

SQL

Generell

- Undokumentierte Features werden grundsätzlich nicht eingesetzt.
- Datenbanken sollten das Charset UTF-8 verwenden.

Sicherheit

- Grundsätzlich erhält jede Datenbank einen eigenen Login (Benutzername/Passwort) zur Kommunikation mit Webapplikationen.
- Der Login für eine Datenbank besitzt keine Administratoren-Rechte.

Tabellen

- Bei Tabellennamen ist der Plural vorzuziehen. Zum einen können damit Konflikte mit SQL-Keywords vermieden werden, und zum anderen entspricht dies dem Sinn einer Datenbank als Informationsträger besser.
- Tabellennamen werden deutsch oder englisch geschrieben, wobei englisch vorzuziehen ist. Im Zweifelsfall aber lieber die korrekte deutsche Bezeichnung als eine missverständliche oder falsche in einer anderen Sprache.
- Tabellennamen sind generell klein zu schreiben.
- Tabellen sollten nach Möglichkeit einen Präfix erhalten, damit sie eindeutig einem Applikationsteil zugeordnet und problemlos in eine bereits existierende Datenbank übernommen werden können. Trennzeichen für den Präfix ist _
Beispiel: blog_, wiki_, tk_
- Join-Tabellen erhalten den Suffix _join und sind nach dem Muster tabelle1_tabelle2_join aufgebaut.
- Join-Tabellen enthalten einen Primärschlüssel.
- Jede Tabelle enthält einen Primärschlüssel.

Spalten

- Für Primärschlüssel (Primary Keys) sollen grundsätzlich als UUIDs (Strings) verwendet werden. Damit wird die Portabilität der Datenbank erhöht. Bei reinen Jointabellen kann der Primärschlüssel auch ein numerischer Autowert sein.
- Spaltenname sollten lowercase geschrieben werden (Beispiel: summary nicht Summary)
- Die Spalte 'id' als Primärschlüssel in der Haupttabelle erhält keinen Präfix des Tabellennamens (Bsp: nur id und nicht userid).
- In Jointabellen wird der Fremdschlüssel mit nach dem Muster 'tabellenname_id' definiert.
- Folgende Spalten sollten in Datentabellen vorkommen:

Spalte	Typ	Pflicht	Bemerkung
id	varchar, uid	*	Primary
dte_new	datetime	*	
dte_change	datetime	*	
dte_delete	datetime	*	

Abfragen / Queries

- SQL-Keywords werden versal geschrieben (Bsp: SELECT, INSERT, WHERE, ORDER usw).
- Abfragen mit Wildcards (*,%) sind in den seltensten Fällen optimal. Der bekannt SELECT * ist generell zu vermeiden.
- Enthält eine Abfrage mehrere Statements, ist jedes einzelne Statement mit einem **Semikolon ;** abzuschliessen.
- SQL-Statements werden leserlich formuliert:

Beispiel 1: Einfacher Select

```
SELECT  spalte1, spalte2
FROM    tabelle
WHERE   spalte3 = 10
        AND spalte4 = 'Haus'
ORDER  BY spalte5
```

- Beispiel 2: Select mit Join-Tabellen

```
SELECT  tab1.spalte1, tab1.spalte2,
        tab2.spalte1, tab2.spalte2
FROM    tabelle1 tab1
        INNER JOIN tabelle2 tab2
        ON tab1.spalteid = tab2.spalteid
WHERE   tab1.spalte3 = 10
        AND tab2.spalte3 = 'Haus'
ORDER  BY tab1.spalte4
```

Oder

```
SELECT  tabelle1.spalte1, tabelle1.spalte2,
        tabelle2.spalte1, tabelle2.spalte2,
FROM    tabelle1
        INNER JOIN tabelle2
        ON tabelle1 .spalteid = tabelle2.spalteid
WHERE   tabelle1.spalte3 = 10
        AND tabelle2.spalte3 = 'Haus'
ORDER  BY tabelle1.spalte4
```

Verknüpfungsanweisungen und logische Operatoren werden dabei auf de

...

```
INNER JOIN tabelle2
        ON tabelle1.spalteid = tabelle2.spalteid
```

...

```
AND tabelle2.spalte3 = 'Haus'
```

Verknüpfungsanweisungen und logische Operatoren werden dabei auf der gleichen Zeile geschrieben, wie die effektive Anweisungen:

```
...  
INNER JOIN tabelle2  
    ON tabelle1.spalteid = tabelle2.spalteid  
...  
AND tabelle2.spalte3 = 'Haus'
```

- Aliase für Tabellennamen sind zulässig, sollten aber zur besseren Lesbarkeit selten verwendet werden.

