

Data Science Cup 2017 – Lies Mich

Datenimport

- Schema.sql-Skript ausführen um alle Relationen und Sichten (Views) in der Datenbank zu erstellen
- Die im .zip-Format vorliegenden CSV-Dateien entpacken
- Die Platzhalter #PATH# in import.sql mit dem Pfad zu den entpackten CSV-Dateien ersetzen
- Das angepasste import.sql-Skript ausführen um die Daten zu importieren
- Hinweis: Die SQL-Skripte sind für das Datenbanksystem PostgreSQL erstellt worden, die Verwendung anderer Datenbanksysteme kann eine Anpassung der Skripte notwendig machen.

Datenbankschema

In Abbildung 1 ist das Datenbankschema der importierten Daten dargestellt. Es dient der Veranschaulichung wie die beiden Faktentabellen *Items* und *supply_and_demand* mit den restlichen Dimensionstabellen in Verbindung stehen, um den Einstieg in die Arbeit mit den Daten zu erleichtern.

Über die dargestellten Sachverhalte hinaus ist folgendes zu beachten:

Items: In dieser Faktentabelle sind die Verkäufe der einzelnen Items abgelegt. *Typeid*, *stationid* und *time* geben an, welches Item in welcher Station an welchem Tag verkauft wurde. *Sales_units* bezeichnet die Summe aller Verkäufe eines Items in einer Station an einem Tag. *Avgprice*, *minprice* und *maxprice* geben den durchschnittlichen, minimalen und maximalen Preis der Verkäufe an. Die Spalte *orders* zeigt aus wie vielen verschiedenen Orders die Verkäufe stammen (Kauf- und Verkaufsgebote deren Restbestand sich in Laufe des Tages verändert hat.).

Supply and demand: Die Informationen in dieser Faktentabelle sind ähnlich zu denen in der Tabelle *Items*. Angebot und Nachfrage werden getrennt voneinander aufgeführt. Die Spalten *supp_bids* und *dem_bids* geben an aus wie vielen Geboten die aufsummierten Angebote und Nachfragen stammen.

Regionshierarchie: Die Hierarchie besteht von klein zu groß geordnet aus *stations* → *solarsystems* → *constellations* → *regions*. Zur vereinfachten Arbeit besitzen die jeweils feingranulareren Hierarchiestufen Verweise zu allen darüber liegenden Stufen (diese Verweise sind in Abbildung 1 der Übersichtlichkeit halber nicht dargestellt). Es existieren Stationen, welche durch Spieler eingenommen werden können, diese sind mit *stationids* gekennzeichnet welche mit 61... beginnen und sind nicht in der Tabelle *stastations* verzeichnet, da für sie keine Metainformationen verfügbar sind.

Patches: Patches werden von Spielehersteller bereitgestellt, um das Spiel zu erweitern oder besser zu balancieren. Dadurch können sich die Eigenschaften verschiedenen Artikel (*types*) verändern. Für jede stattgefunden Änderung gib es einen neuen Eintrag in den Relationen *types*, *typesattributes* und/oder *attributes*.

Types und attributes: Die Relationen *types*, *typesattributes* und *attributes* sind nicht nur über die angegebenen id-Schlüssel verbunden sondern auch über *patch*, *patchversion* und *patchbuild* (ebenfalls der Übersichtlichkeit halber nicht dargestellt).

Aufgabe 1

Die jeweilige Typ-Zeitreihe kann unter Angabe der Typeid mit Hilfe folgender SQL-Query angefragt

```
SELECT time, sum(sales_units) FROM items WHERE typeid=#typeid# GROUP BY time ORDER BY time
```

werden:

Den Platzhalter #typeid# durch die jeweilige Typeid ersetzen.

