Hochschule Karlsruhe
University of
Applied Sciences

Hochschule Karlsruhe – Data Engineering
WS 2021/2022
DSCB330 – Vorlesung 1 - Einführung

Foto: Quirin Leppert

Kontakt

Kontaktdaten

Thomas Bierweiler
thomas.bierweiler@h-ka.de
thomas.bierweiler@siemens.com

Fragen

- Über ILIAS
- per E-Mail

Folien

Die Folien werden Ihnen zur Verfügung gestellt.

Werdegang

- Studium der Physik an der Universität Karlsruhe (TH) (heute KIT) und an der University of Massachusetts (USA)
- Entwicklungsingenieur bei der inspectomation GmbH (2002-2006)
- Projektleiter Technology&Innovation, Automatisierungstechnik für prozesstechnische Anlagen, Siemens AG (seit 2007)

Notengebung

- Schriftliche Prüfung im Februar 2022
- Bis zu 10 % Bonus bei Abgabe aller Übungsaufgaben mit sinnvollen Lösungen oder
- durch Vorbereitung und Vortragen eines Themas

Themenübersicht

Data Integration

- Data Formats (csv, XML, json)
- Extract, Transform, Load
- Object Relation Mapper (ORM)
- Staging

Data Processing

- Relationale Datenbanken
- nicht-relationale Datenbanken
- Resource Description Framework (RDF)
- Ontologien
- Data Warehouse

Data Modelling

- Serialisierung
- OPC UA
- MQTT
- Pub/Sub
- Data pipelines
  - Apache Airflow
  - gRPC

Web-Service Architektur

- Front-End
- Backend for Frontend (BFF)
- Micro Services
- Docker Container

Security

- Security ist wichtig in allen Phasen der Softwareentwicklung und Datenbereitstellung.

Übung

- Erstellung eines Daten-Modells einer prozesstechnischen Anlage
- Statische Daten
- Dynamische Daten
- Auswertung der Daten

Software & Tools

Relationale Datenbank

- Microsoft SQL Server
  Download: https://www.microsoft.com/de-de/sql-server/sql-server-downloads
- SQL Server Management Studio (SSMS)
  Download: https://docs.microsoft.com/de-de/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

Nicht-relationale Datenbank

- mongoDB
  Download:
  https://www.mongodb.com/try/download/community?tck=docs_server

Sourcecode-Verwaltung mit git

- git bash
  Download: https://git-scm.com/download

Programmiersprache

- Python (aktuelle Version: 3.9.7)
  Download: https://www.python.org/downloads/
- Datenbanktreiber pyodbc für MS SQL
  python –m pip install pyodbc

Editor für python

- z.B. Visual Studio Code
  Download: https://code.visualstudio.com/download

Aufgaben eines Data Engineers

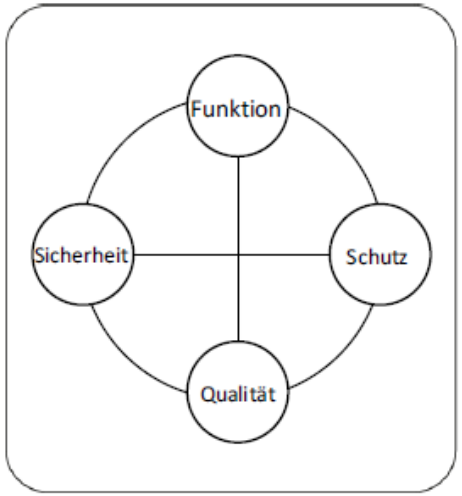
- Instandhaltung der Datenbank-Infrastruktur
- Extraktion, Zusammenführen und Bereitstellung von Daten in einer Pipeline (ETL)

Big Data

- Volume of data – Scale of Data
  Die Datenmenge hat in den letzten Jahren rapide zugenommen
- Variety – Different Forms of Data
  Daten aus verschiedenen Datenbanken in unterschiedlichen Formaten müssen vereinheitlicht werden
- Velocity – Analysis of Streaming Data
  Auswertung von Daten nahezu in Echtzeit
- Veracity – Uncertainty of Data
  Unsicherheit der in den Datenbanken abgespeicherten Informationen

Smart Data

- intelligente Daten
- haben Informationen über sich selbst
- kennen ihre Bedeutung und Deutung (Semantik)



Merkmale von Smart Data

Quellen: Data Engineering with python, Paul Crickard, Packt, Oktober 2020; Data Engineering 4.0, Kompositionale Informationsmodelle für industrielle Anwendungen, Herbert Weber, Springer, 2021

Schreiben und Lesen von csv-Dateien (1)

Speichern und Lesen einer Materialdatenbank

- Speichern mit Paket csv
- Lesen mit Paket csv
- Lesen mit pandas -> Sonderzeichen (Umlaute, °) werden nicht eingelesen.

Lesen von csv-Dateien (1)

Speichern und Lesen einer Materialdatenbank

- Skripts in 01-WriteReadCSV
- Speichern mit Paket csv
- Lesen mit Paket csv
- Lesen mit pandas -> Sonderzeichen (Umlaute, °) werden nicht eingelesen.

