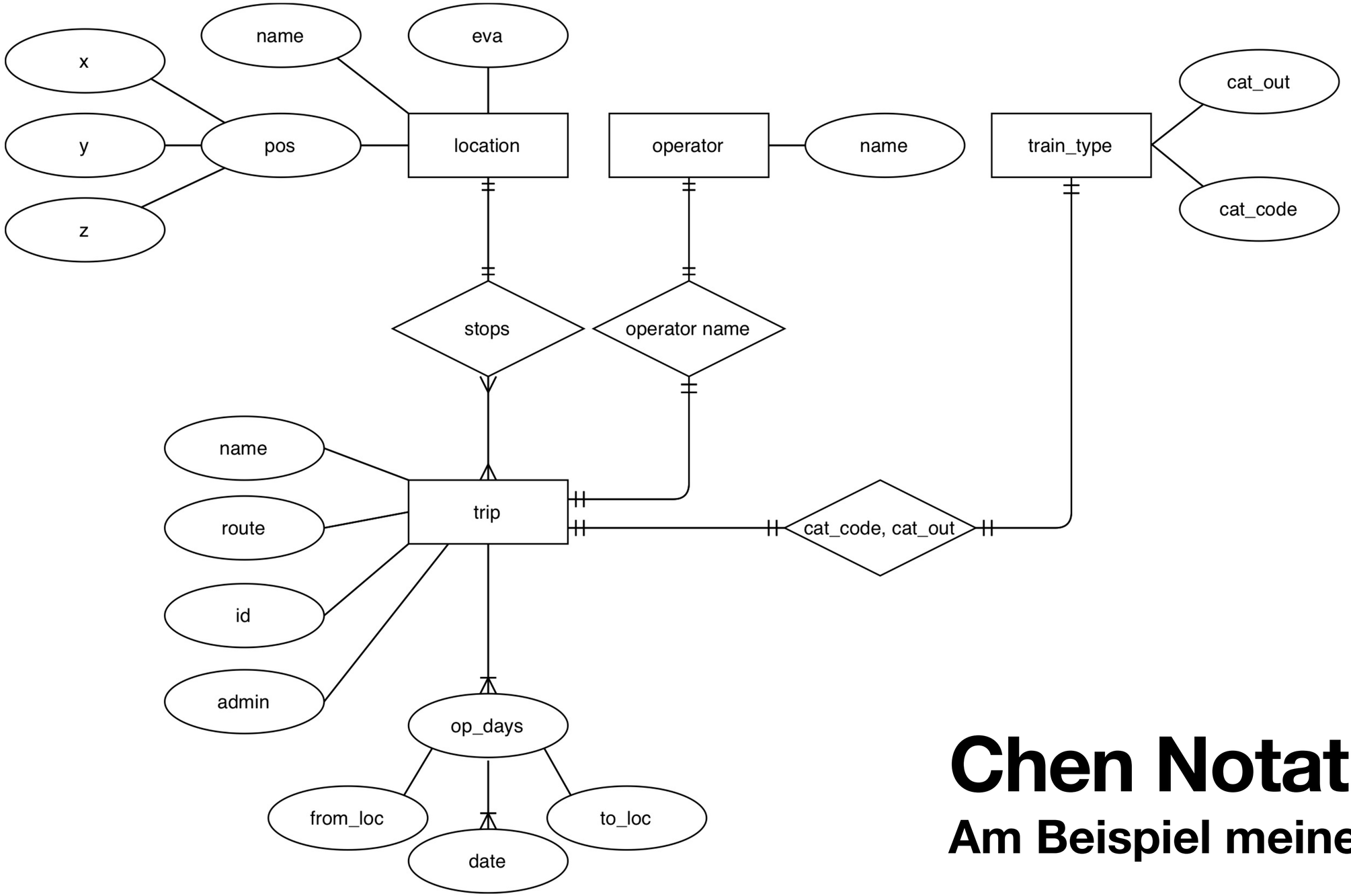
Chen Notation

Beziehungen

- Beschreiben Verknüpfungen zwischen verschiedenen Typen
- Können Attribute haben







