SQL:

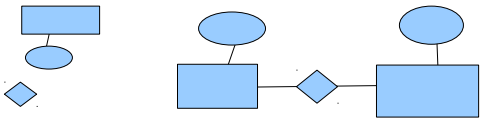
Tabelle erzeugen:	create table Name (Attribute)
Schlüssel:	primary key(Name)
	foreign Key (Name) references Tabelle(Name)
Abfragen:	select Projektionsliste [distinkt]
	from T
	where Bedingungen
T :	einzel Tabellen
	Tab1 natural join Tab2
	Tab1 join Tab2 on Tab1.attr=Tab2.attr
	Tab1,Tab2
Mengenops:	Abfrage1 OP Abfrage 2
	OP={ union, except}
Unterabfragen:	select bla bla..
	from bla ,(Abfrage) as TMP
	where Attr [not] in(Abfrage)
Änderungen:	insert into Tab [(Attributliste)] values (Werte)
	update Tab set attr=W, att2=W2 [where ..]
	delete from Tab [where..]
where Ausdruck:	>=,<=,=,<> , log. Verknüpfungen (and,or,not)
	Like (%=bel.viele Zeichen, _ ein Zeichen)
Typkonvert:	cast(Attr as Typ)

Relationenalgebra: Operationen: $A \cup B \; A \cap B \; A - B$

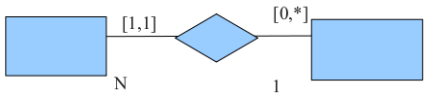
$\pi_{Spalten}(Tabelle)$	=Projektion (entfernen doppelter Einträge)
$\sigma_{Bed}(Tabelle)$	=Selektion
$\beta_{neu \to alt}(Tabelle)$	=Umbenennung
Tab1 $\bowtie$ Tab2	= Verbund (nat. join)

ER-Modell:

Entitytypen:
Attribute:
Beziehungstypen:



Kardinalität: 1:N, 1:1, M:N



ER → Relationen

Entity-Typen + Beziehungen → Relationenschemata
Attribute → Attribute der Relationsschemata + Übernehmen der Schlüssel
Kardinalitäten → durch Wahl der Schlüssel eindeutig
in einigen Fällen →Verschmelzung der RS von ETs + Beziehungen
Beziehungstypen: neues RS mit allen Attributen des Bts+Übernahme PS
m:n → beide PS zusammen bilden PS im RS
1:n → PS n Seite → neuer PS
1:1 → beide PS werden je ein PS im neuen RS,
PS wird dann ausgewählt (von den beiden)
Verschmelzungen: 1:n → die n Seite kann in das RS der Beziehung übernommen werden
1:1 → beide ET können in das RS der Beziehung übernommen werden

ER-Konzept	relationales Konzept
ET Attr von ET PS	RS Attr RS PS
BT Attr des BT kard. siehe oben	Primärschlüssel der Ets Attr
IST Bez	RS bekommt noch P1

Ableitungsregeln

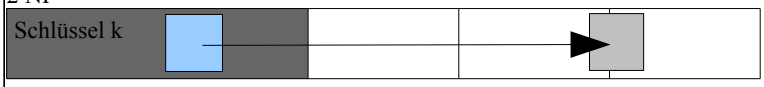
F1: Reflexivität	X ⊇Y ⇒ X → Y
F2: Augmentation	{ X →Y } ⇒ XZ →YZ sowie XZ →Y
F3: Transitivität	{ X →Y , Y → Z } ⇒ X →Z
F4: Dekomposition	{ X →YZ } ⇒ X →Y
F5: Vereinigung	{ X →Y , X →Z } ⇒ X →YZ
F6: Pseudotransitivität	{ X →Y , WY → Z } ⇒ WX → Z

F1-F3 = Armstrong Axiome

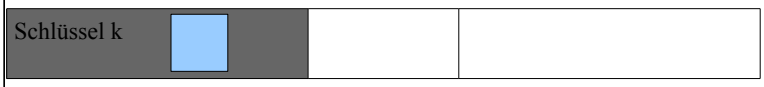
Normalformen(NF):
1 NF: nur atomare Attribute
2 NF: Eliminierung von part. Abhängigkeiten bei NichtschlüsselAttr
3 NF: Eliminierung von tran. Abhängigkeiten (NichtschlüsselAttr als Ende)
Boyce Codd NF: Eliminierung von tran. Abhängigkeiten zwi. Primär-Attr

Transformationseigenschaft	Kurzcharakteristik
Abhängigkeitstreue	alle gegebenen Abhängigkeiten sind durch Schlüssel repräsentiert
Verbundtreue	Orginalrelationen können durch den Verbund der BasisRela wiedergewonnen werden

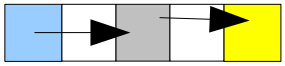
2 NF



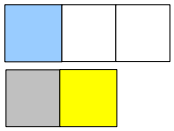
wird zu:



3 NF



wird zu:



Bereichskalkül:

$R \cup S$ → { Attribute | $R(..) \vee S(..)$ }
$R - S$ → {Attr| $R(..) \wedge \neg S(..)$ }
R nat join S → { Attr| $R(.., v_1, .., v_n, ..) \wedge S(..., v_1, .., v_n, ..)$ }

QBE:

Projektion:	mit P. markierte Spalten
Selection:	Vergleiche als Spalteneinträge oder Condition Box
Umbenennung:	explizite Ausgabetab
Verbund:	mittels Beispielelementen (Bereichsvariablen)

Assertion-Klausel: create assertion Name check(Prädikat)

Trigger: create Trigger Name
after| before Ereignis
on Relation | where |
referencing old|new as Name
begin atomic
SQL Anweidung
end;
Ereignis={insert, update,delete}

Sichten: create view Name as SQL-Anweisung

Steve Göring stg7@gmx.de