Vor dem Einlesen der Dateien achten auf

- Dezimaltrennzeichen (3,14159 oder 3.1459)
- Tausender-Trennzeichen (1000,5 oder 1.000,5 oder 1,000.5)
- Trennzeichen zwischen Spalten (Leerzeichen, Komma oder Semikolon)
- Zeichen für Textfelder (z.B. “)
- Kaufmännische Schreibweise für Zahlen (z.B. ´050 für 50000)
- Position des Minuszeichens (vor oder nach der Zahl)
- Wissenschaftliche Schreibweise (1.158e-23)
- Datum und Zeitformate

Lesen von csv-Dateien (2)

Lesen von Zeitreihen

- Beispiele von Zeitreihen
- Messwerte
- Aktienkurse

Definition des Datum/Zeitformats: DIN ISO 8601

- Angabe in UTC: 2021-09-09T11:13:01.914243
- UTC: Universal Time Coordinated
- Angabe 31.10.2021 2:15 ist wegen der Umstellung von Sommer auf Winterzeit nicht eindeutig.

Genauigkeit des Zeitstempels:

- python datetime bis Mikrosekunden
- Pandas bis 1 Nanosekunde
- Numpy bis 1 Attosekunde (10^{-18})
- C# struct DateTime bis 100 Nanosekunden
- MS SQL Server, Datentyp datetime2 bis 100 Nanosekunden

Beispiele in 02-ReadTimeseriesCSV

```
Id,timestamp,LI10002/MonAnalog.PV_Out#Value,YC10001/Valve$Analog.MV#Value,P
795444,2018-10-10T14:52:04.9181551,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
795445,2018-10-10T14:52:05.9181551,148.569610595703,100,0.0740740299224854,
795446,2018-10-10T14:52:06.9181551,148.533294677734,100,0.0752314329147339,
795447,2018-10-10T14:52:07.9181551,148.603851318359,100,0.0752314329147339,
795448,2018-10-10T14:52:08.9181551,149.060119628906,100,0.0752314329147339,
795449,2018-10-10T14:52:09.9181551,149.266906738281,100,0.0746527910232544,
795450,2018-10-10T14:52:10.9181551,149.433074951172,100,0.0734953880310059,
795451,2018-10-10T14:52:11.9181551,149.565307617188,100,0.0729166269302368,
795452,2018-10-10T14:52:12.9181551,149.669342041016,100,0.0752314329147339,
795453,2018-10-10T14:52:13.9181551,149.745666503906,100,0.0752314329147339,
795454,2018-10-10T14:52:14.9181551,149.745666503906,100,0.0729166269302368,
795455,2018-10-10T14:52:15.9181551,149.712951660156,100,0.0706018209457397,
795456,2018-10-10T14:52:16.9181551,149.712951660156,100,0.0723379850387573,
795457,2018-10-10T14:52:17.9181551,149.593444824219,100,0.0659722089767456,
795458,2018-10-10T14:52:18.9181551,149.401885986328,100,0.0746527910232544,
795459,2018-10-10T14:52:19.9181551,149.34228515625,100,0.0717592239379883,1
795460,2018-10-10T14:52:20.9181551,149.307464599609,100,0.0758101940155029,
795461,2018-10-10T14:52:21.9181551,149.291961669922,100,0.0729166269302368,
795462,2018-10-10T14:52:22.9181551,149.303894042969,100,0.0677083730697632,
795463,2018-10-10T14:52:23.9181551,149.303894042969,100,0.0734953880310059,
795464,2018-10-10T14:52:24.9181551,149.345855712891,100,0.0700231790542603,
795465,2018-10-10T14:52:25.9181551,149.404846191406,100,0.0711805820465088,
795466,2018-10-10T14:52:26.9181551,149.449890136719,100,0.0717592239379883,
795467,2018-10-10T14:52:27.9181551,149.511108398438,100,0.0717592239379883,
```

Lesen von csv-Dateien (3)

Lesen von Zeitreihen

- \+ Sensormesswerte mit unterschiedlicher Abtastrate
- \+ Speicherung nur bei Änderung des Messwertes

Datei Zeitreihe-Abtastraten.csv

```
LI10002#time,LI10002#Value,YC10001#time,YC10001#Value,PI12002#time,PI12002#Value
2018-10-10T14:52:04.000,21.5,2018-10-10T14:52:04.000,235.4,2018-10-10T14:52:04.000,23
2018-10-10T14:52:05.000,22.5,2018-10-10T14:52:04.500,237.4,2018-10-10T14:52:05.100,24
2018-10-10T14:52:06.000,22.7,2018-10-10T14:52:05.000,238.4,,
2018-10-10T14:52:07.000,22.7,2018-10-10T14:52:05.500,234.4,,
,,2018-10-10T14:52:06.000,234.1,,
,,2018-10-10T14:52:06.500,234.0,,
,,2018-10-10T14:52:07.000,234.6,,
,,2018-10-10T14:52:07.500,233.2,,
```

