JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

# Data - Dictionary

**Version 1.0**

**Note:**

**Betreuer: Prof. Borko**

**Begonnen am 4. Mai 2016**

**Beendet am 11. Mai 2016**

# Inhaltsverzeichnis

# 1 Einführung

Erarbeiten Sie in 2er-Gruppen folgende Fragestellungen in einem Dokument und geben Sie dieses als PDF ab. Beachten Sie dabei die Zitierregeln!

1. Wo liegt der große Unterschied zwischen den Data Dictionaries der einzelnen DBMS und wie erfolgt der Zugriff (MySQL, PostgreSQL, Oracle)?
2. Kann eine Performancesteigerung durch Manipulation am System Catalog erzielt werden?
3. Wie könnte man über den System Catalog den Datentypen eines Attributes einer bestimmten Tabelle ändern? Tun Sie dies und erläutern Sie was dabei nach einem SELECT auf dieses Attribut passiert!
4. Wo liegt der Unterschied der einzelnen Index-Typen von PostgreSQL? Listen Sie diese tabellarisch auf!
5. Wie sind B-Bäume grundsätzlich aufgebaut?
6. Wieso kann bei B-Bäumen stets die maximale Zugriffszeit berechnet werden und in welchen Zusammenhang steht die Ordnungszahl mit den Knoten?
7. Wie verläuft die Suche bei B-Bäumen?
8. Welche weitere Bäume werden bei Datenbank-Managementsystemen verwendet? Erläutern Sie kurz die Unterschiede!

Die Fragen 1-4 wurden von Patrick Wichert ausgearbeitet und die 5-8 von Christian Gabrail.

# 2 Fragen

## 2.1 Unterschied zwischen den Data Dictionaries

**Wo liegt der große Unterschied zwischen den Data Dictionaries der einzelnen DBMS und wie erfolgt der Zugriff (MySQL, PostgreSQL, Oracle)?**

- MySQL

  Das Information_schema gibt jeden einen Zugriff auf die Metadaten. Das sind Daten wie zum Beispiel Name der Datenbank, Datentyp der Spalten usw. Die Datenbank Information_schema speichert die Informationen der schon vorgelegten bzw. selbst erstellten Datenbanken. Die Datenbank information_schema kann mit einem use Befehl verwendet werden. Es können aber nur Select-Befehle ausgeführt werden und keine Insert, Update oder Delete-Befehle, da diese nur ein View ist.

```
mysql> SELECT table_name, table_type, engine
    -> FROM information_schema.tables
    -> WHERE table_schema = 'db5'
    -> ORDER BY table_name;
+------------+------------+--------+
| table_name | table_type | engine |
+------------+------------+--------+
| fk         | BASE TABLE | InnoDB |
| fk2        | BASE TABLE | InnoDB |
| goto       | BASE TABLE | MyISAM |
| into       | BASE TABLE | MyISAM |
| k          | BASE TABLE | MyISAM |
| kurs       | BASE TABLE | MyISAM |
| loop       | BASE TABLE | MyISAM |
| pk         | BASE TABLE | InnoDB |
| t          | BASE TABLE | MyISAM |
| t2         | BASE TABLE | MyISAM |
| t3         | BASE TABLE | MyISAM |
| t7         | BASE TABLE | MyISAM |
| tables     | BASE TABLE | MyISAM |
| v          | VIEW       | NULL   |
| v2         | VIEW       | NULL   |
| v3         | VIEW       | NULL   |
| v56        | VIEW       | NULL   |
+------------+------------+--------+
```

- PostgreSQL

  Bei PostgreSQL werden die Informationen über die Daten in eigene Tabellen gespeichert. Diese Tabellen starten mit pg_. Diese Tabellen kann man überarbeiten bzw. etwas hinzufügen, nur die Gefahr besteht  das ganze System und alle Daten zu zerstören.

