

1 SQL

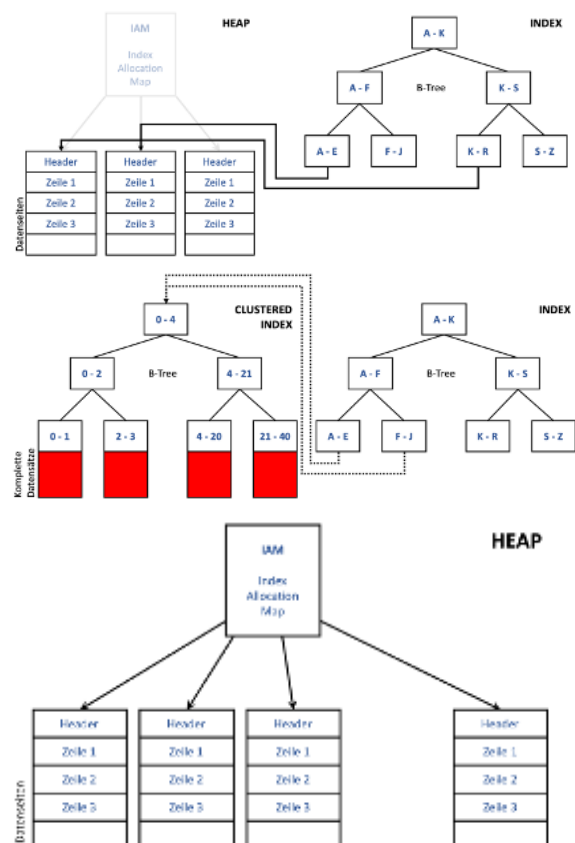
Command	Description
USE	wechselt den Ausführungskontext auf eine bestimmte Datenbank. USE master , in diesem Fall wird zur Metadaten-DB gewechselt.
SELECT	SELECT [Column name] FROM [Table name]
UPDATE	UPDATE [Table name] SET [column name] = [value], ... WHERE [condition]
DELETE	DELETE FROM [Table name] WHERE [condition]
INSERT INTO	INSERT INTO [Table name] ([column1, column2,...]) VALUES ([value1, value2,...])
GO	wird verwendet, um die Ausführung zu erzwingen.
CREATE DATABASE	kreiert eine neue Datenbank, zu welcher man mit USE wechseln kann
DROP DATABASE	schmeisst die Datenbank aus dem Fenster
ON	gibt an, wo die Daten physisch gespeichert werden.
CREATE TABLE	um eine Tabelle zu kreieren, CREATE TABLE [table name] ([Coll1, coll2 etc])
DROP TABLE	schmeisst die Table aus dem Fenster
ALTER TABLE	ALTER TABLE [Table name] [DROP / ALTER] COLUMN [column name] [datatype (only if ALTER)]
CONSTRAINT	bedingung → z.B Key CONSTRAINT [key name] PRIMARY KEY [type (optional)] ([referenz]) bzw CONSTRAINT [key name] FOREIGN KEY ([target]) REFERENCES [tabelle]([spalte])
CREATE INDEX	CREATE [type] INDEX [index name] ON [table] ([spalte]) man kann dann mit INCLUDE andere Spalten includen.
DROP INDEX	schmeisst den Index zum Fenster raus

3 Indexes

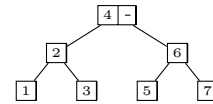
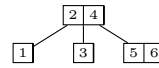
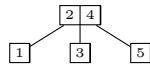
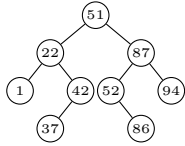
2 Data Type

