Kapitel 7

Datenmodellierung

Stefan Keller

Danke an Dr. Andreas Neumann

Ueberblick

- Was ist Datenmodellierung?
- Query-Performance-Optimierung: Tipps

Was ist Datenmodellierung?

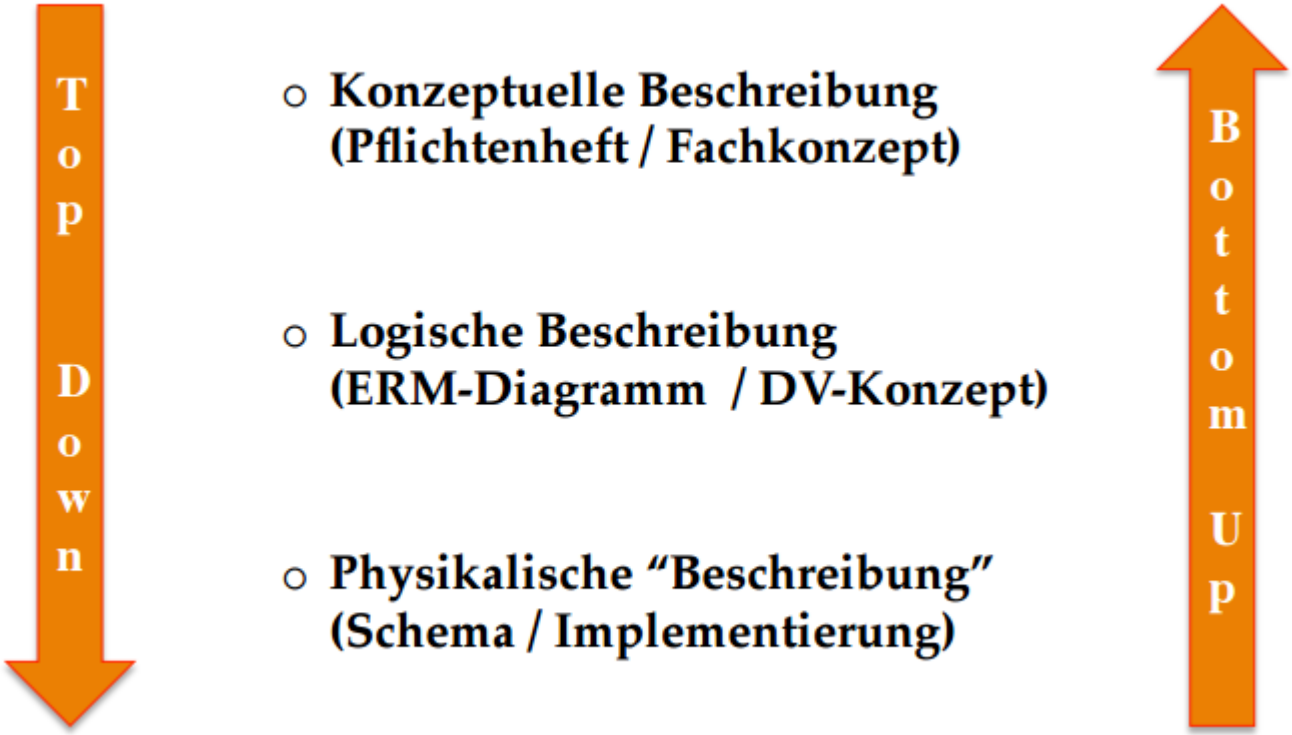
- Datentechnische Repräsentation realer Objekte, ihrer Eigenschaften und Beziehungen → am Besten Datenbankspezialist zusammen mit Fachperson(en)
- Nur Daten, welche zum fachlich-inhaltlichen Zweck der Systeme gehören, werden einbezogen (z.B. keine Konfigurationsparameter welche technisch benötigt werden)
- Verschiedene Stufen der Datenmodellierung
  - Konzeptuelles Schema (Skizze, verbale Beschriebe, Mind map)
  - Logisches Datenbankschema (ER-Diagramm)
  - Physisches Datenbankschema (SQL DDL Statements)

Nutzen von Datenmodellierung?