**Table of Contents**

- Oracle

  Eines der wichtigsten Teile der Oracle Datenbank ist, dass sie nur ausgelesen werden kann um die Informationen der Datenbanken bzw. Tabellen lesen zu können.
  Sachen die in der Oracle data dictionary beinhaltet sind:
  - Die Definition aller Schemen Objekte in der Datenbank (Tabellen, Views, indexes, usw.)
  - Wie viel Platz die Schemen Objekte einnehmen
  - Default Werte für Spalten
  - Der Name des Oracle Users
  - Die Rechte die jedem User zugeteilt wurden
  - Informationen wie z.B. wer darf eine Tabelle Updaten bei gewissen Schemen Objekten
  - Zusätzliche Datenbank Informationen

  Die Daten sind in Tabellen und Views gespeichert und sind alle in der Datenbank SYSTEM gespeichert.

### 2.2 Performancesteigerung

**Kann eine Performancesteigerung durch Manipulation am System Catalog erzielt werden?**

Ja es kann dadurch eine Performancesteigerung erzielt werden. Dieses sollte man nur machen wenn man sich wirklich sicher ist was man macht, da die ganze Datenbank dadurch zerstört werden könnte und alle Daten die in der Datenbank enthalten sind auch zerstört sind. Falls man sich dazu doch entscheiden sollte, kann man davor ein Backup der Datenbanken machen und diese bei Fehlern wieder neu aufspielen.

### 2.3 System Catalog

**Wie könnte man über den System Catalog den Datentypen eines Attributes einer bestimmten Tabelle ändern? Tun Sie dies und erläutern Sie was dabei nach einem SELECT auf dieses Attribut passiert!**

Die Informationen der Datentypen sind in der Tabelle pg_attribute gespeichert. Wenn man bei dieser Tabelle ein Update macht, kann man die Datentypen ändern und so ändert sich das Ganze auch für alle anderen Tabellen die diese Datentypen verwenden.

## 2.4 Unterschied der einzelnen Index-Typen

**Wo liegt der Unterschied der einzelnen Index-Typen von PostgreSQL? Listen Sie diese tabellarisch auf!**

Jeder von diesen Index-Typen benutzt verschiedene Algorithmen zur Findung der besten Sortierung. Bei einer Default Eingabe des Create Index Befehles wird immer ein B-Baum gewählt (welche die häufigsten Situationen löst).

B-Bäume haben 5 verschiedene such bzw. Ordnungsfaktoren (siehe Tabelle unterhalb). Diese können einfache Abfragen oder Ordnungen wie z.B. Liegt etwas zwischen 1-20 oder sortiere vom größten zum kleinsten.

Hash benutzt einen einfachen = Befehl. Man sieht nur nach ob der Wert gleich dem anderen ist und ist somit der einfachste Index-Typ aber gleichzeitig auch der, der am wenigsten macht.

GiST Index-Typen sind nicht nur eine einzelne Art von Indizes sondern eine Infrastruktur, welche viele verschiedene Index Strategien implementiert. Die Operatoren von GiST sind immer unterschiedlich, je nachdem welche Index Strategie benutzt wird (in der operator class). In PostgreSQL werden als Standard die unten angegebenen Operatoren verwendet.

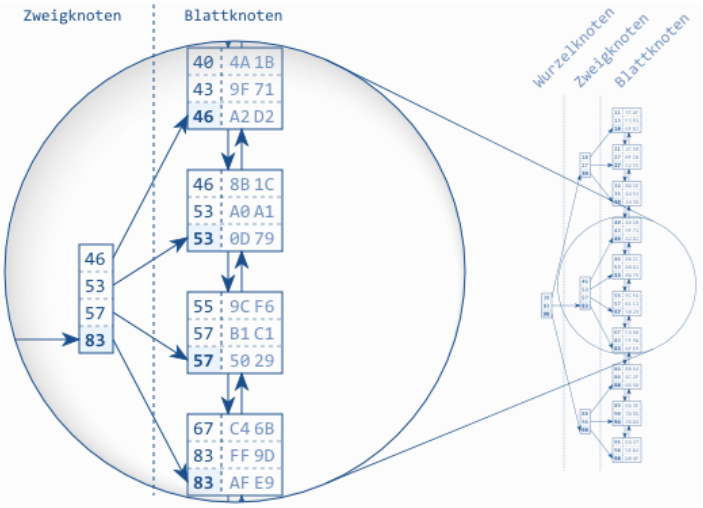
GIN Indizes sind inverse Indizes welche Werte verarbeiten kann die mehr als nur einen Key haben z.B. Arrays. Genauso wie GiST, kann auch GIN verschiedene Operatoren verwenden. Es muss wie bei GiST davor angegeben werden welche Index Strategie verwendet werden soll. Bei PostgreSQL werden die unten angegebenen Operatoren als Standard verwendet.

