

## Datenbank Entwurf

### 0.1 Tabellen ohne Normalisierung: Tags

| ID | Tag |

### 0.2 Tabelle ohne Normalisierung: EventLabel

| ID | Name | OberLabel? |

### 0.3 Tabelle ohne Normalisierung: Events

Event: | ID | Start | Ende | **Label** | Name | DayCode | Repeating? |  
Eventlabel: | ID | Name |

### 0.4 Table ohne Normalisierung: Thema

| ID | Name | OberThema? |

### 0.5 Tabelle ohne Normalisierung: Karteikarten und Art

Karteiart: ID | Art |  
Karteikarten: | ID | **Thema** | Object | **Tag** | **EventLabel** | **KarteikartenArt** |

Nutzung von SQL Like wird uns helfen die einzelnen Veranstaltungen welche an mehreren Tagen laufen zu finden.

## 1 Vorläufige Zerlegung für Events

Wir unterscheiden zwischen sich wiederholende Events, ersetzende Events und einmaligen Events. Die Zerlegung dient zur Minimierung der doppelten Daten. Folgendes kam zur Stande: Wir trennen das Eventlabel von den Events

ID	Label
⋮	⋮

Wir entfernen Start und Endzeit aus der Datenbank und serialisieren das Event-Objekt. Zum ermitteln, welche Veranstaltung wann ist (bei den sich wiederholenden Events), nutzen wir einen Daycode (bspw. Mo,Di). Falls an mehreren Tagen, nutzen wir die Konkatenation MoDi etc. Suche erfolgt über Select... From ... where .. like ...

ID	Lable(ID)	Name	serialized Object	DayCode
⋮	⋮	⋮	⋮	⋮

Hierbei sind auch die sich wiederholenden Deadlines inbegriffen. Für einmalige Events wird nur der Daycode ersetzt durch ein Datum.

ID	Lable(ID)	Name	serialized Object	Date
⋮	⋮	⋮	⋮	⋮

Ersetzende Events durch beispielsweise Ausfälle sind wie folgt definiert:

ID	RepeatEvent(ID)	serialized Object	Date
⋮	⋮	⋮	⋮

Genauere Information werden dann durch die Suche im sich wiederholenden Event geholt. Das serialisierte Event beinhaltet auch zusätzliche Informationen wie Ort und Zeit - Minimierung erfolgt hier im Beschränkung der Suchparameter. Die Suchparameter sind folgende:

1. Name
2. Label
3. DayCode  $\oplus$  Datum

Weitere Überlegung: DayLeft als Attribut für einmalige Events: Kann man aber eventuell über den Datentype Date berechnen (Programm intern).