odoo

# Business Intelligence

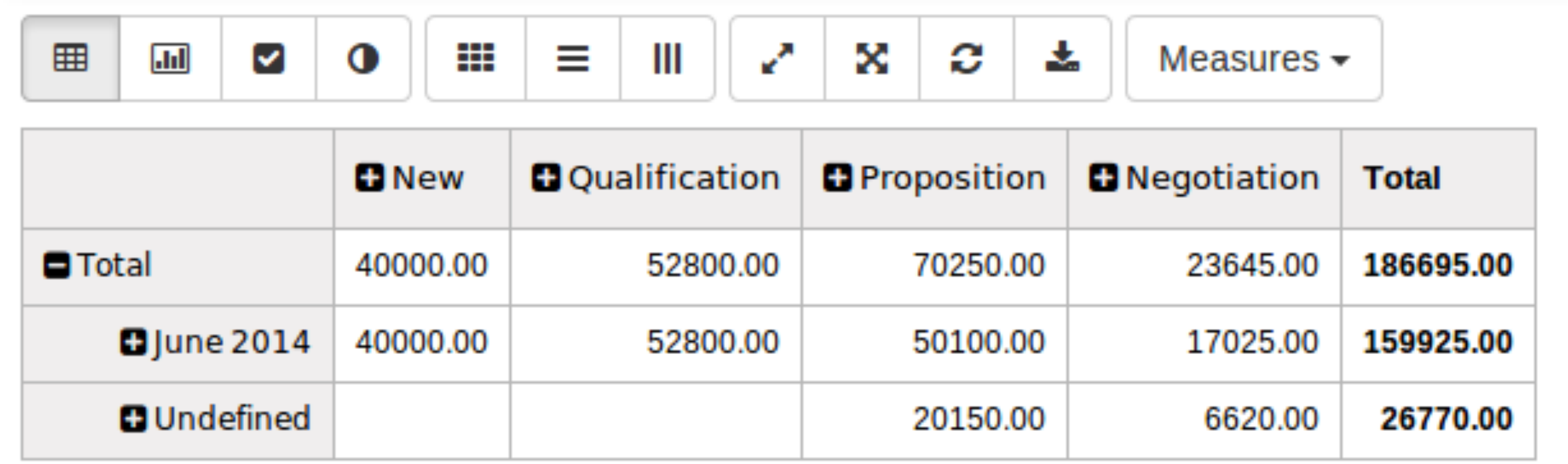
Develop cube views for your own objects

# Content

1. BI/Graph View
2. Technical Overview
3. Preparing Data
4. Displaying cube views
5. Case study: cross models
6. Conclusion

# BI/Graph View

# BI/Graph View



|  | New | Qualification | Proposition | Negotiation | Total |
|---|---|---|---|---|---|
| Total | 40000.00 | 52800.00 | 70250.00 | 23645.00 | **186695.00** |
| June 2014 | 40000.00 | 52800.00 | 50100.00 | 17025.00 | **159925.00** |
| Undefined |  |  | 20150.00 | 6620.00 | **26770.00** |