| Index Typen | Operationen |
|---|---|
| B-Bäume | <, <=, =, >=, > |
| Hash | = |
| GiST | <<, &<, &>, >>, <<\|, &<\|, \|>&, \|>>, @>, @<, ~=, && |
| GIN | <@, @>, =, && |

## 2.5 Aufbau der B-Bäume

**Wie sind B-Bäume grundsätzlich aufgebaut?**

Die Abbilung zeigt einen Index mit 30 Einträgen.
Die Reihenfolge der Leaf-Nodes(zu deutsch: Blattknoten) wird durch eine doppelte Verkettung hergestellt. Root- und Branch-Nodes dienen der Suche nach bestimmten Einträgen.




Die Abbildung hebt hervor, wie ein Zweigknoten auf die darunter liegenden Blattknoten verweist. Dabei entspricht jeder Eintrag im Zweigknoten dem größten Wert im Blattknoten. Da der größte Wert im ersten Blattknoten 46 ist, ist der erste Eintrag im Zweigknoten ebenfalls 46. Dasselbe gilt für alle weiteren Blattknoten, sodass der Zweigknoten letztendlich die Einträge 46, 53, 57 und 83 enthält. Nach diesem Schema wird eine Verzweigungsschicht aufgebaut, bis alle Blattknoten von einem Zweigknoten erfasst sind.

Die nächste Schicht ist ebenso aufgebaut, nur dass dabei auf die Zweigknoten verwiesen wird. Das Ganze wiederholt sich so lange, bis eine Schicht entsteht, die nur aus einem Knoten besteht – dem Wurzelknoten oder *Root-Node*. Das Besondere an dieser Baumstruktur ist, dass sie eine einheitliche Tiefe hat: Der Weg zwischen dem Wurzelknoten und den Blattknoten ist überall gleich lang.

## 2.6 Maximale Zugriffszeit

**Wieso kann bei B-Bäumen stets die maximale Zugriffszeit berechnet werden und in welchen Zusammenhang steht die Ordnungszahl mit den Knoten?**

Ein B-Baum mit der Anzahl von, zum Beispiel 20 Ebenen, würde eine Zugriffszeit von 20 aufweisen. Dabei muss berücksichtigt werden das die Ordnungszahl die Tiefe des Baumes bestimmt und somit, sollte man einen Knoten hinzufügen so würde die Ordnungszahl um einen konstanten Wert von 1 aufsteigen.

*Man schliesst daraus das anhand der Anzahl der Ebenen man auf die Zugriffszeit schliessen kann!*

