# 2.1.1

Das relationale Datenbankmodell stellt eine Datenbank schriftlich ohne Grafik dar. Dabei wird jede Tabelle mit ihren Attributen angegeben. Primärschlüssel werden unterstrichen und Fremdschlüssel mit einem # gekennzeichnet.

Ein Vorteil des Relationalen Datenbankmodelles ist es das man in ihm die Fremdschlüssel erkennen kann. Ein Nachteil ist das man nicht bestimmen kann von wie vielen zu wie vielen die Beziehung ausgeht.

# 2.1.2

1. Interpretierung Aufgabe

Glaube das es keinen Sinn macht hierbei in neue Attribute und so aufzuteilen, da wir auch in anderen Aufgaben mit diesen bestimmten Attributen Arbeiten. Werde weiter mögliche Anpassungen nachträglich angeben.

1. Überführung aus dem ER-Modell in das Relationale Modell:

Hierbei implementiere ich die Auflösungstabellen direkt mit. Dabei wird die Beziehung Impfen trotz, dass sie nur eine n zu 1 Beziehung ist zu einer Auflösungstabelle da sie eigene Attribute hat. Genauso wird liefert eine Auflösungstabelle. Die Fremdschlüssel werden immer von der 1 Seite bei einer 1 zu n Beziehung genommen. Eine Auflösungstabelle integriert die Fremdschlüssel ihrer Nachbartabellen automatisch.

Person{personenID, risiko, name, adresse, gebDatum}

Impfen{#personenID, #bezeichnung, termin1, termin2}

Standort{ortID, temperatur, istAnszahlImpfdosen, kapazitaetImpfdosen, #bezeichnung}

Impfstoff{bezeichnung, lagerTemperatur, gruppe, entwickler}

Liefert{#bezeichnung, #prodID, anzahlImpfdosen, termin}

Produzent{prodID, land, name, plz, ort}

2. Regeln zur Normalisierung:

* Die **Nullte Normalform** ist dann gegeben, wenn alle **Informationen in einer Tabelle vorhanden** sind und noch **unnormalisiert vorliegen**.
* Die **Erste Normalform (1NF)** ist dann gegeben, wenn **alle Informationen** in einer Tabelle **atomar vorliegen**.
* Ein Relationstyp (Tabelle) befindet sich genau dann in der **zweiten Normalform (2NF)**, wenn er sich in der [ersten Normalform (1NF)](https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/erste-normalform/) befindet und jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten [voll funktional](https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/abhaengigkeiten-normalisierung/) abhängig ist.  Eine **vollständig funktionale Abhängigkeit** liegt dann vor, wenn dass Nicht-Schlüsselattribut nicht nur von einem Teil der Attribute eines zusammengesetzten Schlüsselkandidaten funktional abhängig ist, sondern von allen Teilen eines Relationstyps.
* Ein Relationstyp befindet sich genau dann in der **dritten Normalform (3NF)**, wenn er sich in der [zweiten Normalform (2NF)](https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/zweite-normalform/) befindet und kein Nichtschlüsselattribut [transitiv](https://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/abhaengigkeiten-normalisierung/) von einem Kandidatenschlüssel abhängt.

3. Normalisierung umsetzen:

1. Die Informationen liegen schon vollständig for.
2. Die Attribute sind schon Attomar.
   1. Bei Adresse bei Person ließe sich darüber streiten ob diese wegen Atomarität aufgeteilt werden muss. Allerdings glaube ich das es Zewecksgebunden hier extra um die Tabelle nicht zu groß zu machen in einem vorliegt.
3. Die vollfunktionalle Abhängigkeit liegt schon for.
4. Produzent währe nicht konform der 3. Normalform da Attribute die nicht der Primärschlüssel sind davon abhängig sind: allternative:

Person{personenID, risiko, name, adresse, gebDatum}

Impfen{#personenID, #bezeichnung, termin1, termin2}

Standort{ortID, temperatur, istAnszahlImpfdosen, kapazitaetImpfdosen, #bezeichnung}

Impfstoff{bezeichnung, lagerTemperatur, gruppe, entwickler}

Liefert{#bezeichnung, #prodID, anzahlImpfdosen, termin}

Produzent{prodID, land, name, plz}

Ort{plz, ort, land}