Aufgabe 2

Die Preisindizes sind als Sichten (View) im Datenbanksystem hinterlegt und können wie eine Tabelle über

```
SELECT * FROM CPI
```

SQL angefragt werden. Beispiel für den CPI:

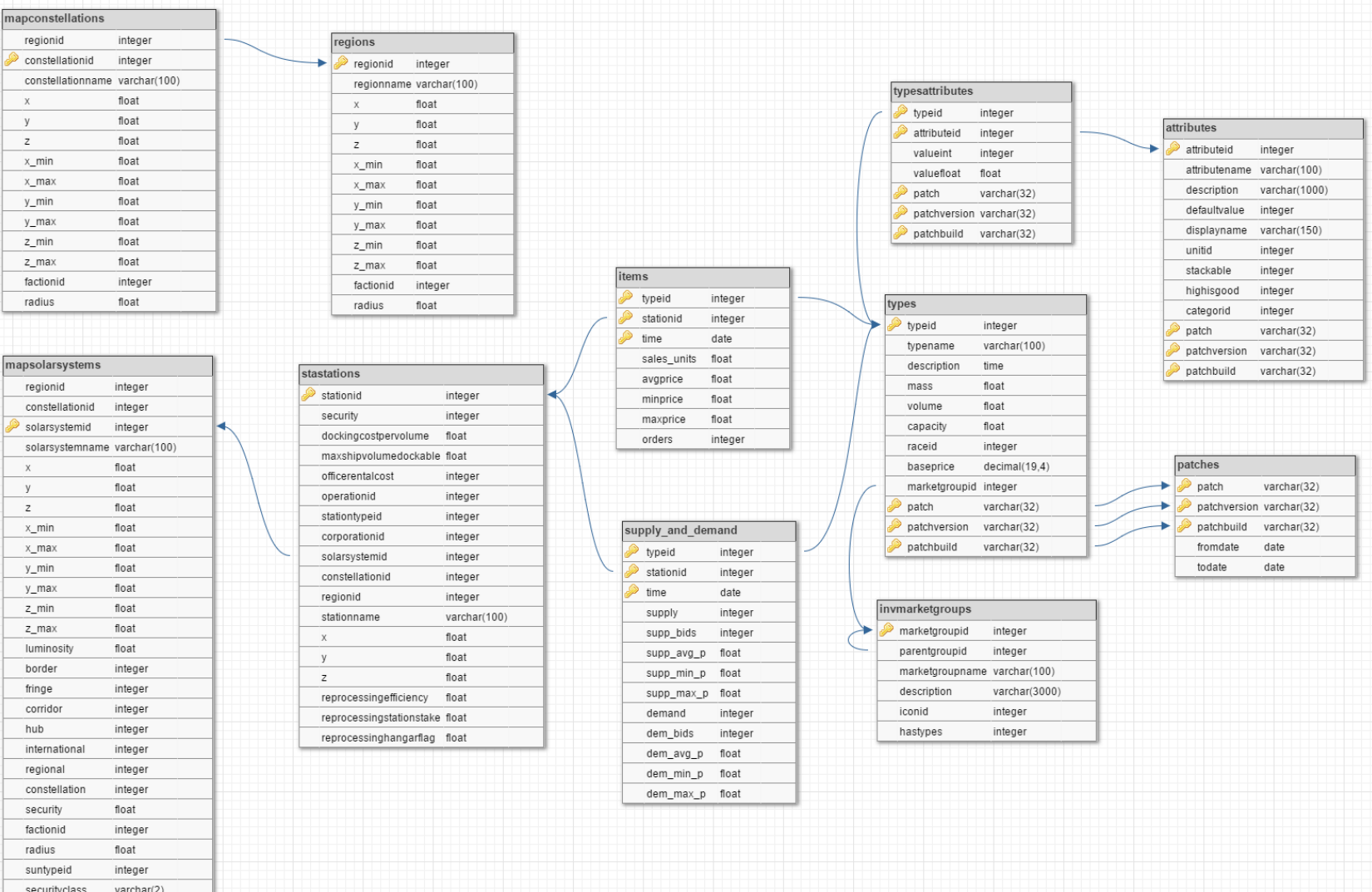


Abbildung 1: Datenbankschema