Data type	use
bigint	64 bit number
int	32 bit number
smallint	16 bit number
tinyint	8 bit number
bit	1 bit number
decimal(precision, scale)	floating point number
numeric	same as decimal
money	64 bit int shifted
smallmoney	32 bit int shifted
float(n)	float 1 - 24
real	float(24)
datetime	date and time 3ms
smalldatetime	date and time 1min
char	char max 8000
varchar(n)	use this instead of char
nchar	char in unicode
nvarchar(n)	varchar in unicode
text	long texts
ntext	unicode text
binary	malware
varbinary(n)	use this instead of binary
image	binary, but longer
cursor	reference as cursor
sql_variant	never use this
table	query result for later usage
timestamp	timestamp
uniqueidentifier	GUID

- NONCLUSTERED
- CLUSTERED



4 binary search tree and b-tree



4.1 b-tree ersparniss

$$\text{maximale Schritte} = \frac{\log_{10}(n)}{\log_{10}(\text{Anzahl Verzweigungen})} \text{ vs } \text{Heap } n \rightarrow \emptyset \frac{n}{2}$$

5 DM

Element	Voranalyse	Konzeptionelles DM	Logisches DM
Entitäten Namen	X	X	
Entitäten Beziehungen	X	X	
Attribute Namen		X	
Primärschlüssel		X	X
Fremdschlüssel			X
Tabellen Namen			X
Spalten Namen			X
Datentypen			X

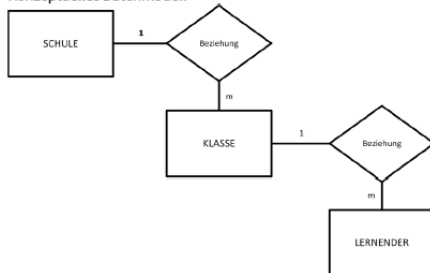
6 4 Fälle

- überlappend (X kann auch Y sein)
- disjunkt (X kann nicht auch Y sein)
- total (jedes X muss auch zu Y gehören)
- partiell (X muss nicht unbedingt zu Y gehören)

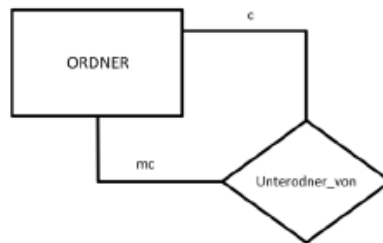
7 Beziehungen

7.1 Hierarchisch

Konzeptuelles Datenmodell

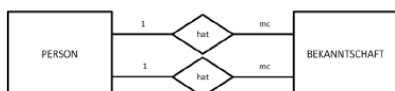
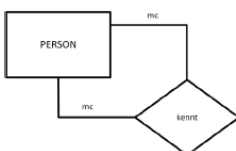


Konzeptuelles DM



7.2 Netzwerkartig

(Die zweite Darstellung ist zu bevorzugen)



8 Spezialisierung / Generalisierung

- Superklasse und Subklasse je eine Tabelle (bekannt)
- Eine Klasse pro Subklasse (keine Superklasse)
- Alles in einer Tabelle1 zusätzliches Attribut
- Alles in einer Tabelle mit 1 zusätzlichen Attribut pro Subklasse

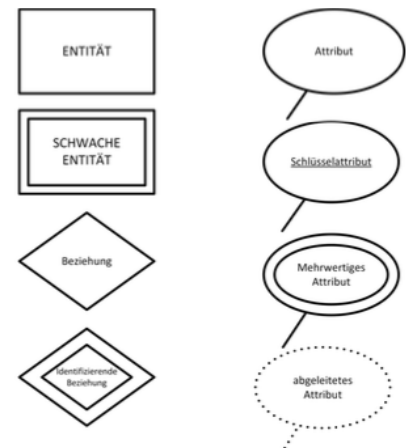
8.1 Spezialisierung

Top Down Sicht, ein in mehrere aufteilen.

8.2 Generalisierung

Bottom Up Sicht, mehrere zusammenfassen.

9 Diagramm



- UML
- Nur Attribut namen in Tabelle
- Alle Details in Tabelle

9.1 TADESI

T	Tabellenname
A	Attribute
D	Datentypen
E	Constraints (NULL, DEAFULT)
S	Schlüsselarten (PK, FK)
I	Indexe