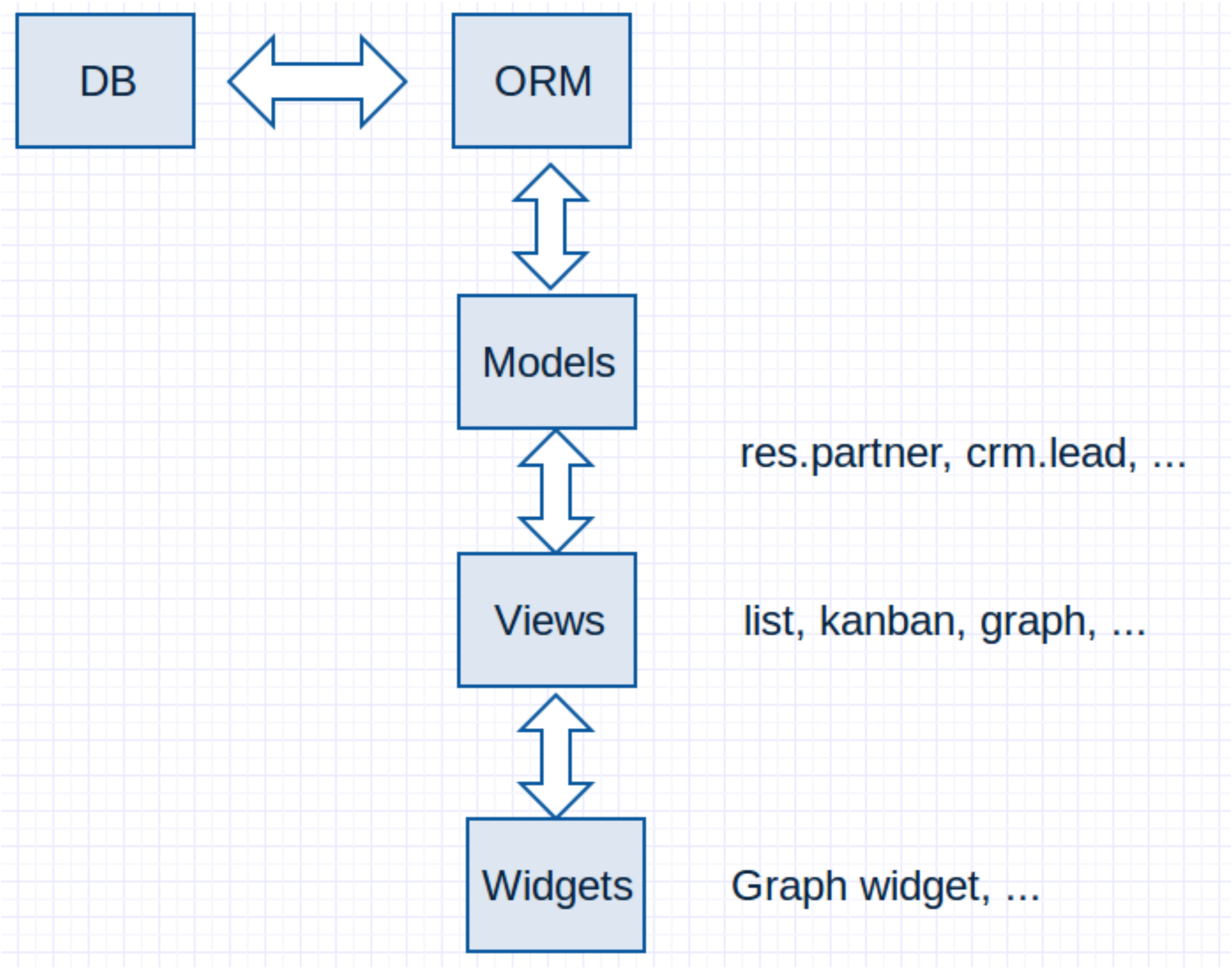
- measure : can be aggregated (right now, only summed)
- dimension : can be grouped

# A Short History of BI in Odoo

- pre 2014: list view + group bys, graph view
- Q1/Q2 2014: graph view rewrite -> pivot table + graphs, lots of backend work
- future: ? we're looking at searchview/BI view integration.

# Technical Overview

# Odoo architecture



DB

ORM

Models

res.partner, crm.lead, ...