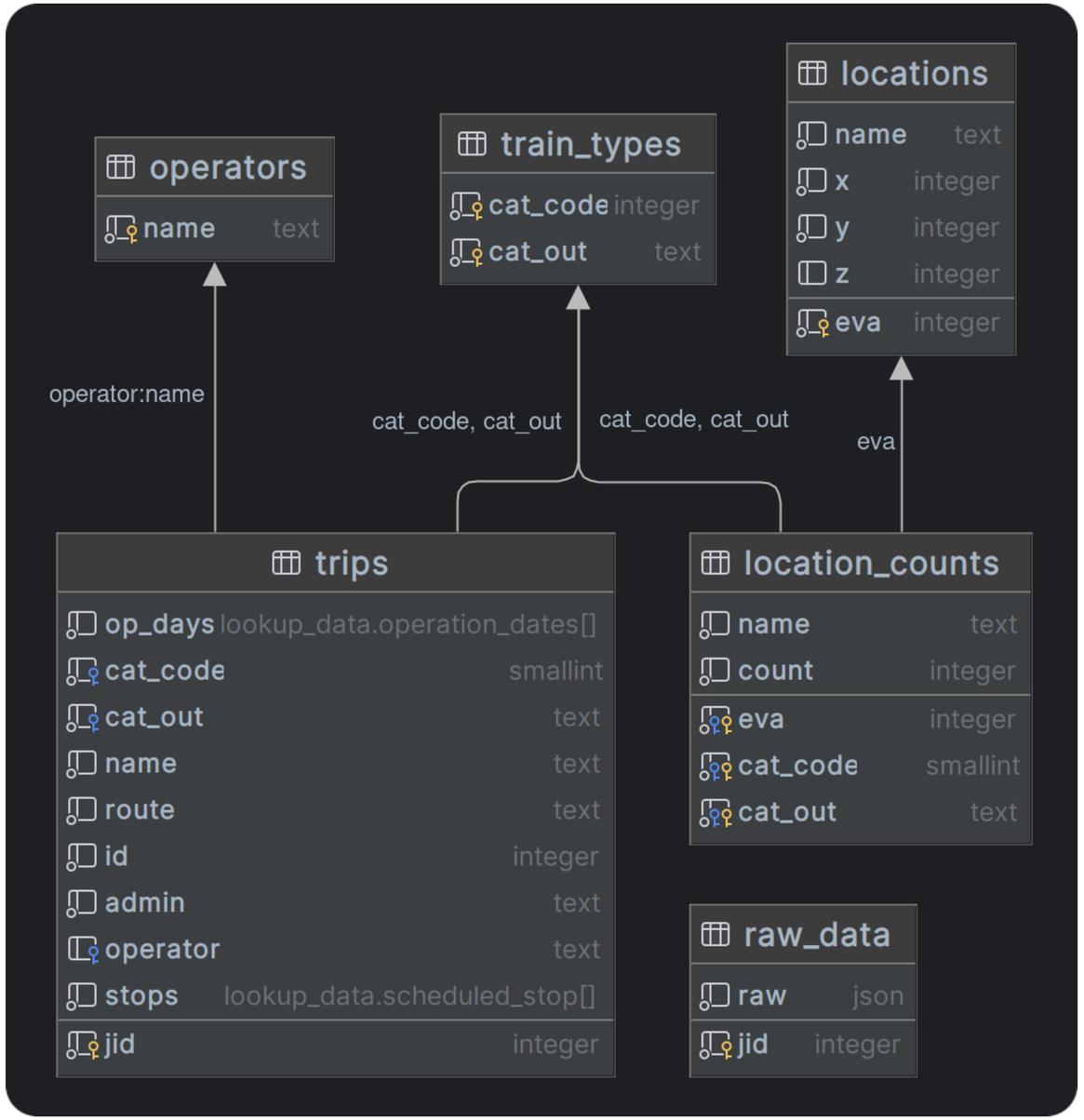
Chen Notation

Am Beispiel meiner Datenbank

Chen Notation

Gutes Datenbankdesign

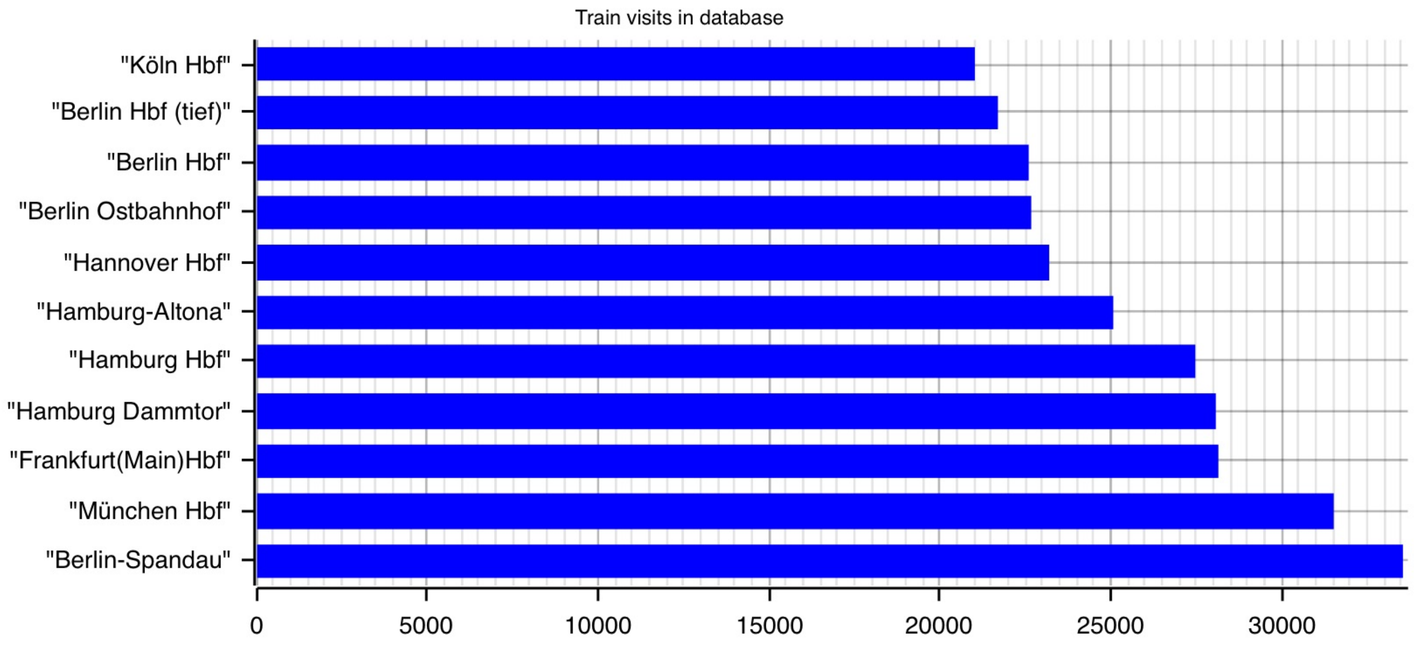
- Wenig unnütze Daten
- Sinnvolle Verteilung auf mehrere Tabellen
- Möglichst keine nullbaren Spalten
- Einfach erweiterbar
- Konzentration auf das Ziel



Ergebnispräsentation

Beispiel Datenauswertung

- Hier: Häufigkeit ICE-Durchfahrten pro Bahnhof; Vergleich der am häufigsten angefahrenen Bahnhöfe
- Gesamter Fahrplan kann ausgewertet werden
- Geplante Erweiterung auf Verspätungsdaten



Fragen?

Bildquellen in Moderationsnotizen.
Bilder ohne Quellen sind selbst erstellt.