



1. Kartei erstellen

Um ein sinnvolles Karteikartensystem zu entwickeln, müssen Sie vorgängig einige Fragen klären:

Welche Merkmale zur Beschreibung der Person sind für uns wesentlich?

Wie sollen diese Merkmale auf der Karteikarte organisiert sein, damit die Daten gezielt abgerufen werden können? Wo sind die wesentlichen Merkmale, wo die unwesentlichen?

Wie kann ich einfach eine Möglichkeit schaffen Karteikarten (Personen) auswählen zu können, deren Merkmale auf ein festgelegtes Suchkriterium zutreffen?

Aufgabe: 10 Karteikarten mit mind. 12 Angaben zu einer Person
Arbeitsform: Team
Zeit: 60 Minuten

Erstellen Sie Karteikarten für 10 Personen. Die Form der Karteikarte können sie frei wählen. Auch die Angaben auf den Karteikarten sind frei wählbar. Die Karteikarten sollten nach verschiedenen Kriterien sortiert werden können. Im Folgenden sehen sie ein Beispiel einer Karteikarte. Die Sortierungskriterien sind Folgende: Nachname, Sport, Musik, Party, Geschlecht.

C Alphabet des Nachnamens!		
Cuche	Sport	X
Didier	Musik	
Fribourg	Party	X
14.06.1976		
+4179 453 22 34		
Haarfarbe	blond	
Gewicht	81kg	
Körpergrösse	182cm	
M Geschlecht		

Betreibt Sport oder nicht?
Macht Musik oder nicht?
Geht an Partys oder nicht?

Angabe zur Haarfarbe!
Angabe zum Gewicht in Kilos!
Angabe zur Körpergrösse cm!



2. Datentypen

2.1. MySQL

Wenn man Daten in einer Datenbank ablegt, muss man sich vorgängig unbedingt Gedanken dazu machen, welche Werte mit welchem Datentyp abgespeichert werden. Dies ist eine wichtige Voraussetzung für eine integere Datenbank. MySQL unterstützt mehrere Datentypen in den folgenden Kategorien: numerische Typen, Datums- und Zeittypen und String- (Zeichen-) Typen. Nachfolgend sehen sie eine Übersicht zu diesen Datentypen.

Ganze Zahlen:

Datentyp	Speicherplatz	Optionen	Beschreibung
TINYINT	1 Byte	[(M)] [U] [Z]	Ganzzahlen von 0 bis 255 oder von -128 bis 127.
SMALLINT	2 Bytes	[(M)] [U] [Z]	Ganzzahlen von 0 bis 65.535 oder von -32.768 bis 32.767.
MEDIUMINT	3 Bytes	[(M)] [U] [Z]	Ganzzahlen von 0 bis 16.777.215 oder von -8.388.608 bis 8.388.607.
INT	4 Bytes	[(M)] [U] [Z]	Ganzzahlen von 0 bis ~4,3 Mill. oder von -2.147.483.648 bis 2.147.483.647.
INTEGER	4 Bytes	[(M)] [U] [Z]	Alias für INT.
BIGINT	8 Bytes	[(M)] [U] [Z]	Ganzzahlen von 0 bis $2^{64}-1$ oder von $-(2^{63})$ bis $(2^{63})-1$.

Fliesskommazahlen:

Datentyp	Speicherplatz	Optionen	Beschreibung
FLOAT	4 Bytes	[(M,D)] [U] [Z]	Fließkommazahl mit Vorzeichen. Wertebereich von $-3,402823466^{38}$ bis $-1,175494351^{38}$, 0 und $1,175494351^{38}$ bis $3,402823466^{38}$.
DOUBLE	8 Bytes	[(M,D)] [U] [Z]	Fließkommazahl mit Vorzeichen. Wertebereich von $-1,798^{308}$ bis $-2,225^{308}$, 0 und $2,225^{308}$ bis $1,798^{308}$.
REAL	8 Bytes	[(M,D)] [U] [Z]	Alias für DOUBLE.
DECIMAL	M+x Bytes	[(M,D)] [U] [Z]	Fließkommazahl mit Vorzeichen. Speicherbedarf: x=1 wenn D=0, sonst x=2. Ab MySQL 5.1 binär gespeichert, zuvor als String.
NUMERIC	M+x Bytes	[(M,D)] [U] [Z]	Alias für DECIMAL.

