

Hochschule Düsseldorf

Fachbereich Medien

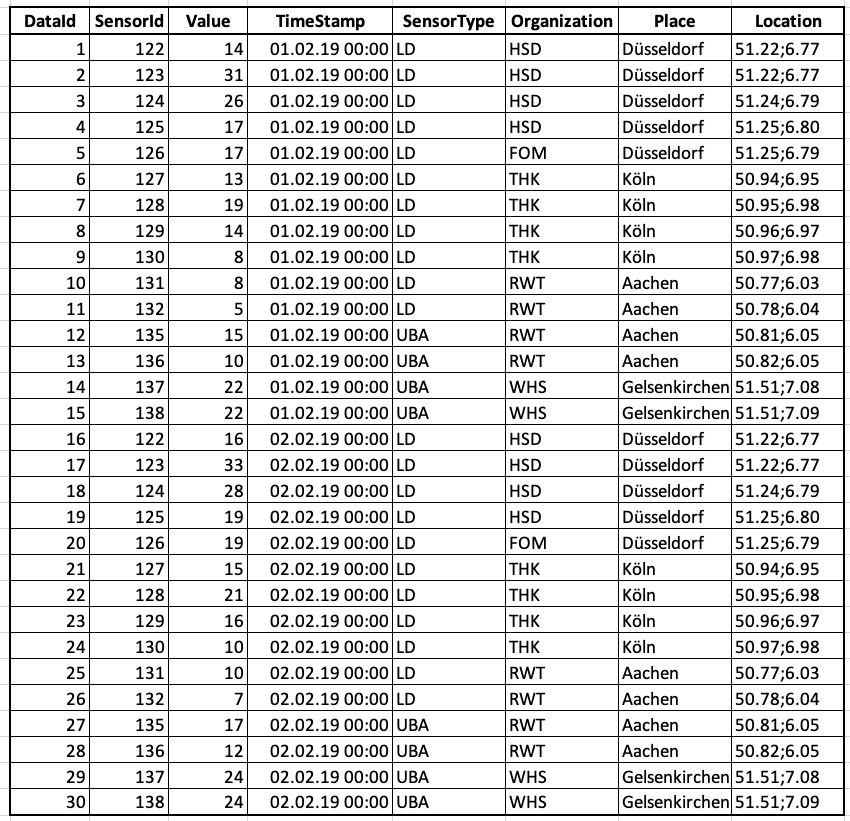
Prof. Dr.-Ing. Thomas C. Rakow

Professor für Informatik, insbesondere   
Datenbanken und E-Business

|  |  |  |  |
| --- | --- | --- | --- |
| Datenbanksysteme 1 | | Praktikum: Normalisierung und Funktionale Abhängigkeiten | |
| Studiengang | B. Sc. Medieninformatik | Bearbeiter | Rakow, Salgert, Drewers, Bringezu |
| Semester | SoSe 2023 | Datum | 09.07.2023 |
| Status | Fertig |  |  |