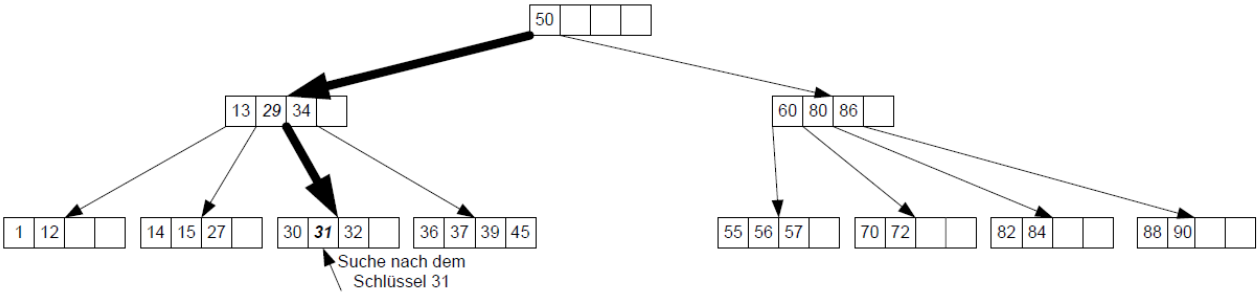
Jede Ebene hat einen gewissen Speicherraum von Knoten welche er in einer Ebene verstauen kann. Sollte dieser überschritten werden, dann müsste man die überschüssigen Knoten in einer untergelegten Ebene verfrachten.

## 2.7 Suche in B-Bäumen

**Wie verläuft die Suche bei B-Bäumen?**

Das Suchen in einem B-Baum ist dem Vorgehen in einem binären Suchbaum ähnlich. Man hat beim Binärbaum aber nur je zwei Auswahlmöglichkeiten,
beim B-Baum je nach Grad entsprechend mehr; weiterhin muss man beim B-Baum auch die Knoten durchsuchen.

Will man die Zahl 31 suchen, so weiß man, dass sie im linken Unterbaum (auf den der linke Verweis des Schlüssels 50 zeigt, da 31 < 50) zu finden sein wird. Man darf nicht vernachlässigen, dass eigentlich auch der Wurzelknoten Eintrag für Eintrag durchsucht werden müsste; hier erübrigt sich das, da nur ein Schlüssel vorhanden ist.
Den Knoten auf Ebene 1 durchsucht man Schlüssel für Schlüssel, den gewünschten Eintrag wird man dort nicht finden, allerdings muss man nach der 29 und noch vor der 34 eine Stufe hinab steigen (dem entsprechenden Pointer folgen). Auf Ebene 2 durchsucht man schließlich den entsprechen den Knoten von links nach rechts und wird beim zweiten Element fündig.

