Overview

- purpose
- analytical modelling and OLAP
- slicing and dicing
- OLAP server
- SQL backend
analytical data modelling
lightweight
aggregation browsing

slicing and dicing
modelling

reporting

aggregation browsing
Architecture
Logical Model

multidimensional, analytical
business/analyst’s point of view
transactions
OLTP

application (operational) data

analysis
OLAP

analytical data
Model

{  
  "name" = "My Model"  
  "description" = ...  
  "cubes" = [...]  
  "dimensions" = [...]  
}

cubes

dimensions

measures

levels, attributes, hierarchy
Facts

most detailed information

measurable

fact data cell
dimensions
Dimension

- provide **context** for facts
- used to **filter** queries or reports
- control **scope of aggregation** of facts
Hierarchy

2010 May 1st

levels
Dimension

- levels and attributes
- hierarchy*
- key attributes
- label attributes

```
"dimensions" = [
  {
    "name":"date",
    "levels": ...
    "hierarchy": ...
  },
  ...
]
```

*partial support for multiple hierarchies
<table>
<thead>
<tr>
<th>Sector</th>
<th>Amount</th>
<th>Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction work</td>
<td>11 800 763 211 €</td>
<td>54.60 %</td>
</tr>
<tr>
<td>unknown</td>
<td>2 221 940 679 €</td>
<td>10.28 %</td>
</tr>
<tr>
<td>T services: consulting, software development, Internet and support</td>
<td>945 041 393 €</td>
<td>4.37 %</td>
</tr>
<tr>
<td>Radio, television, communication, telecommunication and related equipment</td>
<td>904 353 511 €</td>
<td>4.18 %</td>
</tr>
<tr>
<td>Transport equipment and auxiliary products to transportation</td>
<td>822 930 235 €</td>
<td>3.81 %</td>
</tr>
<tr>
<td>Repair and maintenance services</td>
<td>704 187 088 €</td>
<td>3.26 %</td>
</tr>
<tr>
<td>Architectural, construction, engineering and inspection services</td>
<td>654 548 217 €</td>
<td>3.03 %</td>
</tr>
<tr>
<td>Medical equipments, pharmaceuticals and personal care products</td>
<td>644 938 964 €</td>
<td>2.98 %</td>
</tr>
<tr>
<td>Office and computing machinery, equipment and supplies except furniture and software packages</td>
<td>534 306 205 €</td>
<td>2.47 %</td>
</tr>
</tbody>
</table>
Cube

- dimensions
- measures

```
"cubes" = [
    {
        "name": "contracts",
        "dimensions": [
            "date",
            "category"
        ],
        "measures": [
            {
                "name": "amount",
                "label": "Contract Amount",
                "aggregations": ["sum"]
            }
        ]
    },
    ...
]
```

*partial support for multiple hierarchies*
localizable
model and attributes

"attributes": [
{
   "name":"group",
   "label": "Group code"
},
{
   "name":"group_label",
   "label": "Group",
   "locales": ["en", "sk"]
}]

[Image]
Aggregation Browser
∑ measures

<table>
<thead>
<tr>
<th>Name</th>
<th>Amount</th>
<th>Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction work</td>
<td>11 800 763 211 €</td>
<td>54.60%</td>
</tr>
<tr>
<td>unknown</td>
<td>2 221 940 679 €</td>
<td>10.28%</td>
</tr>
<tr>
<td>IT services: consulting, software development, Internet a...</td>
<td>945 041 393 €</td>
<td>4.37%</td>
</tr>
<tr>
<td>Radio, television, communication, telecommunication and r...</td>
<td>904 353 511 €</td>
<td>4.18%</td>
</tr>
<tr>
<td>Transport equipment and auxiliary products to transportation</td>
<td>822 930 235 €</td>
<td>3.81%</td>
</tr>
<tr>
<td>Repair and maintenance services</td>
<td>704 187 088 €</td>
<td>3.26%</td>
</tr>
<tr>
<td>Architectural, construction, engineering and inspection s...</td>
<td>654 548 217 €</td>
<td>3.03%</td>
</tr>
<tr>
<td>Medical equipments, pharmaceuticals and personal care pro...</td>
<td>644 938 964 €</td>
<td>2.98%</td>
</tr>
<tr>
<td>Office and computing machinery, equipment and supplies ex...</td>
<td>534 306 205 €</td>
<td>2.47%</td>
</tr>
<tr>
<td>Petroleum products, fuel, electricity and other sources o...</td>
<td>493 242 675 €</td>
<td>2.28%</td>
</tr>
<tr>
<td>others</td>
<td>1 888 831 284 €</td>
<td>8.74%</td>
</tr>
</tbody>
</table>
“batteries” that are included
Browser Workspace