- Unterstützung der Kommunikation zwischen Entwicklern, Fachexperten und Nutzern
- Hilfe für Analysten um eine Domäne zu verstehen
- Liefern von Input für den Systemdesignprozess
- Dokumentieren die (ursprünglichen) Anforderungen und stehen damit als Referenz zur Verfügung

- Falls kein Datenmodell erstellt wurde kann ein Re-Engineering erfolgen (einlesen bestehender SQL-Datenstrukturen und Umsetzung als ER-Diagramm)

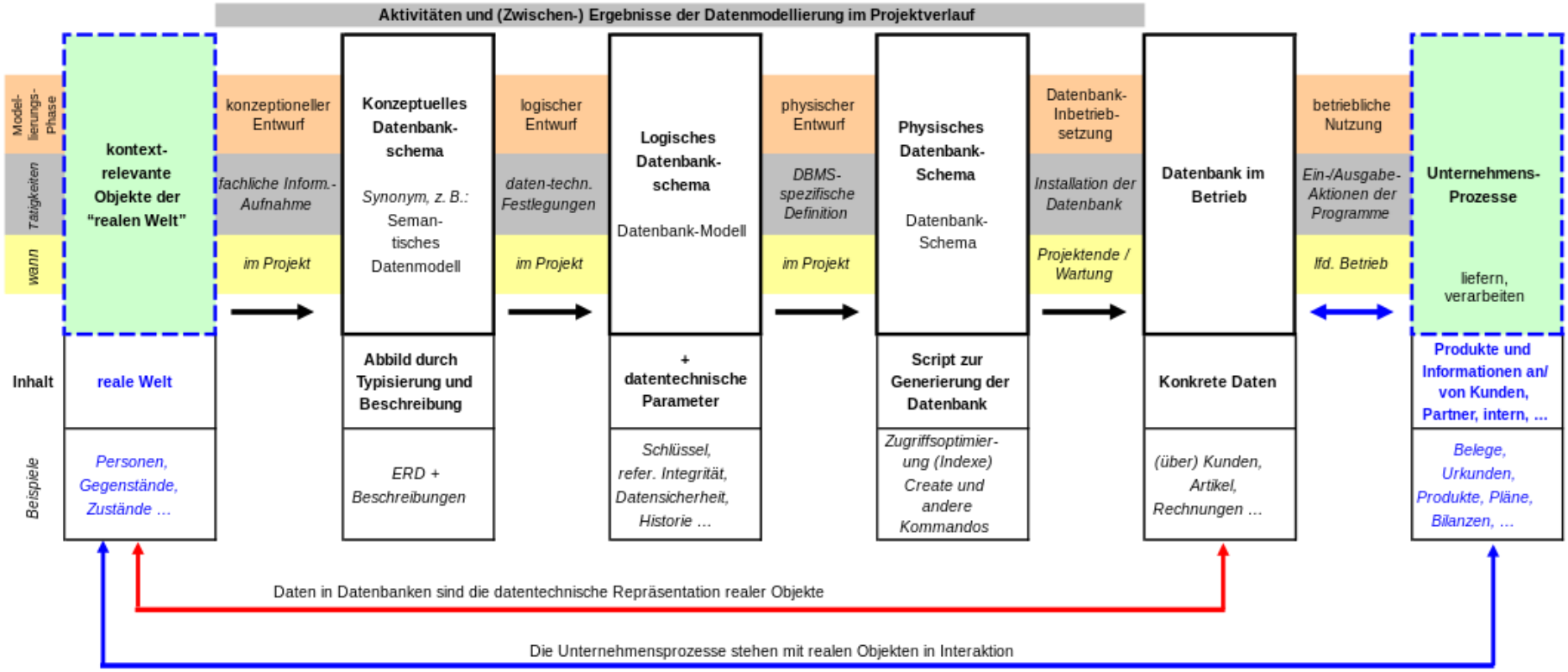
Stufen der Datenmodellierung



Oft ein iterativer Prozess (Verbesserung des Modells)