Indexe

- Indexe sind bewusst einzusetzen. Sie können Abfragen massiv beschleunigen, aber nur wenn sie korrekt und vor gezielt eingesetzt werden.
- Indexe lohnen sich vor allem dann, wenn die indizierten Spalten häufig in folgenden SQL-Klauseln vorkommen: WHERE, ORDER BY, GROUP BY, TOP und DISTINCT.
- In diesen Situationen ist die Erstellung eines Indexs selten ratsam:
 - Kleine Tabellen benötigen in der Regel keinen Index.
 - Spalten mit grossen Datenmengen sind unbedingt geeignet, beispielsweise bei TEXT-Spalten
 - Werden die Daten einer Tabelle häufig mutiert und selten gelesen, ist die Aktualisierung des Indexes aufwändiger als der Gewinn durch schnellere Abfragen wett machen könnte.
 - Wird bei der Analyse des Execution Plan festgestellt, dass ein vorhandener Index nicht verwendet worden ist, sollte man ihn auch nicht weiter einsetzen und pflegen.

Kommentare

- Benutze Kommentare in Store Procedures, Funktionen und anderen SQL-Scripts.

Tipps

SQL allgemein

- [Doppelte Datensätze aus der Datenbank löschen](#)
- [Mehrere Inserts bei MSSQL, MySQL und Postgre](#)
- [History für Datenbankeinträge automatisieren: Trigger benutzen](#)
- [INNER JOIN und WHERE-Attribute](#)
- [Null-Werte bei SELECT abfangen](#)
- [Query mit Like und Spezialzeichen](#)

MSSQL

- [mit Query-Resultaten gleich weiterarbeiten](#)
- [Insert und Update im gleichen SQL-Query](#)
- [Arbeiten mit Tabellenvariablen](#)
- [Transaction-Logs analysieren](#)
- [Tabellen per SQL anpassen](#)
- [SQL und unlöschbare Datensätze](#)
- [Optimizing SQL Server Query Performance](#)

MySQL

- [mehrere SQL-Statements im gleichen Query](#)
- [LIMIT bei MySQL Abfragen](#)
- [RegEx mit MySQL](#)

Postgre

- [Vorsicht mit Gross- und Kleinschreibung](#)
- [15 PostgreSQL Skripte für Programmierer](#)
- [15 PostgreSQL Skripte für Administratoren](#)
- [Backup mit pg_dump](#)

externe Quellen

Dokumentationen

- [MSSQL Online-Dokumentation \(deutsch\)](#)
- [MySQL Online-Dokumentation \(deutsch\)](#)
- [PostgreSQL Online-Dokumentation \(englisch\)](#)
- [SQL Reserved Words Checker](#)

CheatSheets

- [MS-SQL CheatSheet](#)
- [MS-SQL Server CheatSheets](#)
- [MySQL CheatSheet](#)
- [PostgreSQL CheatSheet / Refcard](#)
- [PostgreSQL CheatSheet](#)
- [PostgreSQL CheatSheet 2](#)

Code-Formatierung

- [SQLinForm](#)
- [SQL and PL/SQL Formatter](#)

Datenbank-Konvertierung

nach PostgreSQL

- [MySQL nach PostgreSQL](#)
- [MS Access, MSSQL, MySQL, Oracle, DBase, Filemaker, DB2, Interbase nach PostgreSQL](#)

Code-Beispiele

- [Zufälliger Datensatz auswählen, verschiedene Datenbanken \(en\)](#)
- [PostgreSQL-Journal \(en\)](#)