Datum und Zeit:

Datentyp	Speicherplatz	Optionen	Beschreibung
DATE	3 Bytes	-	Datum im Format 'YYYY-MM-DD'. Wertebereich von 01.01.1000 bis 31.12.9999.
DATETIME	8 Bytes	-	Datumsangabe im Format 'YYYY-MM-DD'



			hh:mm:ss'. Wertebereich entspricht DATE.
TIMESTAMP	4 Bytes	-	Zeitstempel. Wertebereich: 1.1.1970 bis 19.01.2038. Das Format variiert in den MySQL-Versionen.
TIME	3 Bytes	-	Zeit zwischen -838:59:59 und 839:59:59. Ausgabe: hh:mm:ss.
YEAR	1 Byte	[(2 4)]	Jahr zwischen 1901 bis 2155 bei (4) und zwischen 1970 bis 2069 bei (2).

String:

Datentyp	Speicherplatz	Optionen	Beschreibung
CHAR	M Byte(s)	(M) [BINARY]	Zeichenkette fester Länge M. Wertebereich für M: 0 bis 255.
VARCHAR	L+1 Bytes	(M) [BINARY]	Zeichenkette variabler Länge, Maximum ist M. Wertebereich für M: 0 bis 255.
BINARY	M Bytes	(M)	Zum Speichern binärer Strings, unabhängig vom Zeichensatz. Wertebereich für M: 0 bis 255. Weiterer Typ: VARBINARY
BLOB	L+2 Bytes	(M)	Binäres Objekt mit variablen Daten. Weitere Typen: TINYBLOB, MEDIUMBLOB und LONGBLOB. M ist ab Version 4.1 definierbar.
TEXT	L+2 Bytes	(M)	Wie BLOB. Ignoriert beim Sortieren & Vergleichen Groß- und Kleinschreibung. Weitere Typen: TINYTEXT, MEDIUMTEXT, LONGTEXT. M ist ab Version 4.1 definierbar.
ENUM	1 oder 2 Bytes	('val1', 'val2')	Liste von Werten (val1, val2, ...). 65.535 eindeutige Elemente sind maximal möglich.
SET	x Bytes	('val1', 'val2')	String-Objekt mit verschiedenen Variablen. 64 sind maximal möglich. Speicherbedarf: x ist 1, 2, 3, 4 oder 8.

Legende

^	= Potenzzeichen
[]	= Optionaler Parameter
BINARY	= Attribut für die Sortierung
D	= Anzahl der Kommastellen bei einer Dezimalzahl
L	= Stringlänge (Berechnung Speicherbedarf)
M	= Maximale Anzahl der gezeigten Stellen
Mill.	= Milliarden
U	= UNSIGNED (Zahl ohne Vorzeichen)
Z	= Zerofill



2.2. Access

Im Folgenden sehen Sie eine Aufstellung der Datentypen in Access:

Datentyp	Beschreibung	Format / Beispiele
Text	alphanumerische Zeichen (bis 255 Zeichen), Ziffern können mathematisch nicht verarbeitet werden.	
Memo	alphanumerische Zeichen (max. 65'535 Zeichen), Ziffern können math. nicht verarbeitet werden.	
Zahl	numerische Zeichen, Ziffern können mathematisch verarbeitet werden.	Byte: ganzzahlige positive Zahlen von Integer: ganzzahlige Zahlen von Long Integer: ganzzahlige Zahlen von Single: Zahlen mit max. 7 Dezimalstellen von Double: Zahlen mit max. 15 Dezimalstellen von
Datum / Zeit	Datum und / oder Zeitangaben, können mathematisch verarbeitet werden.	
Ja / Nein	Boolescher Wert, es sind nur zwei Werte ("0" / "1") möglich, kann logisch verarbeitet werden	
OLE-Objekt (<i>objekt linking and embedding</i>)	verknüpfte oder eingebettete Daten	

Access bietet noch weitere Datentypen (*Felddatentypen*) an. Im Modul 100 werden diese jedoch nicht aufgegriffen. Zur Vervollständigung seien sie an dieser Stelle jedoch kurz aufgeführt:

Datentyp	Beschreibung	Format
Währung	numerische Zeichen mit Währungsangabe	
Hyperlink	Verknüpfungen zu Dateien, welche per Mausklick aufgerufen werden können	
Autowert	numerisches Zeichen, welches automatisch fortlaufend von Access vergeben wird	