logical model + data
Cell
context of interest

cell
Path

Sector: Construction work - Works for complete or part construction and civil engineering work

Year: 2012 - June

[2012, 6] → [45, 2] → list of level keys
1. load_model("model.json")

2. create_workspace("sql", model, url="sqlite:///data.sqlite")

3. model.cube("sales")

4. workspace.browser(cube)
Number of contracts: 3,945
Contracts amount: 2,163,664,086 €
```javascript
browser.aggregate(cell)
```

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of contracts</td>
<td>3,945</td>
</tr>
<tr>
<td>Contracts amount</td>
<td>2,163,664,086 €</td>
</tr>
</tbody>
</table>
browser.aggregate(
    cell, 
    drilldown=["sector"]
)
for row in result.drilldown:

row["amount_sum"]

row[label_attribute]
browser.facts(cell)

browser.values(cell, dimension)

browser.cell_details(cell)
Slicing and Dicing
construction work in April 2012

- **type**: construction work
- **date**: April 2012
- **supplier**
cut types

point: [2010]

set: [[2010, 10], [2010, 12]]

range: from=[2010, 10] to=[2010, 12]
Implicit Hierarchy

drilldown
cell = Cell(cube)
browser.aggregate(cell)

cut = PointCut("date", [2010])

whole cube

cell = cell.slice(cut)
browser.aggregate(cell, drilldown=["date"])
Drill-down Level

```javascript
// drilldown = ["date"]

// implicit: next from cell

// drilldown = {
//   "date": "month"
// }
```
Cross Table

experimental interface
<table>
<thead>
<tr>
<th>Category</th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Due from Banks</td>
<td>3044</td>
<td>1803</td>
</tr>
<tr>
<td>Investments</td>
<td>41012</td>
<td>36012</td>
</tr>
<tr>
<td>Loans Outstanding</td>
<td>103657</td>
<td>118104</td>
</tr>
<tr>
<td>Nonnegotiable</td>
<td>1202</td>
<td>1123</td>
</tr>
<tr>
<td>Other Assets</td>
<td>2247</td>
<td>3071</td>
</tr>
<tr>
<td>Other Receivables</td>
<td>984</td>
<td>811</td>
</tr>
<tr>
<td>Receivables</td>
<td>176</td>
<td>171</td>
</tr>
<tr>
<td>Securities</td>
<td>33</td>
<td>289</td>
</tr>
<tr>
<td>Equity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capital Stock</td>
<td>11491</td>
<td>11492</td>
</tr>
<tr>
<td>Deferred Amounts</td>
<td>359</td>
<td>313</td>
</tr>
<tr>
<td>Other</td>
<td>-1683</td>
<td>-3043</td>
</tr>
<tr>
<td>Retained Earnings</td>
<td>29870</td>
<td>28793</td>
</tr>
<tr>
<td>Liabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borrowings</td>
<td>110040</td>
<td>128577</td>
</tr>
<tr>
<td>Derivative Liabilities</td>
<td>115642</td>
<td>110418</td>
</tr>
<tr>
<td>Other</td>
<td>57</td>
<td>8</td>
</tr>
<tr>
<td>Other Liabilities</td>
<td>7321</td>
<td>5454</td>
</tr>
<tr>
<td>Sold or Lent</td>
<td>2323</td>
<td>998</td>
</tr>
</tbody>
</table>
rows = ["item.category", "item.subcategory"]

columns = ["year"]

measures = ["amount_sum"]

table = result.cross_table(rows, columns, measures)
Slicer
The HTTP OLAP Server
GET /model
GET /aggregate
GET /values
GET /report
logical model  configuration  data

```
$ slicer serve slicer.ini
```
[server]
backend: sql
log_level: info