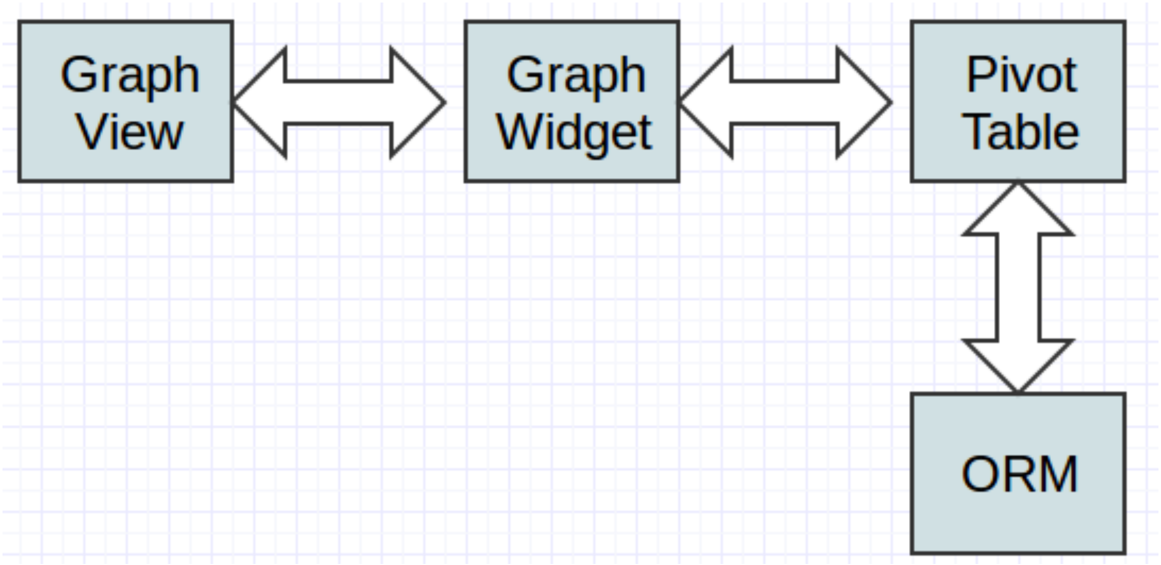
Views

list, kanban, graph, ...

Widgets

Graph widget, ...

# Anatomy of BI/Graph View



- **pivot table:** keeps the data, calls the ORM
- **graph widget** : user interaction
- **graph view** : client interaction

# BI view xml

```
<record id="..." model="ir.ui.view">
    <field name="name">crm.opportunity.report.graph</field>
    <field name="model">crm.lead.report</field>
    <field name="arch" type="xml">
        <graph string="Leads Analysis" type="pivot" stacked="True">
            <field name="date_deadline" type="row"/>
            <field name="stage_id" type="col"/>
            <field name="planned_revenue" type="measure"/>
        </graph>
    </field>
</record>
```

# BI view API

In *graph* tag:

- string: title
- stacked: if bar chart is stacked/not stacked (default=false)
- type: mode (pivot, bar, pie, line) (default=bar)

In *field* tags, *type* attribute:

- row : will be grouped by rows (dimension)
- col : will be grouped by cols (dimension)
- measure : will be aggregated
- if no type, measure by default

# Date/datetime

Always want to be different: date/datetime have a special syntax for groupby:

- field_date:day,
- field_date:week,
- field_date:month (default)
- field_date:quarter,
- field_date:year

```
<graph string="Leads Analysis" type="pivot" stacked="True">
    <field name="date_deadline:week" type="row"/>
    <field name="stage_id" type="col"/>
    <field name="planned_revenue" type="measure"/>
</graph>
```

# Graph widget setup

Graph widget has two more options:

- *visible_ui* (true) : indicate if UI is visible
- *heatmap_mode* ('none') can be set to row/col/both

# Preparing Data

# Odoo Model

Odoo BI view will read the various fields. Depending on their type, it will use them for

- measures : every fields of type *integer*, *float* (except 'id')
- dimensions :
    - right now: every fields defined in the 'group by' category in the search bar.
    - later: every field that can be grouped by the db

# Where is your data?

The data needs to satisfy two conditions:

- be stored in the database (beware of functional fields not stored)
- be accessed from one single odoo model

---

If yes, you're done. If not, two possibilities:

- can you extend a model? (stored functional fields, relational fields)
- can you create a custom model with a postgres view, to link the various models with the data?

