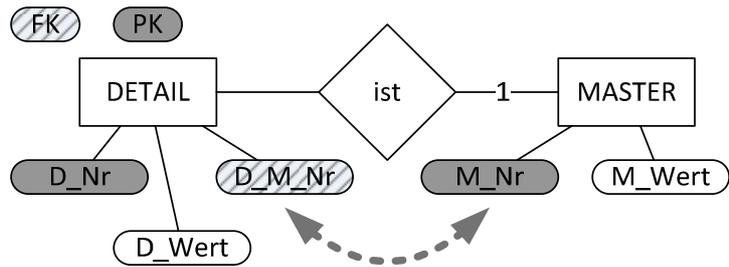


Ausgangssituation : Beziehungen

(Relationen) zwischen Entitäten werden durch Schlüsselattribute abgebildet. Im dargestellten Beispiel gehört zu jedem Datensatz der Detailtabelle genau ein DS der Mastertabelle und wird durch die Attribute MASTER.M_Nr bzw. DETAIL.D_M_Nr realisiert.



D.h. jeder Wert von DETAIL.D_M_Nr muss in MASTER.M_Nr vorhanden sein. In diesem Fall spricht man von **referentieller Integrität** zwischen beiden Tabellen.

Durch Eingaben (INSERT), Änderungen (UPDATE) oder Löschen (DELETE) kann diese verletzt werden (Anomalien). MySQL (Stand Vers. 5.0.x) ist in der Lage, die referentielle Integrität zu überwachen.

- Voraussetzungen** :
- Alle Tabellen müssen vom Typ InnoDB sein.
 - Die beiden Felder müssen vom selben Datentyp sein.
 - Beide Felder müssen mit einem Index ausgestattet sein.
 - Die Mastertabelle muss bereits existieren.

Syntax: FOREIGN KEY IN CREATE-STATE	
①	CREATE TABLE DETAIL (D_Nr INTEGER PRIMARY KEY ,
②	D_Wert CHAR (20) ,
③	D_M_Nr INTEGER ,
④	INDEX (D_M_Nr),
⑤	FOREIGN KEY (D_M_Nr) REFERENCES MASTER (M_Nr)
⑥	[ON DELETE { RESTRICT CASCADE SET NULL NO ACTION }]
⑦	[ON UPDATE { RESTRICT CASCADE SET NULL NO ACTION }]
⑧) ENGINE = InnoDB ;

In ① bis ③ erfolgt die normale Tabellendeklaration. Mit ④ wird für D_M_Nr ein INDEX angelegt. In Zeile ⑤ wird D_M_Nr als Fremdschlüssel mit Verweis auf MASTER.M_Nr deklariert (Tabelle MASTER muss vorhanden sein). Optional kann das Verhalten bei Integritätsverletzung durch Löschen ⑥ bzw. durch Ändern bestimmt werden. Mögliche Optionen sind RESTRICT (Defaultwert), CASCADE, SET NULL oder NO ACTION (identisch mit RESTRICT). Der Tabellentreiber muss vom Type InnoDB sein ⑦.

Syntax: FOREIGN KEY IN ALTER TABLE STATES	
①	ALTER TABLE DETAIL ADD [CONSTRAINT Name]
②	FOREIGN KEY (D_M_Nr) REFERENCES MASTER (M_Nr)
③	[ON DELETE Option] [ON UPDATE Option] ;