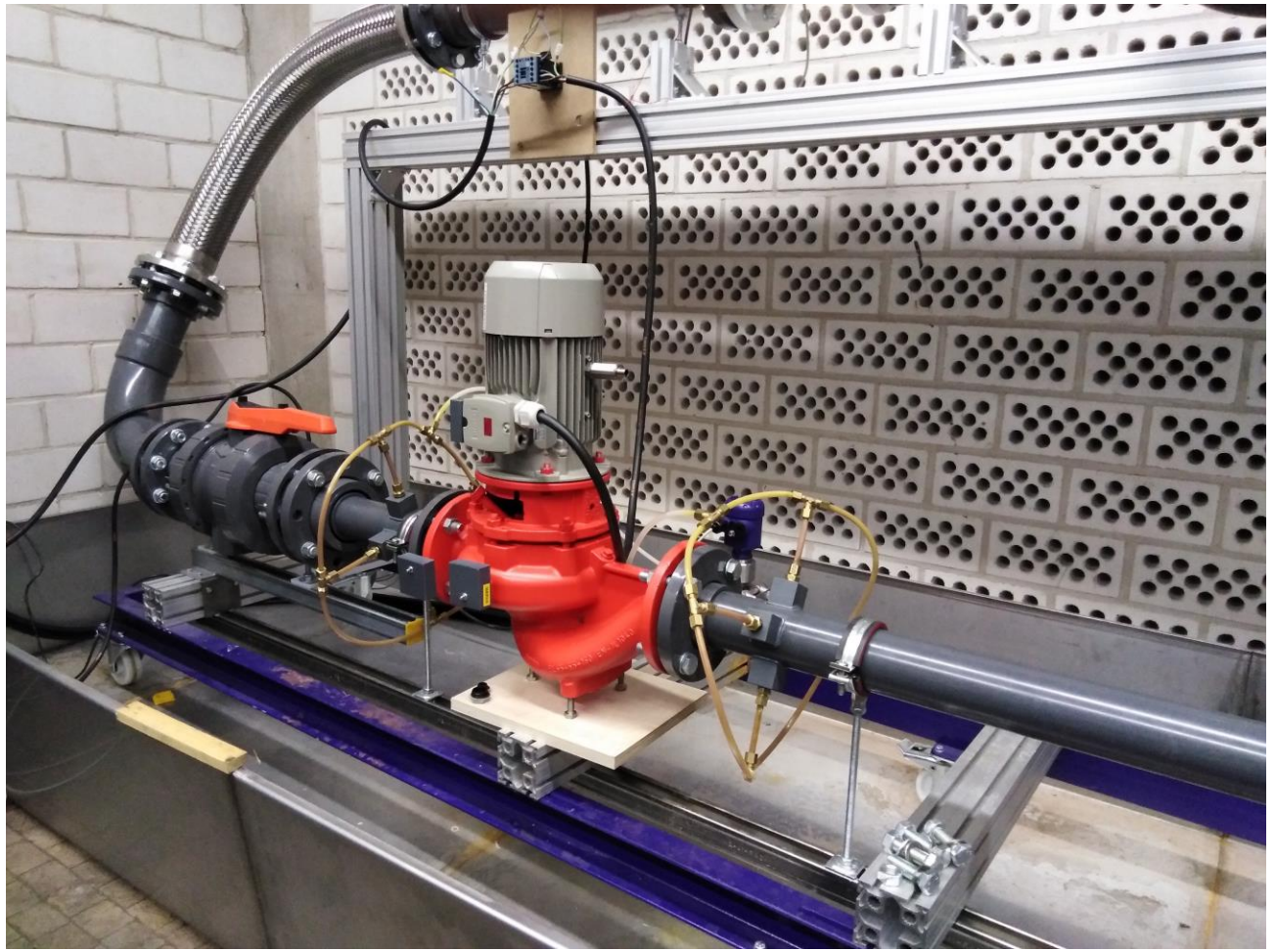
Lesen von csv-Dateien (4)

Lesen von Zeitreihen

+ Sensormesswerte mit hoher Abtastrate

Datei Zeitreihe-hohe-Abtastrate.csv

```
sensor;unit;starttime;samplingrate/Hz
Acceleration X-axis;m/s^2;2021-09-09T12:54:00.459;1e9
number;value
1;2.564
2;2.423
3;1.215
4;1.925
5;1.937
6;2.012
```



Übungsaufgabe 1: Einlesen der Zeitreihe 2

Lesen Sie die Datei 02-ReadTimeseriesCSV\Zeitreihe2.csv ein und wandeln Sie den Zeitstempel in ein datetime-Objekt – wie in readtimeseries1.py gezeigt – um.

- Welche Meldung erscheint?
- Warum erscheint diese Meldung?

Übungsaufgabe 2: Einlesen der Zeitreihe CricketX

Laden Sie die Zeitreihe CricketX unter
http://www.timeseriesclassification.com/Downloads/CricketX.zip

