Datenbanken Praktikum 4

Aufgabe 1d)

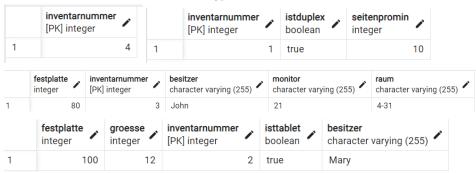
Single Table Strategy:

	festplatte integer	groesse integer	inventarnummer [PK] integer	istduplex boolean	isttablet boolean	seitenpromin integer	device_type character varying (31)	besitzer character varying (255)	monitor character varying (255)	raum character varying (255)
1	[null]	[null]	1	true	[null]	10	Drucker	[null]	[null]	[null]
2	100	12	2	[null]	true	[null]	Notebook	Mary	[null]	[null]
3	80	[null]	3	[null]	[null]	[null]	Desktop	John	21	4-31
4	[null]	[null]	4	[null]	[null]	[null]	Gerät	[null]	[null]	[null]

Joined Table Strategy:



Table-per-Concrete-Class Strategy:



Aufgabe 1e)

Single Table Strategy:

Eine riesige Tabelle mit allen Daten wird erstellt und jede Entität hat einen eigenen Tupel. Die nicht brauchbaren Daten bzw. die, die nicht für die bestimmten Geräte gebraucht werden, werden "null". Es ist ziemlich unübersichtlich aber speicherfreundlich weil nur eine Tabelle gebraucht wird.

Joined Table Strategy:

Jede Subklasse hat ihre eigene Tabelle, die nur die Felder der Subklasse und einen Fremdschlüssel auf die Basisklasse "Gerät" enthält. Es gibt kaum redundante Daten und es wird Nichts zwingend null. Abstrakte Klassen werden mit erstellt. Man kann alle Daten mit einfachen Joins ausgeben lassen.

Table-per-Concrete-Class Strategy:

Jede Tabelle enthält alle Felder der jeweiligen Klasse und ihrer übergeordneten Klassen. Lesbarkeit ist sehr gut. Es wird Nichts zwingend null. Subklassen haben keine Fremdschlüssel auf die Elternklassen, jedoch können dadurch die abstrakten Klassen, als Tabellen, weggelassen werden.