

Inhaltsverzeichnis

1	Was sind Datenbanken	2
1.1	Überblick und Motivation	2
1.2	Architekturen	2
1.3	Einsatzgebiete	3
1.4	Historisches	3
2	Relationale Datenbanken	4
2.1	Relationen für tabellarische Daten	4
2.2	SQL-Datendefinition	4
2.3	Grundoperationen: Die Relationenalgebra	5
2.4	SQL als Anfragesprache	5
2.5	Änderungsoperationen in SQL	5
3	Entity-Relationship-Modell	6
4	Datenbankentwurf	7

1 Was sind Datenbanken

TODO Kapitel zuende zusammenfassen oder so, es ist echt nur einföhrung und wenig inhalt ist hier mit bei der wichtig sein könnte - nett zum veranschaulichen

1.1 Überblick und Motivation

Die Idee einer Datenbank ist, dass sie einmal Informationseinheiten logisch mit einander verknüpft, zum anderen Massen an Daten verwaltet. Unter der Verwaltung gehört unter anderem die Sicherheit vor Verlusten und dass mehr als ein Benutzer auf einmal die Daten Abfragen kann. Beispiele in unserer heutigen Welt sind Social-Media Seiten, Einkaufswebseiten wie Amazon und Navigationsseiten wie Google Maps.

Die meisten Softwaresysteme sind nicht darauf ausgelegt große Mengen von Daten effizient zu speichern, was dafür sorgt, dass die gesuchten Daten entweder den Speicher zu müllen oder lange brauchen gefunden zu werden. Außerdem können mehrere Benutzer nur in wenigen Fällen parallel (gut) auf den selben Daten arbeiten.

1.2 Architekturen

Laut einem Ted Codd muss eine Datenbank 9 Sachen beinhalten/erfüllen, die 9 Codd'schen Regeln:

1. Integration: einheitliche nichtredundante Datenverwaltung
2. Operationen: Speichern, Suchen, Ändern
3. Katalog: Zugriffe auf Datenbankbeschreibungen im Data Dictionary
4. Benutzerschichten
5. Integritätssicherung: Korrektheit des Datenbankinhalts
6. Datenschutz: Ausschluss unautorisierter Zugriffe
7. Transaktionen: mehrere DB-Operationen als Funktionseinheit (Mehrere Operationen gleichzeitig durchführen)
8. Synchronisationen: parallele Transaktionen koordinieren
9. Datensicherung: Wiederherstellung von Daten nach Systemfehlern

1.3 Einsatzgebiete

1.4 Historisches

2 Relationale Datenbanken

2.1 Relationen für tabellarische Daten

Man kann sich eine Datenbank als eine Menge von Tabellen vorstellen. Jede Tabelle besitzt ein Schlüssel, dieser ist meist in die erste Spalte der Tabelle und muss nicht unbedingt ein Integer sein. Ein Schlüssel kann auch eine Attribut Kombination sein. Das Attribut, welches als Schlüssel dient wird oft durch Unterstreichen gekennzeichnet. Eine Kopie der Schlüssel kann in anderen Tabellen vorkommen, als Möglichkeit die Zwei Tabellen mit einander zu vernetzen und nach bestimmten Eigenschaften zu suchen.

2.2 SQL-Datendefinition

SQL kann verschiedene Datentypen einlesen. Unter anderem:

integer (oder auch integer4, int)

smallint (oder auch integer2)

float(p) (oder auch kurz float)

decimal(p, q) und **numeric(p, q)** mit jeweils q Nachkommastellen

character(n) (oder kurz char(n), bei $n = 1$ auch char) für Strings fester Länge n

character varying(n) (oder kurz varchar(n) für Strings variabler Länge bis zur Maximallänge n

bit(n) oder **bit varying(n)** analog für Bitfolgen

date oder **time** oder **timestamp** für Datums-, Zeit- und kombinierte Datums-Zeit-Angaben

Man kann in SQL vorbestimmen ob Einträge bestimmte Eigenschaften erfüllen damit sie eingespeichert werden können. Ein Pflichtkriterium ist, jede Tabelle braucht ein Primary-Key. Dieser ist nie Null und muss von allen anderen Verschieden sein. Oft werden Integer verwendet, welche hoch zählen, aber es kommt auch vor, dass Strings verwendet werden. Ähnlich wie die Primary-Keys kann man eine Spalte designieren, die Primary-Keys einer anderen Tabelle mit einzuspeichern. Man kann auch zu der Art des Datentypes noch weitere Argumente angeben, wie "not null" was ebenfalls einen Fehler geben würde, wenn das Feld beim Eintragen leer gelassen werden würde.

Man kann ebenfalls ein "default"Wert angeben, welcher Angenommen wird, wenn man das Feld leer lässt. Man kann eine check-Klausel verwenden, um zu sehen ob der eingegebene Wert bestimmte Eigenschaften erfüllt. Weiterhin gibt es eine "create domain"Anweisung (keine ahnung was das macht hlp pls)

2.3 Grundoperationen: Die Relationenalgebra

Mit Basisoperationen kann man neue Ergebnistabellen erzeugen, welche Datenbanktabellen nehmen und an bestimmten Attributen mit einander vereinigen. Die Operationen kommen aus der Relationenalgebra und können beliebig mit einander und in einander verknüpft werden.

- Selektion: In der Relationenalgebra mit einem σ gekennzeichnet. Erlaubt eine Selektion basierend auf Eigenschaften die man im σ spezifiziert. (so was wie aus spalte X möchte ich alle Einträge mit einem Wert größer als 2 welche sich in der Tabelle Y befinden wäre $\sigma_{X>2}(Y)$)
- Projektion: In der Relationenalgebra mit einem π gekennzeichnet. Hier Kann man sich ganze Spalten ausgeben lassen, wo keine Duplikate vorkommen. (Ein Beispiel hierfür wäre aus Tabelle X möchte ich Spalte Y $\pi_Y(X)$)
- Verbund: In der Relationenalgebra mit \bowtie oder \Join bezeichnet. Hiermit werden zwei Tabellen anhand gemeinsamer Attribute in Spalten mit der selben Bezeichnung vereint/verschmilzt (Ein Beispiel hierfür wäre wenn man die Tabellen X und Y Verbinden wollte $X \Join Y$)
- Umbenennung:

2.4 SQL als Anfragesprache

2.5 Änderungsoperationen in SQL

3 Entity-Relationship-Modell

4 Datenbankentwurf