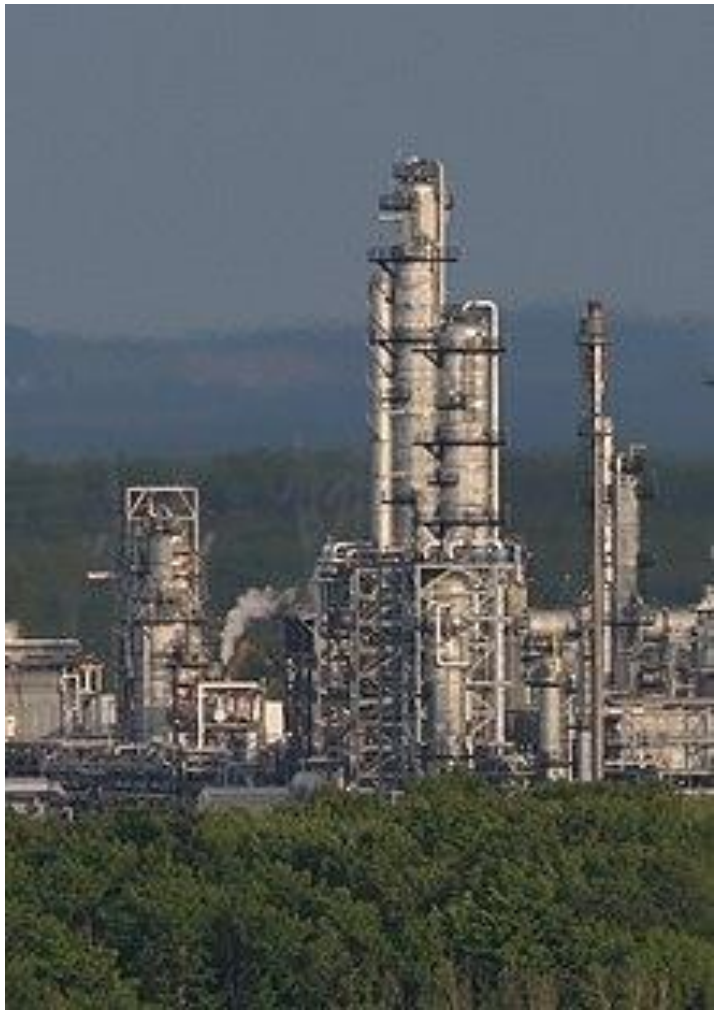
- Lesen Sie mit python die Daten in der Datei CricketX_TRAIN.txt in einen pandas dataframe ein. Berechnen Sie den kleinsten Wert (Minimum), den größen Wert (Maximum), den Mittelwert und den Median.
- Lesen Sie mit python die Daten in der Datei CricketX_TRAIN.arff in einen pandas dataframe ein.

Entwicklung eines Beispiels

Prozesstechnische Anlage in der chemischen Industrie

Für eine prozesstechnische Anlage fallen verschiedene Daten an, u.a.

- Daten für das Design des chemischen Prozesses
- Daten für die Planung und Dokumentation der Anlage
- Parametrierung der Steuer- und Messgeräte
- Messwerte während des Betriebs der Anlage
- Daten aus dem ERP-System (Auftrag zur Erstellung eines Produkts)
- Informationen zur Instandhaltung
- Daten über Ausgangsstoffe
- Daten über entstandene Produkte (Menge, Qualität)
- Informationen zu Störfällen
- Informationen zu Sicherheitsbetrachtungen



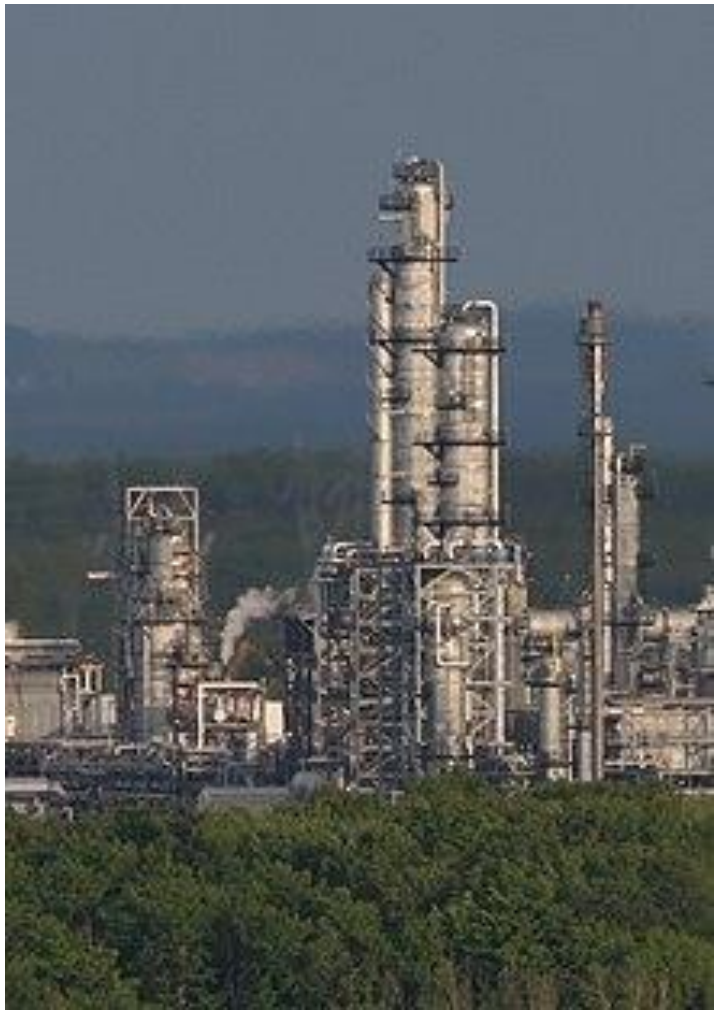
Quelle: https://de.wikipedia.org/wiki/Rektifikationskolonne