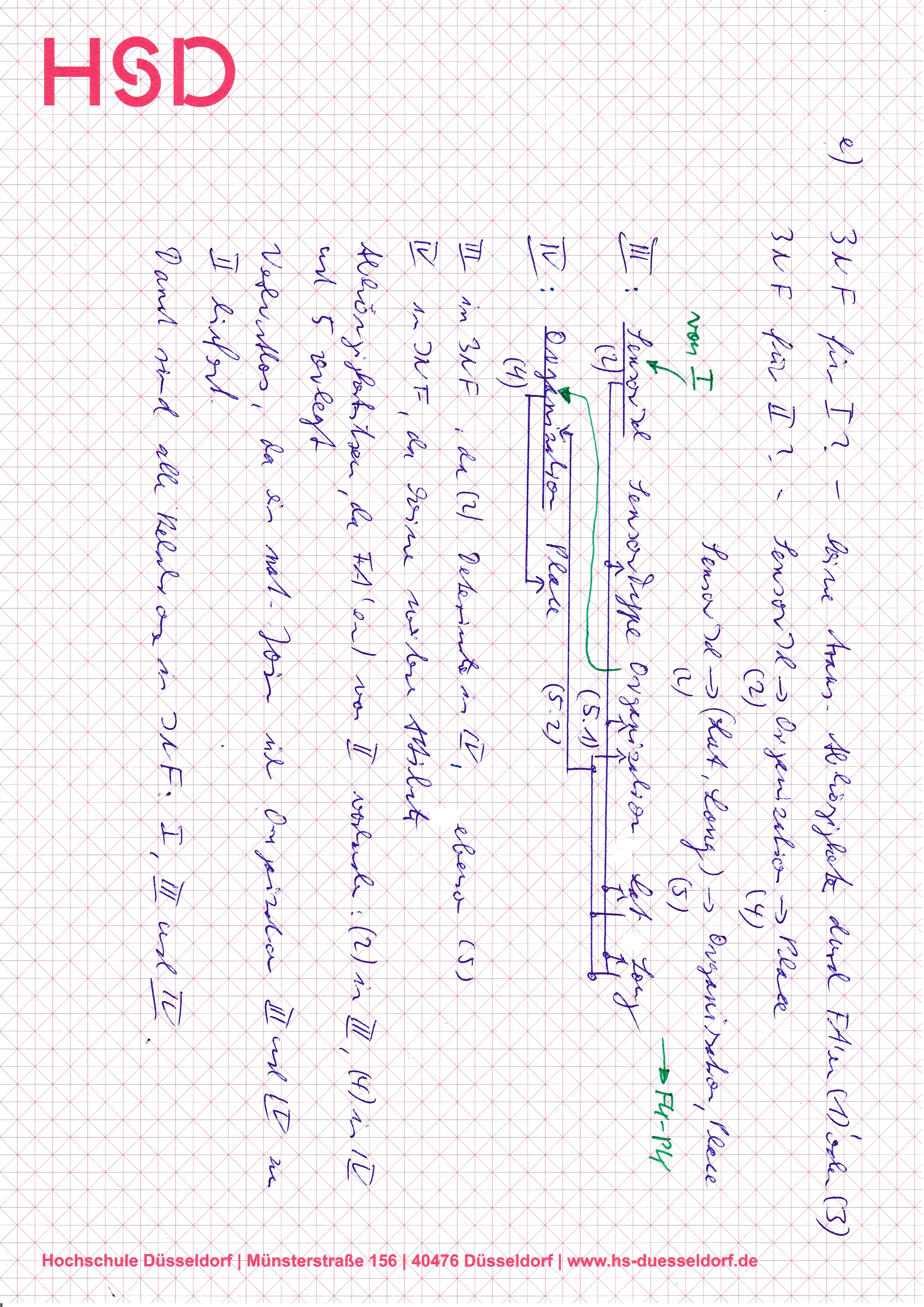
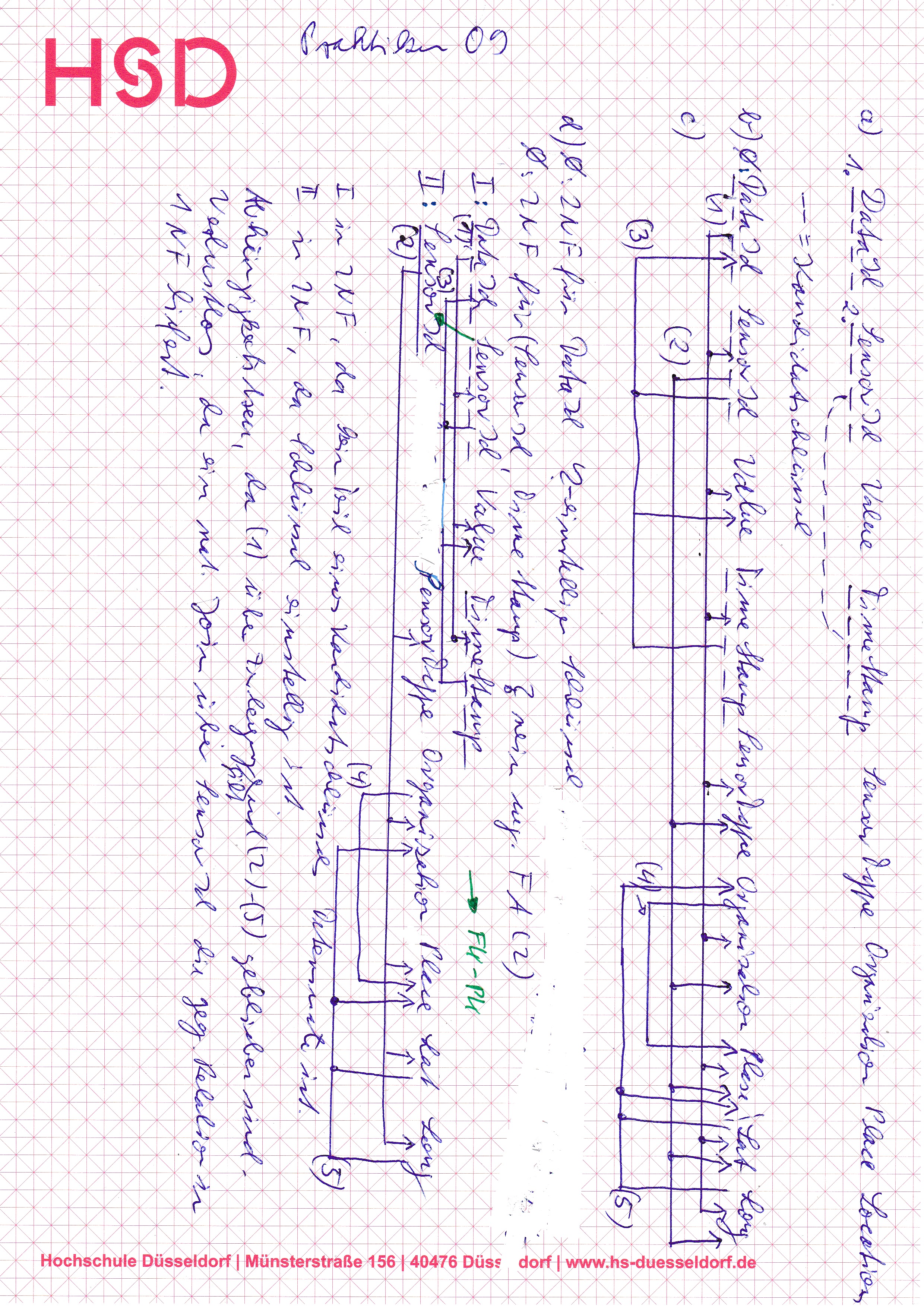
Aufgaben