Stufen der Datenmodellierung

Datenmodellieren: Entwicklung von der fachlichen, implementierungsunabhängigen Konzeption bis zur Datenbank



Konzeptuelles Schema

- Skizzen, Brainstorming und verbale Beschriebe
- Identifizieren des relevanten Informationsbedarfs (Attribute)
- Identifizieren von Entitätstypen und Beziehungstypen
- Zuordnen der Attribute zu Entitätstypen
- Festlegen möglicher Attributwerte, Vorschläge für identifizierende Attribute
- Bestimmen der Beziehungskardinalität: 1:1, 1:n, n:1, n:m
- Fachliches Beschreiben der Entitäts- und Beziehungstypen und der Attribute

Quelle: https://de.wikipedia.org/wiki/Datenmodellierung

Logisches Schema

- Z.B. ER-Diagramm
- Methodisches Überprüfen der fachlich modellierten Ansätze (z. B. durch Normalisierung)
- Gegebenenfalls Bilden neuer Entitätstypen, z. B. durch Spezialisierung / Generalisierung
- Überführen des ER-Modells in ein Relationenmodell
- Festlegen der identifizierenden Schlüssel (Primärschlüssel)
- Festlegungen zur technischen Umsetzung von Beziehungen: Fremdschlüssel, Beziehungstabellen (Zwischentabellen bei n:m)

Quelle: https://de.wikipedia.org/wiki/Datenmodellierung

Logisches Schema

- Festlegungen zur referentiellen Integrität bei Fremdschlüsselbeziehungen
- Erweitern des Datenbankmodells im Zusammenhang mit Historien- und Versionsführung, Mandantenfähigkeit etc.
- Ergänzen des Modells um Lookup-Tabellen, Parametertabellen etc.
- Wahl einer geeigneten Datenbank (RDBMS)

Quelle: https://de.wikipedia.org/wiki/Datenmodellierung

Physisches Schema

- Formulieren der SQL-Scripte / DDL Kommandos zum Einrichten und Konfigurieren der Datenbank (in der Syntax des DBMS)
- Optimierungsmöglichkeiten für Datenzugriffe (z.B. durch Indexe) einstellen
- Festlegungen zur Datensicherung

Quelle: https://de.wikipedia.org/wiki/Datenmodellierung

Foreign Key Constraints

- Referential Actions (was passiert bei Updates und Deletes) bei der referenzierenden Tabelle?
  - CASCADE
  - RESTRICT
  - NO ACTION
  - SET DEFAULT , SET NULL
  - Triggers
- Referential actions werden separat für DELETE und UPDATE definiert
- Es können nur PRIMARY KEY und UNIQUE Spalten referenziert werden

Foreign Key Constraints

- CASCADE:
  - Wenn Einträge in Elterntabelle gelöscht oder aktualisiert werden, werden diese auch in Kindtabelle (referenzierende Tabelle) gelöscht oder aktualisiert
- SET DEFAULT:
  - Ein Default Wert wird gesetzt
  - Der Default Wert muss in der referenzierten Tabelle vorhanden sein
- SET NULL:
  - Nicht so sinnvoll → Datenleichen?
  - Wenn verwendet, sollte zeitnah wieder ein gültiger Wert gesetzt werden