Entwicklung eines Beispiels

Prozesstechnische Anlage in der chemischen Industrie

Folgende Daten sollen abgefragt werden

- Kosten für Planung, Bau und Betrieb der Anlage (Total Cost of Ownership (TCO))
- Erkennung von Anomalien, Diagnose (mittels maschinellem Lernen)
- Daten zum Beobachten und Steuern der Anlage



Quelle: https://de.wikipedia.org/wiki/Rektifikationskolonne

Planungsdaten der Anlage (Schema der Anlage, R&I)

Struktur für Datenbank des R&I-Fließschema (Rohrleitungs- und Instrumentenfließschema), englisch P&ID

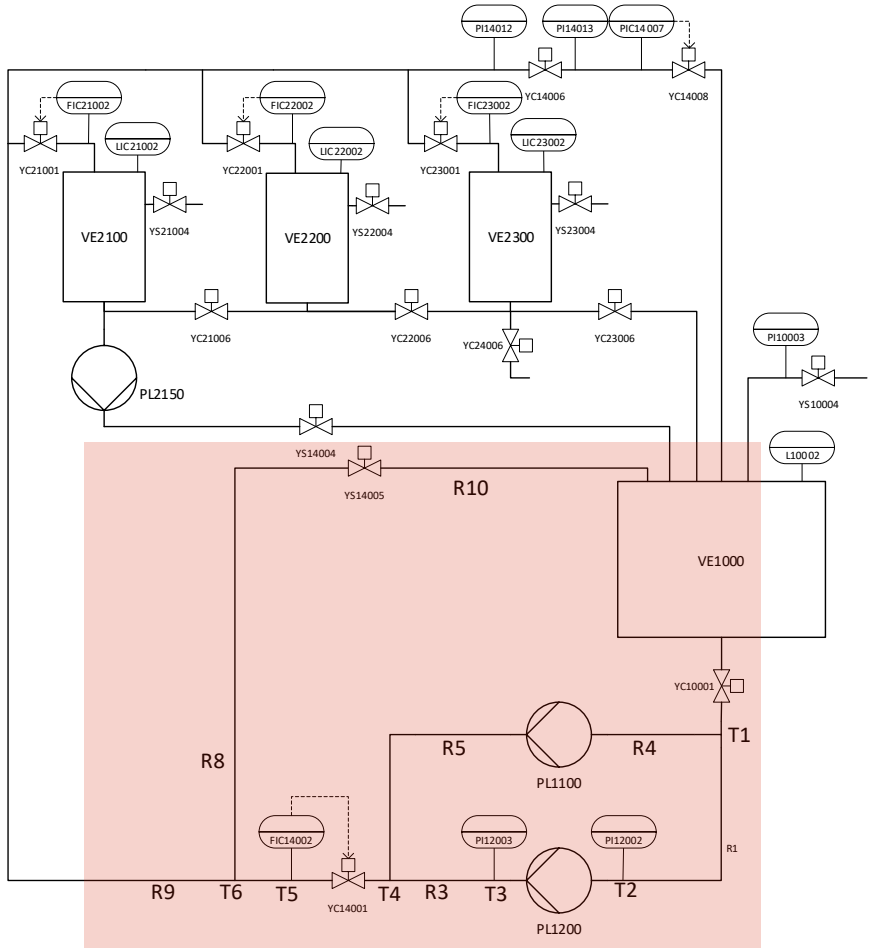
Nutzung vorhandener Standards

- VDI/VDE 3714 Implementierung und Betrieb von Big-Data-Anwendungen in der produzierenden Industrie
- ECLASS zur Speicherung von Produktstammdaten
- NE 100 Nutzung von Merkmalleisten im PLT-Engineering-Workflow (https://www.namur.net/de/empfehlungen-und-arbeitsblaetter/aktuelle-nena.html)

Tabellenstruktur

- AssetType, z.B. „Rohr“, „Pumpe“
- AKZ
- Entity-Relationship: ist_verbunden_mit
- Entity-Relationship: wirkt_auf

Übung 03-Schema-der-Anlage



Übung: Auslesen der Planungsdaten (1)

```
import pyodbc


server = 'md2c0gdc'

database = 'HSKA_Anlagenschema'

username = 'sa'

password = 'Password'

cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for S
QL Server};SERVER='+server+';DATABASE='+database+';
UID='+username+';PWD='+ password)

cursor = cnxn.cursor()


# select query for all assets

cursor.execute("SELECT * FROM [Assets_R_I]")

row = cursor.fetchone()
```

```
while row:

    print('ID {} hat den Typ {} mit dem Anlagenkenn
zeichen {}.'.format(row[0],row[1],row[2]))

    row = cursor.fetchone()
```

Übung 03-Schema-der-Anlage

Übung: Auslesen der Planungsdaten (2)