Eine Anwendung zur Messung von Feinstaubdaten verwendet die folgende Tabelle. Jeder Datensatz mit dem Primärschlüssel **DataId** enthält im Attribut **Value** den gemessenen Wert eines Sensors **SensorId** zu einem bestimmten Zeitpunkt **TimeStamp**. Bei der Spalte **SensorType** handelt es sich um den Sensortyp. **Organization** beinhaltet den Betreiber des Sensors mit dessen Ortsangabe **Place**. Der Aufstellort des Sensors wird in Breiten- und Längengrad mit **Location** vermerkt.



1. Geben Sie alle Kandidatschlüssel an.
2. Bilden Sie die Tabelle auf genau eine äquivalente Relation in 1NF ab.
3. Bestimmen Sie alle vollen funktionalen Abhängigkeiten.
4. Bilden Sie auf die Normalformen ab. Erstellen Sie jeweils das Relationendiagramm (!) und begründen die Zerlegung mit Abhängigkeitstreue und Verlustlosigkeit
5. ... für die Zerlegung in 2NF.
6. ... für die Zerlegung in 3NF.

Lösung



Erklärung

1. Das Attribut **DataId** bestimmt und auch die beiden Attribute **SensorId** und **TimeStamp** zusammen bestimmen jeweils alle anderen Attribute. Wir kennzeichen die Kandidatschlüssel:

**DataId, SensorId, Value, TimeStamp, SensorType, Organization, Place, Location**

1. Das Attribut **Location** enthält zwei Zahlenwerte und widerspricht damit der Anforderung der 1NF nach atomaren (einfache) Werten. Wir interpretieren hier die Angaben als Längen- und Breitengrad und ersetzen **Location** durch die zwei Attribute **Latitude** und **Longitude**:

**DataId, SensorId, Value, TimeStamp, SensorType, Organization, Place, Latitude, Longitude**

Auch **TimeStamp** enthält mehere Werte, kann aber idR als Datentyp verwendet werden und wird daher als zusammengesetztes Attribut angenommen.

1. Wir bestimmen anhand der Wertekombinationen, ob eine Funktionseigenschaft gegeben ist, und wählen die Determinanten minimal:
2. **DataId** -> **SensorId, Value, TimeStamp, SensorType, Organization, Place, Latitude, Longitude**
3. **SensorId** -> **SensorType, Organisation, Place, Latitude, Longitude**

SensorId alleine ist kein Kandidatschlüssel der Relation in b).