---

Bottom line: it needs to be in the DB

# Extending a model

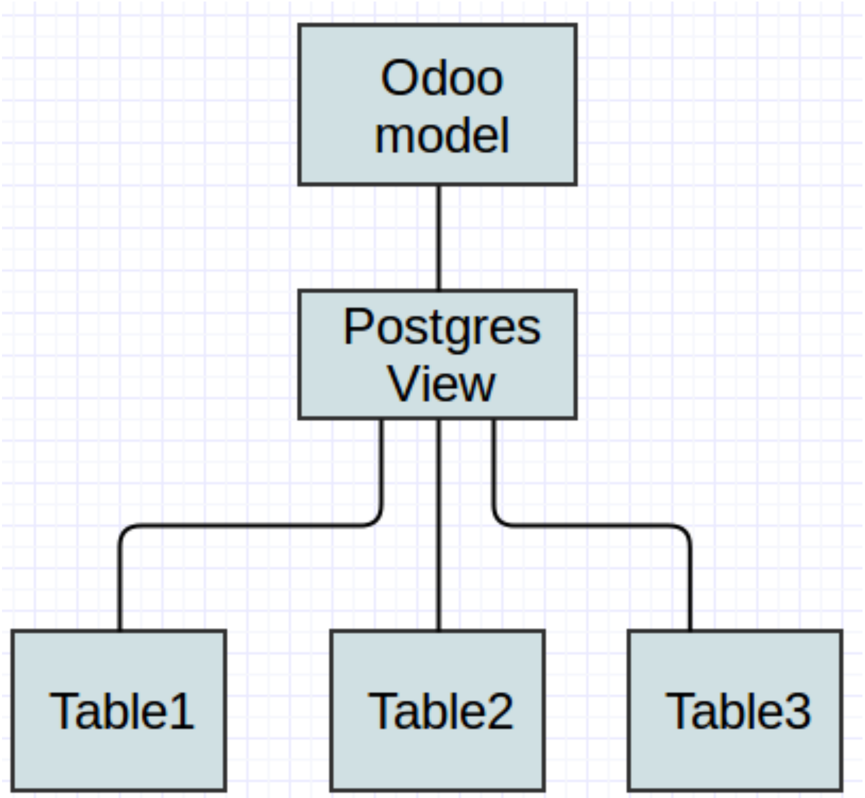
WARNING: old API... Do not try this at home!!!

```
class res_partner(osv.osv):
    _name = 'res.partner'
    _inherit = 'res.partner'

    def _total_invoice(self, cr, uid, ids, ...):
        ...
        # [insert here nice looking code to
        # compute the total invoice of a customer]
        ...
        return result

    _columns = {
        'total_invoiced': fields.function(_total_invoice,
                string="Total Invoiced", type='float', store=True)
        }
```

# More advanced: Cross model analysis



Example: purchase/report/purchase_report.py

All reporting views use that technique. Warning: bypass the ORM

# Displaying cube views

# Edit in live

1. go to developer mode
2. edit action, add 'graph',
3. edit views, create 'graph'
4. profit!

Good for testing.

# Adding a BI view with xml

Add the desired graph view:

```xml
<record id="view_project_task_graph" model="ir.ui.view">
    <field name="name">project.task.graph</field>
    <field name="model">project.task</field>
    <field name="arch" type="xml">
        <graph string="Project Tasks" type="bar">
            <field name="project_id" type="row"/>
            <field name="planned_hours" type="measure"/>
        </graph>
    </field>
</record>
```

# Adding a BI view with xml(2)

Add it to the action:

```
<record id="action_view_task" model="ir.actions.act_window">
    ...
    <field name="view_mode">kanban,tree,form,calendar,gantt,graph</field>
    ...
```

You can force the correct view:

```
<field name="view_id" ref="view_project_task_graph"/>
```

# Advanced: client action

In js, create a widget and append it to your view:

```
this.graph_widget = new openerp.web_graph.Graph(
    this,
    some_model,
    some_domain,
    options);
this.graph_widget.appendTo(this.$el);
```

# Future of BI in odoo?

# Thank you