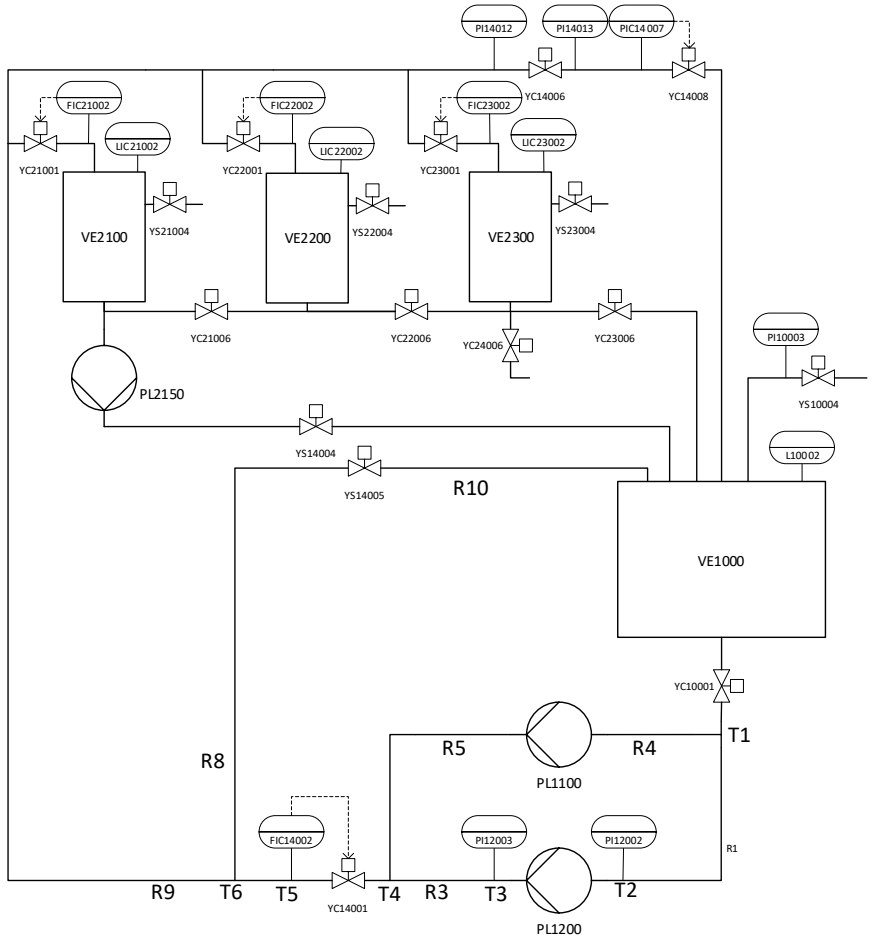
Mit welchen Anlagenteilen ist die Pumpe PL1100 verbunden?

```
cursor.execute("SELECT [AKZ1],[AKZ2] FROM [ist_verb
unden_mit] WHERE AKZ1='PL1100' OR AKZ2='PL1100'")
```

+ R4 ist verbunden mit PL1100.
+ PL1100 ist verbunden mit R5.

Welche Verbindungen gibt es von der Pumpe PL1100 zum Ventil YC14001?