1. **SensorId, TimeStamp** -> **Value, DataId**
2. **Organization** -> **Place**In anderen Anwendungen gilt die FA vielleicht nicht, hier aber anahnd der Daten schon!
3. **Latitude, Longitude** -> **Organization, Place**
4. Für die Erstellung in 2NF betrachten wir jede Relation, ob jeder Kandidatschlüssel volle Determinante für alle andeen Attribute ist. Wenn wir
5. einen einstelligen Schlüssel haben ist die Bedingung erfüllt,
6. einen mehrstelligen Schlüssel haben, prüfen wir, ob Teile davon Determinanten sind.

Hier haben wir für **DataId** den Fall i. vorliegen. Für den anderen Kandidatschlüssel **(SensorId, TimeStamp)** haben wir in der vollen FA (2) den Fall ii erfüllt. Also zerlegen wir anhand dieser sogenannten sekundären FA die Relation in zwei Relationen:

1. **DataId, SensorId, Value, TimeStamp**
2. **SensorId, SensorType, Organization, Place, Latitude, Longitude**

Die Relation I ist in 2NF, da kein Teil des zwiten Kandidatschlüssel noch Determinante ist.

Die Relation II besitzt nur einen Schlüssel, der als einstelliges Attribut als Primärschlüssel gesetzt wird. Mit einem einstelligen Primärschlüssel ist die Relation II in 2NF.

Zwischen I:**SensorId** und II:**SensorId** besteht eine FS-PS-Beziehung.

Die Zerlegung ist

1. abhängigkeitstreu: Die vollen FA’en sind bestehen geblieben oder wurden in einer Zerlegung verwendet:
2. mit Schlüssel aus Relation II und ohne die anderen Attribute aus II,
3. in Relation II,
4. ist in Relation I,
5. ist in II vorhanden,
6. ist in II vorhanden.
7. verlustlos: Wir können über einen natürlichen Join des „Verursachers“ **SensorId** die Datensätze der ursprünglichen Relation genau wiederherstellen.
8. Die Relation I. ist in 3NF, da keine transitiven Abhängigkeiten in dieser Relation vorhanden sind. Als Primärschlüssel wählen wir den Surrogateschlüssel **DataId**.

Bei Relation II finden wir die transitiven Abhängigkeiten

1. **SensorId** -> **Organization**

(4) **Organization** -> **Place**

und auch

1. **SensorId** -> **Latitude, Longitude**

(5) **Latitude, Longitude -> Organization, Place**

Wir zerlegen nach der ersten transitiven Abhängigkeit, denn zusammengesetzte Schlüssel sind nicht so schön. Eine Zerlegung anhand der transitiven FA vom Primärschlüssel

auf zwei Tabellen liefert anstelle von Relation II:

1. **SensorId, SensorType, Organization, Latitude, Longitude**
2. **Organization, Place**

Wir können in den beiden neuen Relationen innerhalb der FA’en keine transitiven Abhängigkeiten finden:

1. **SensorId** -> **Organization** –Organisation bestimmt Place in anderer Relation IV

(4) **Organization** -> **Place** – kein weiteres Attribut in Relation IV

(5) **Latitude, Longitude -> Organization, Place** – s. oben zu (2) und (4)

Damit sind alle Relationen in 3NF:

1. **DataId, SensorId, Value, TimeStamp**
2. **SensorId, SensorType, Organization, Latitude, Longitude**
3. **Organization, Place**

Zwischen I:**SensorId** und III:**SensorId** sowie III:**Organization** und IV:**Organization** besteht eine FS-PS-Beziehung.

Die Zerlegung ist

1. abhängigkeitstreu: Die vollen FA’en der Relation II sind bestehen geblieben oder wurden in einer Zerlegung verwendet:

(2) bleibt bestehen,

(4) nun in Relation IV,

(5) ist für **Organization** in III vorhanden,

(5) ist für **Place** mittels der verlustlosen Zerlegung mit **Organization** in Relation III bzw. als Schlüssel in IV indirekt vorhanden.

1. verlustlos: Wir können über einen natürlichen Join des „Verursachers“ **Organization** genau die – alle und nicht mehr – Datensätze der ursprünglichen Relation wiederherstellen.