[model]
path: model.json
locales: en,sk

[workspace]
url: postgres://localhost/database
schema: datamart
fact_prefix: ft_
dimension_prefix: dm_
\[ \sum \text{amount} \]

GET /aggregate
GET aggregate

{
    "cell": [],
    "drilldown": [],
    "summary": {
        "record_count": 62,
        "amount_sum": 1116860
    }
}
\[ \sum \text{amount} \]

GET /aggregate?cut=date:2010
GET aggregate?cut=year:2010

```json
{
    "cell": [
        {
            "path": ["2010"],
            "type": "point",
            "dimension": "year",
            "level_depth": 1
        }
    ],
    "drilldown": [],
    "summary": {
        "record_count": 31,
        "amount_sum": 566020
    }
}
```
GET aggregate?drilldown=year

```json
{
    "cell": [],
    "total_cell_count": 2,
    "drilldown": [
        {
            "record_count": 31,
            "amount_sum": 550840,
            "year": 2009
        },
        {
            "record_count": 31,
            "amount_sum": 566020,
            "year": 2010
        }
    ],
    "summary": {
        "record_count": 62,
        "amount_sum": 1116860
    }
}
```
GET report

Content-Type: application/json

```json
{
    "cell": [
        {
            "dimension": "date",
            "type": "range",
            "from": [2009],
            "to": [2011, 6]
        }
    ],
    "queries": {
        "by_segment": {
            "query": "aggregate",
            "drilldown": ["segment"]
        },
        "by_year": {
            "query": "aggregate",
            "drilldown": {"date": "year"}
        }
    }
}
```
SQL Backend

What data does it work with?
or
dimensions

fact table
### Logical vs Physical

#### Logical

<table>
<thead>
<tr>
<th>Name</th>
<th>Amount</th>
<th>Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>open procedure</td>
<td>6 078 066 210 €</td>
<td>28.12%</td>
</tr>
<tr>
<td>unknown</td>
<td>5 479 519 613 €</td>
<td>25.35%</td>
</tr>
<tr>
<td>competitive dialogue</td>
<td>4 223 187 949 €</td>
<td>19.54%</td>
</tr>
<tr>
<td>restricted procedure</td>
<td>3 297 236 367 €</td>
<td>15.25%</td>
</tr>
<tr>
<td>negotiated procedure without publ.</td>
<td>2 537 073 324 €</td>
<td>11.74%</td>
</tr>
</tbody>
</table>

#### Physical
SQL Features

- does not require DB write access
- denormalisation
  - denormalised browsing, indexing
- simple date datatype dimension
  - extraction of date parts during mapping
- multiple schema support
Slicer

command-line tool
- model validation
  slicer model validate model.json

- model translation
  slicer model translate model.json translation.json

- workspace testing
  slicer test config.ini

- denormalization
  slicer denormalize --materialize --index config.ini
Future
- formatters for visualisation libraries
- JavaScript library*
- backends
- derived measures

*http://github.com/Stiivi/cubes-js
Open Data

- shared repository of models
- shared repository of dimensions
- public cubes

open Slicer HTTP APIs

http://github.com/Stiivi/cubes/wiki
stay light

**Nutrition Facts**
Serving Size 1 cube

<table>
<thead>
<tr>
<th>Amount Per Serving</th>
<th>% Daily Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Fat 0g</td>
<td>0%</td>
</tr>
<tr>
<td>Saturated Fat 0g</td>
<td></td>
</tr>
<tr>
<td>Trans Fat 0g</td>
<td></td>
</tr>
</tbody>
</table>
Thank You

source:
github.com/Stiivi/cubes

documentation:
packages.python.org/cubes/

eamples:
github.com/Stiivi/cubes-examples
this presentation:

bit.ly/cubes-ep2012