+ Nutzung einer Graphdatenbank

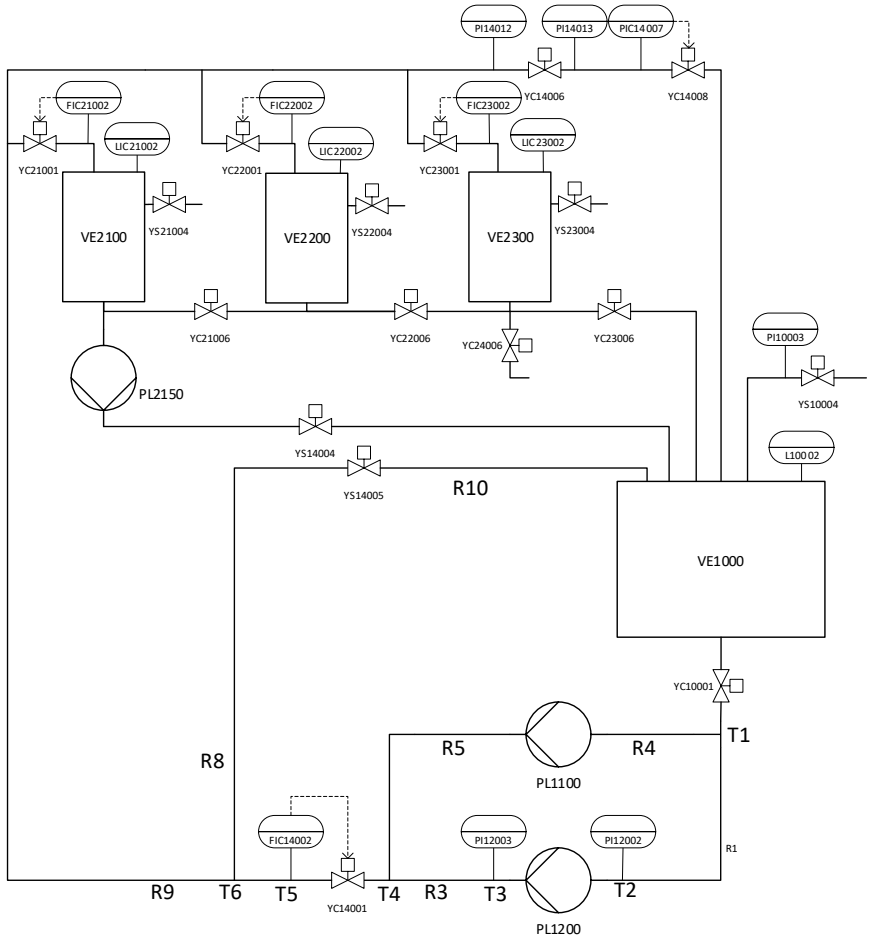


Graph-Datenbank

Übung: 04-Schema-der-Anlage-Graphdatabase

```
-- Find shortest path from vessel VE1000 to valve YC14001
SELECT AKZ1, ConnectedAssets
FROM (
SELECT
asset1.AKZ AS AKZ1,
STRING_AGG(asset2.AKZ, '->') WITHIN GROUP (GRAPH PATH) AS
ConnectedAssets,
LAST_VALUE(asset2.AKZ) WITHIN GROUP (GRAPH PATH) AS LastNode
FROM
Assets AS asset1,
connectedTo FOR PATH AS cTo,
Assets FOR PATH  AS asset2
WHERE MATCH(SHORTEST_PATH(asset1(-(cTo)->asset2)+))
AND asset1.AKZ = 'VE1000'
) AS Q
WHERE Q.LastNode = 'YC14001'
```

	AKZ1	ConnectedAssets
1	VE1000	YC10001->T1->R1->T2->R3->T4->YC14001

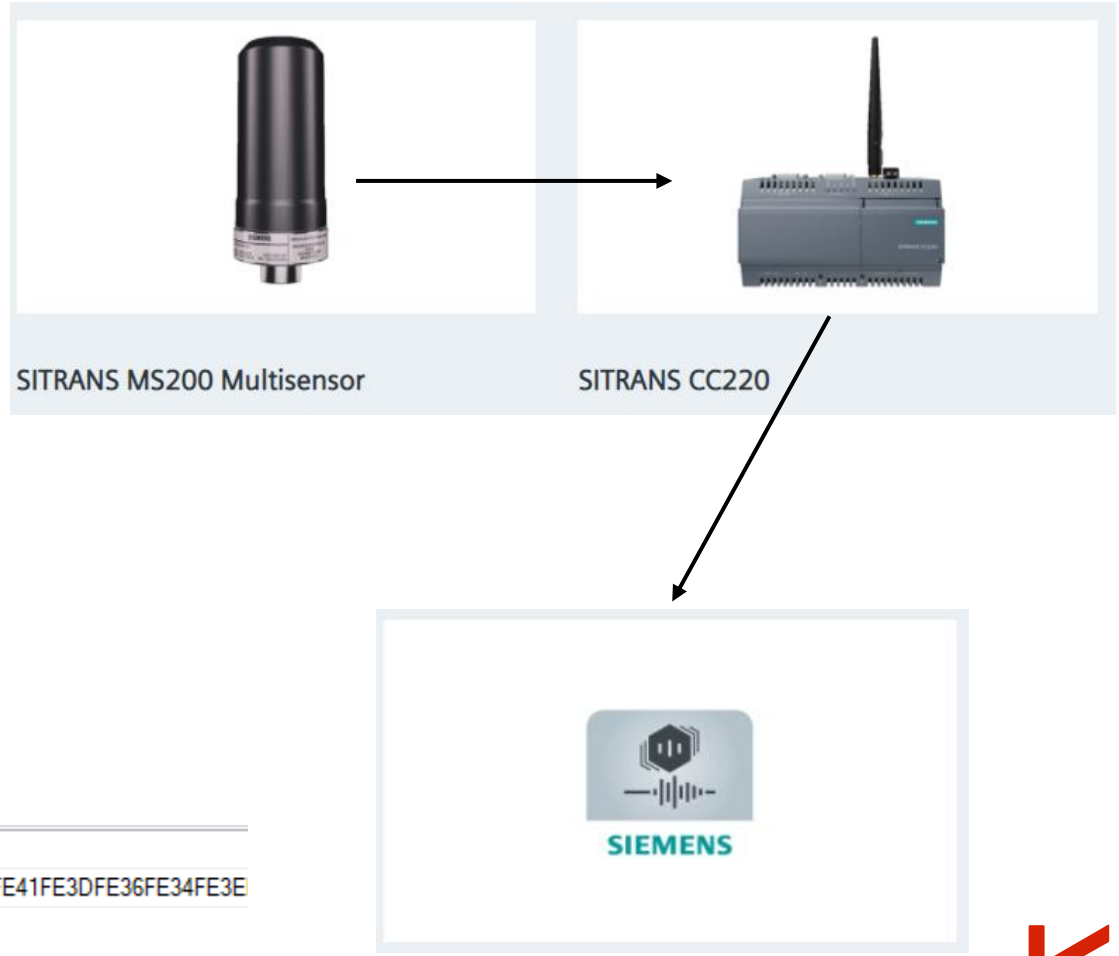


Quelle: https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-sample?view=sql-server-ver15
https://docs.microsoft.com/en-us/sql/relational-databases/graphs/sql-graph-shortest-path?view=sql-server-ver15

Daten im Binärformat (1)