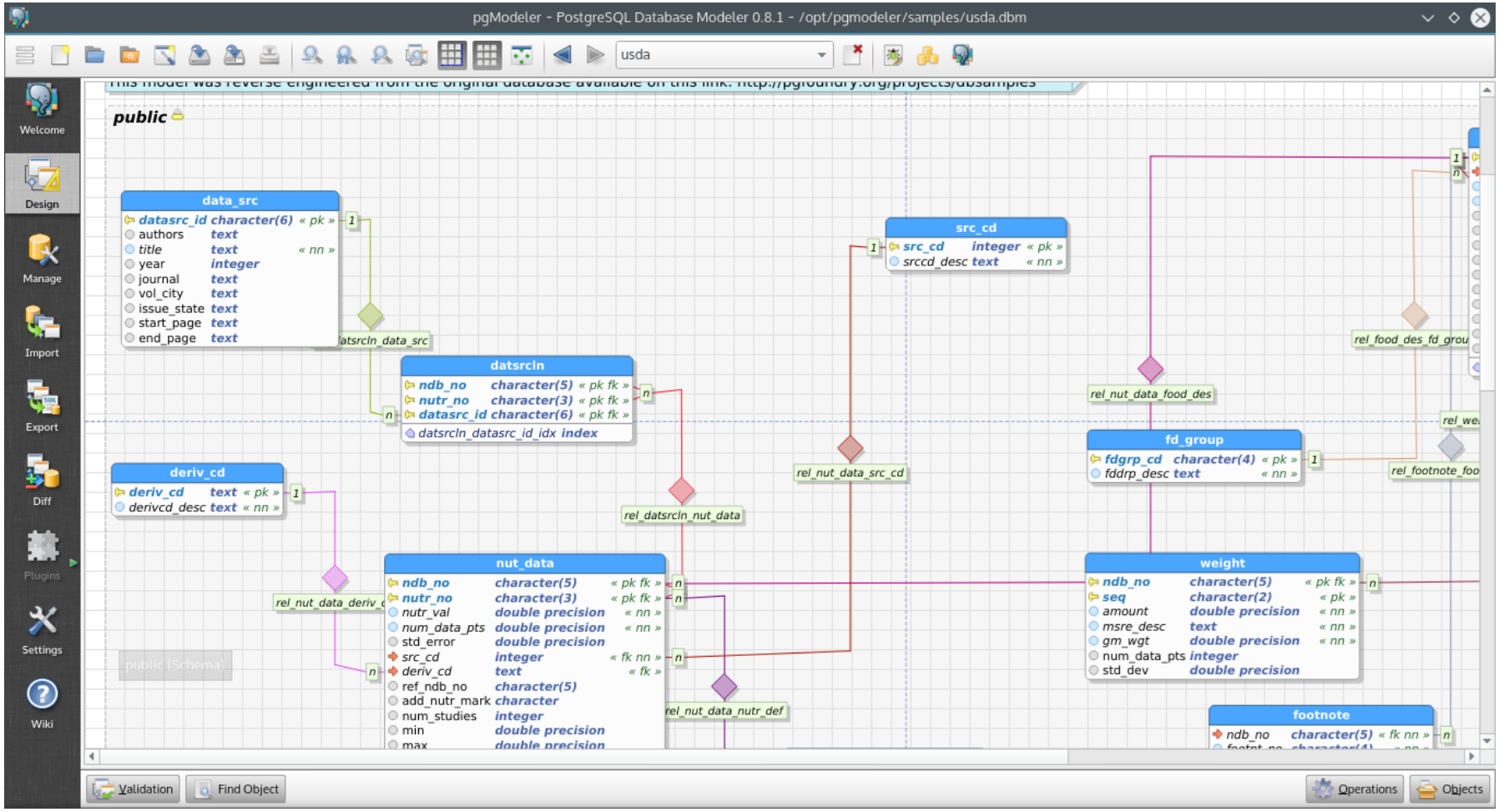
Foreign Key Constraints

- RESTRICT:
  - Solange in der Kindtabelle (referenzierende Tabelle) noch Einträge auf die Elterntabelle zeigen, können die entsprechenden Records in der Elterntabelle nicht gelöscht oder aktualisiert werden
- NO ACTION:
  - Sehr ähnlich zu RESTRICT
  - Unterschied:
    - bei NO ACTION wird Fehlermeldung ausgegeben
    - Der Check bei NO ACTION ist „deferred“ - wird erst am Ende der Transaktion geprüft, RESTRICT verhindert Manipulation von vornherein