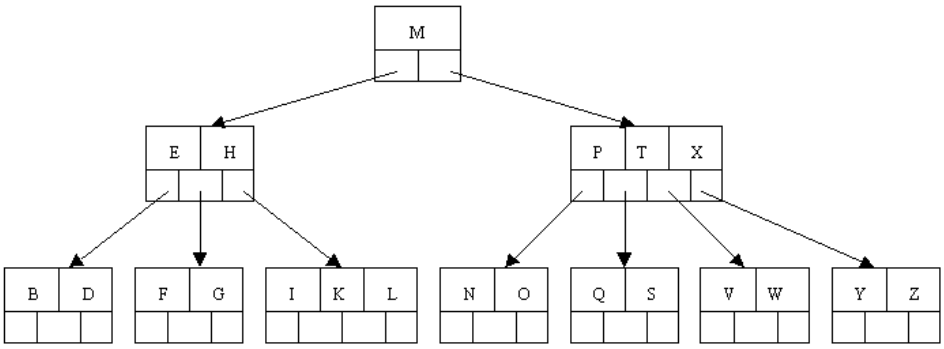


## 2.8 Bäume

**Welche weitere Bäume werden bei Datenbank-Managementsystemen verwendet? Erläutern Sie kurz die Unterschiede!**

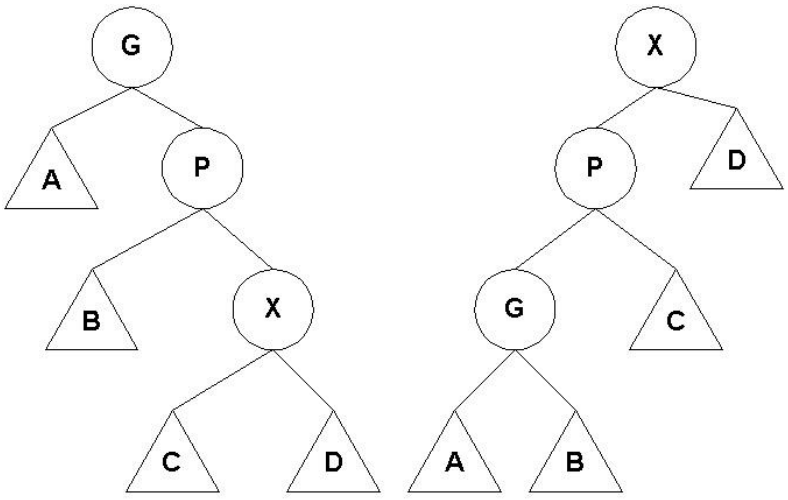
### B-Tree

Ein binärer Baum beschreibt das Verfahren zum Plazieren und Lokalisieren von Dateien in einer Datenbank , insbesondere dann, wenn alle Daten bekannt sind, und sie sich direkt im RAM befinden. Der Algorithmus findet Daten , indem er wiederholt die Anzahl der letztendlich zugängliche Aufzeichnungen halbiert bis nur noch eins überbleibt.



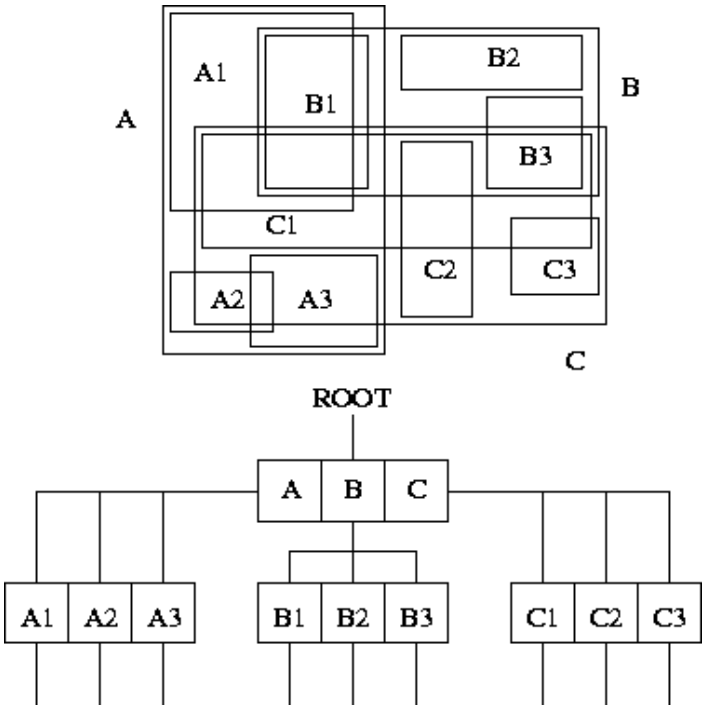

### Splay-Tree

Ein Splay-Baum (auch Spreizbaum) ist ein selbst anpassender Suchalgorithmus für die Platzierung und Auffindung von Dateien in einer Datenbank. Die angefragten Elemente werden in die Nähe der Wurzel „gespült“, so dass sie bei einer alsbaldigen erneuten Suche schneller gefunden werden.



**R-Tree**

R- Bäume sind Baumdatenstrukturen welche für räumliche Zugriffsverfahren verwendet werden , das heißt , für die Indizierung multidimensionale Informationen wie geographische Koordinaten , Rechtecke oder Polygone .