SITRANS MS200 Multisensor

- Batteriegetrieben, Lebensdauer 5 Jahre
- Messung der Beschleunigung in x-, y- und z-Richtung
- Abtastrate (sampling rate) 6,6 kHz
- Messung von 512 Werten
- Einmal pro Minute wird eine Messung durchgeführt
- Pro Richtung fallen nach 5 Jahren $1{,}35 \cdot 10^9$ Messwerte an



Nachteile bei der Speicherung jedes einzelnen Messwertes

- Zeitstempel ist ungenau
  (datetime2 hat eine Auflösung von 100 ns)
- Hoher Speicherbedarf (Zeitstempel und Messwert)
- Datenbankabfrage ist langsam

Speicherung als Stream

Id	timestamp	UUID	streamlength	RawStreamData
2	2019-01-17 09:20:50.1990000	B080	512	0x24FE51FE41FE49FE45FE59FE49FE48FE40FE4AFE45FE32FE38FE41FE3DFE36FE34FE3E

Quelle: https://new.siemens.com/de/de/produkte/automatisierung/prozessinstrumentierung/digitalisierung/smart-condition-monitoring.html

Daten im Binärformat (2)

Extraktion der Streamdaten mit python

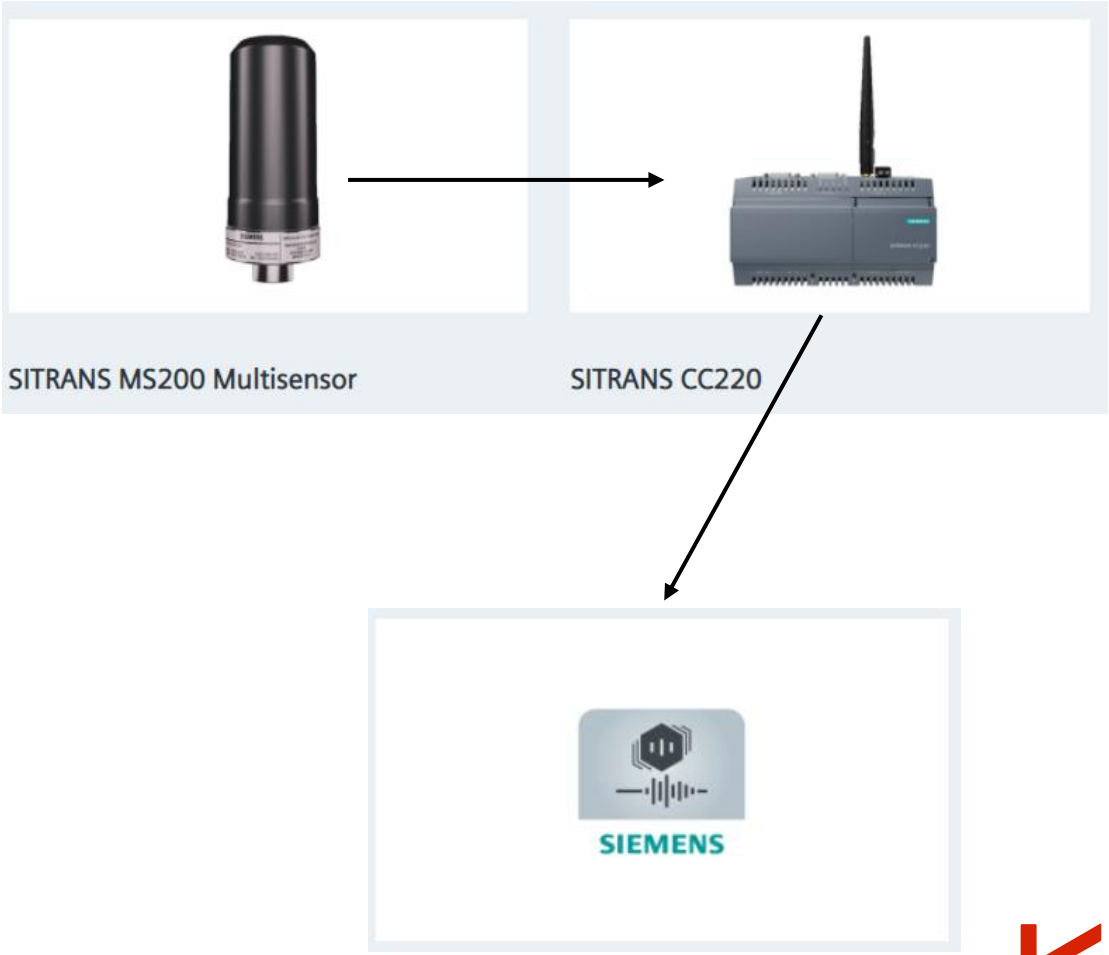
Übung 05-Data-Binary-Format

```
cursor.execute("SELECT TOP 1 RawStreamData FROM [ti
meseries_c464e3fe9e76lcsstream]")

row=cursor.fetchone()

valuesnp=numpy.frombuffer(row[0],dtype=np.int16)
```

+ Effizientes Lesen und Umwandeln der Daten im Binärformat mit `numpy.frombuffer`



Hochschule Karlsruhe
University of
Applied Sciences

www.h-ka.de