Datenmodellierungswerkzeuge

- Liste von DB-Modeler Werkzeugen für PostgreSQL:
- https://wiki.postgresql.org/wiki/GUI_Database_Design_Tools
- PgModeler: http://pgmodeler.com.br/
- UML Editor (für Interlis): http://www.umleditor.org/
- ili2pg (Import INTERLIS-Struktur nach PostGIS): http://www.eisenhutinformatik.ch/interlis/ili2pg/
- pgAdmin4 (einfach): http://www.pgadmin.org/

pgModeler


pgModeler - PostgreSQL Database Modeler 0.8.1 - /opt/pgmodeler/samples/usda.dbm
usda
Welcome
Design
Manage
Import
Export
Diff
Plugins
Settings
Wiki
public
data_src
datasrc_id character(6) « pk »
authors text
title text « nn »
year integer
journal text
vol_city text
issue_state text
start_page text
end_page text
datsrcln
ndb_no character(5) « pk fk »
nutr_no character(3) « pk fk »
datasrc_id character(6) « pk fk »
datsrcln_datasrc_id_idx index
src_cd
src_cd integer « pk »
srccd_desc text « nn »
deriv_cd
deriv_cd text « pk »
derivcd_desc text « nn »
nut_data
ndb_no character(5) « pk fk »
nutr_no character(3) « pk fk »
nutr_val double precision « nn »
num_data_pts double precision « nn »
std_error double precision
src_cd integer « fk nn »
deriv_cd text « fk »
ref_ndb_no character(5)
add_nutr_mark character
num_studies integer
min double precision
max double precision
fd_group
fdgrp_cd character(4) « pk »
fddrp_desc text « nn »
weight
ndb_no character(5) « pk fk »
seq character(2) « pk »
amount double precision « nn »
msre_desc text « nn »
gm_wgt double precision « nn »
num_data_pts integer
std_dev double precision
footnote
ndb_no character(5) « fk nn »
rel_nut_data_food_des
rel_nut_data_src_cd
rel_datsrcln_nut_data
rel_nut_data_nutr_def
rel_food_des_fd_grou
rel_footnote_foo
Validation
Find Object
Operations
Objects

Was ist zu beachten bei logischer/physischer Modellierung?

- Objektnamen immer klein schreiben
- “_“ als Trendzeichen verwenden, z.B. haeuser_stadtverwaltung
- Zusammengehörige Datenbankobjekte in ein separates Schema ablegen
- Primärschlüssel: siehe Folien 03 Datentypen...
- Sinnvolle Datentypen verwenden (Wertebereiche beachten)
- NOT NULL, UNIQUE und Check Constraints verwenden
- Default-Werte definieren
- Foreign Key Constraints (Fremdschlüsselbeziehungen) bei Beziehungen zwischen Tabellen verwenden
- Indexe erstellen auf Geometriespalten und Spalten auf denen häufig gesucht wird

Query-Performance-Optimierung: Tipps

- Query-Ebene
  - Function ST_SimplifyPreserveTopology() vor ST_Intersects
  - Query auf Subselect oder CTE umschreiben
- Daten-Ebene
  - Ist VACUUM nachgeführt?
  - Sind die Daten valid?
  - Daten partitionieren
- Physische Ebene
  - Index erstellen
  - Tabelle clustern (Syntax CLUSTER...)
  - Materialized View erstellen → Challenge des „Refresh“ lösen
  - Mehr Memory und/oder schnellere Disk beschaffen
  - Schema umstellen
  - Auf Grid (Raster) umstellen, Bsp. Hektarraster