**Unterschiedliche Vorteile eines B+ trees und B trees:**

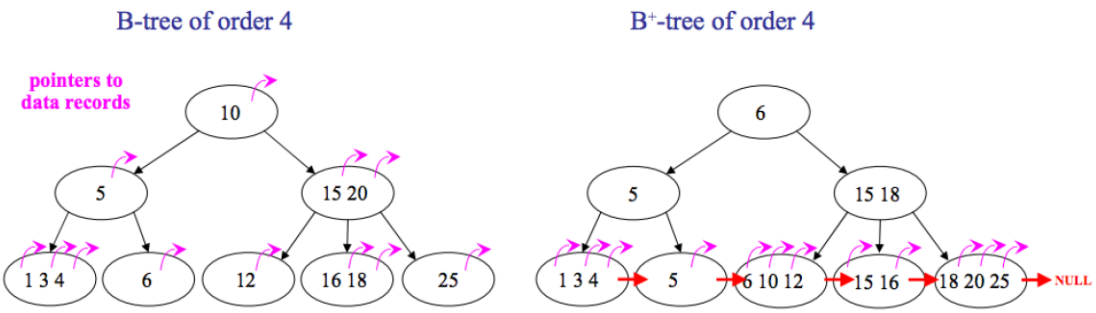
Vorteile des B+ trees:

- Da B+ Bäume keine Daten mit den inneren Nodes verbunden sind, passen mehrere Keys auf einer Seite des Speichers. Daher sind weniger Cache-Fehlzugriffen erforderlich um auf die Daten zuzugreifen, die auf einem Leaf Node sind.
- Die Leaf Nodes eines B+ Trees sind miteinander verbunden. Das heißt ein vollständiger Scan aller Obejkt in dem Baum kann durch einen einzigen linearen Durchgang erfolgen. Ein B tree müsste hingegen jedes Level des Baums durchsuchen!

Vorteile des B trees:

Da B Büme mit jedem Key Daten beinhalten, können auf häufig benutzten Keys schneller zugegriffen werden da sie näher zum Root liegen.

- A B⁺-tree can be viewed as a B-tree in which each node contains only keys (not pairs), and to which an additional level is added at the bottom with linked leaves



# 3 Quellen

| INDEX | Information |
|---|---|
| [1] | **Wikipedia:** https://de.wikipedia.org/wiki/Data-Dictionary<br>**Kapitel:** 2<br>**zuletzt abgerufen am [10.05.2016]** |
| [2] | **Link:** http://use-the-index-luke.com/de/sql/anatomie/index-baum<br>**zuletzt abgerufen am [10.05.2016]** |
| [3] | **Link:** http://use-the-index-luke.com/sql/anatomy/the-tree<br>**zuletzt abgerufen am [10.05.2016]** |
| [4] | **PDF:** https://www.informatik-forum.at/attachment.php?attachmentid=948<br>**zuletzt abgerufen am [10.05.2016]** |
| [5] | **Link:** http://searchsqlserver.techtarget.com/definition/B-tree<br>**zuletzt abgerufen am [10.05.2016]** |
| [6] | **Link:** http://searchoracle.techtarget.com/definition/splay-tree<br>**zuletzt abgerufen am [10.05.2016]** |
| [7] | **Link:** https://docs.oracle.com/html/A96524_01/c05dicti.htm<br>**zuletzt abgerufen am [10.05.2016]** |
| [8] | **Link:** http://www.postgresql.org/docs/9.3/static/catalog-pg-attribute.html<br>**zuletzt abgerufen am [10.05.2016]** |
| [9] | **Link:** http://www.postgresql.org/docs/9.0/static/indexes-types.html<br>**zuletzt abgerufen am [10.05.2016]** |
| [10] | **PDF:** *Oracle, mysql-infoschema-excerpt-5.1, S.1f*<br>**zuletzt abgerufen am [10.05.2016]** |
| [11] | **PDF:** *postgresql-9.4 manual, S. 138 /5.7.5*<br>**zuletzt abgerufen am [10.05.2016]** |

# 4 Aufwandsabschätzung

| Arbeit | Stundenabschätzung | Gruppenmitglied |
|---|---|---|
| **Formatierung** | 40 min | Gabrail |
| **Aufgaben 1-4** | Ca. 1:30 h | Wichert |
| **Aufgaben 5-8** | 4:30 h | Gabrail |