eazyBI COMMUNITY DAYS 2023

Going Beyond 3D: Reporting in Multi-Dimensional Data Cube

Oskars Laganovskis

- **What is and why use MDX?**
- **MDX main concepts**
- **Report context**

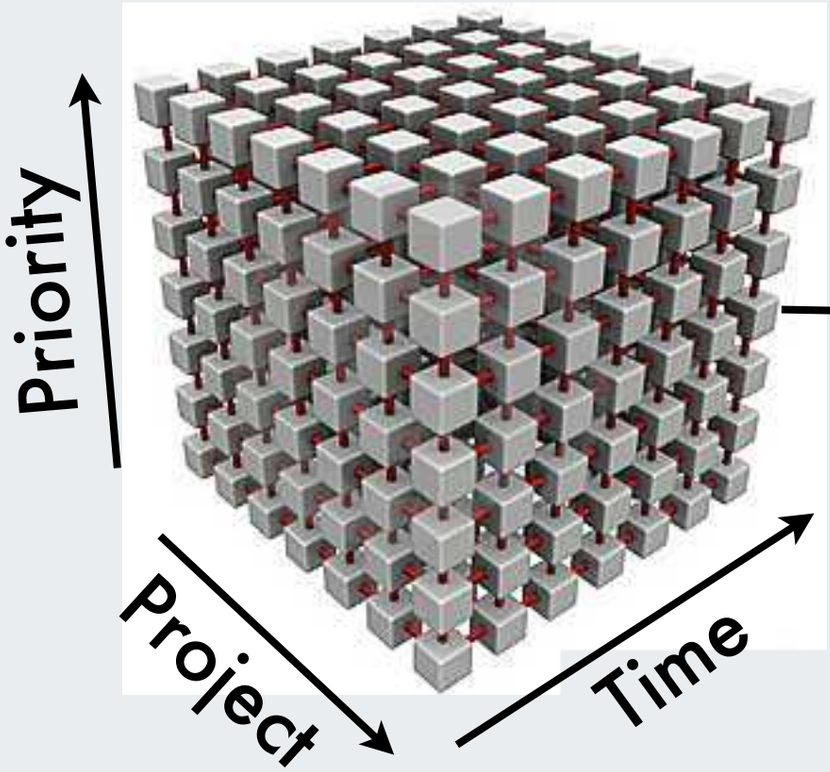
Multidimensional Expressions (MDX) is a query language for online analytical processing (OLAP) using a database management system

Multi-dimensional data model

Multi-dimensional
"data cube"

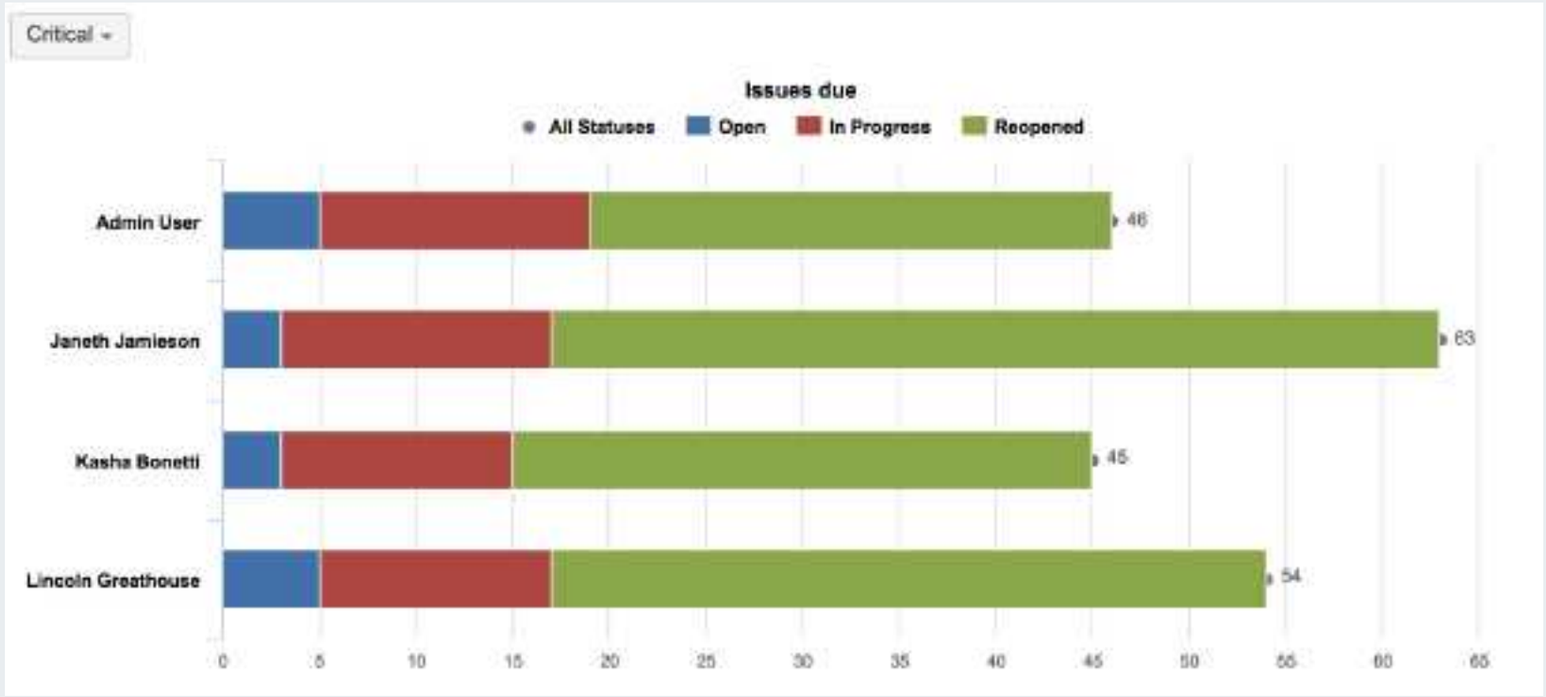
Dimensions
Hierarchies and levels

Measures



Measures
Issues created
Issues due
Issues resolved

eazyBI report



MDX query

```
1  select NON EMPTY Crossjoin({[Measures].[Issues due]}, {[Status].[All Statuses], [Status].[All Statuses].Children}) ON COLUMNS,
2    NON EMPTY Generate(NonEmptyCrossJoin([Assignee].[User].Members, {[Measures].DefaultMember}), {[Assignee].CurrentMember}) ON ROWS
3  from [Issues]
4  where [Priority].[Critical]
```

SQL query

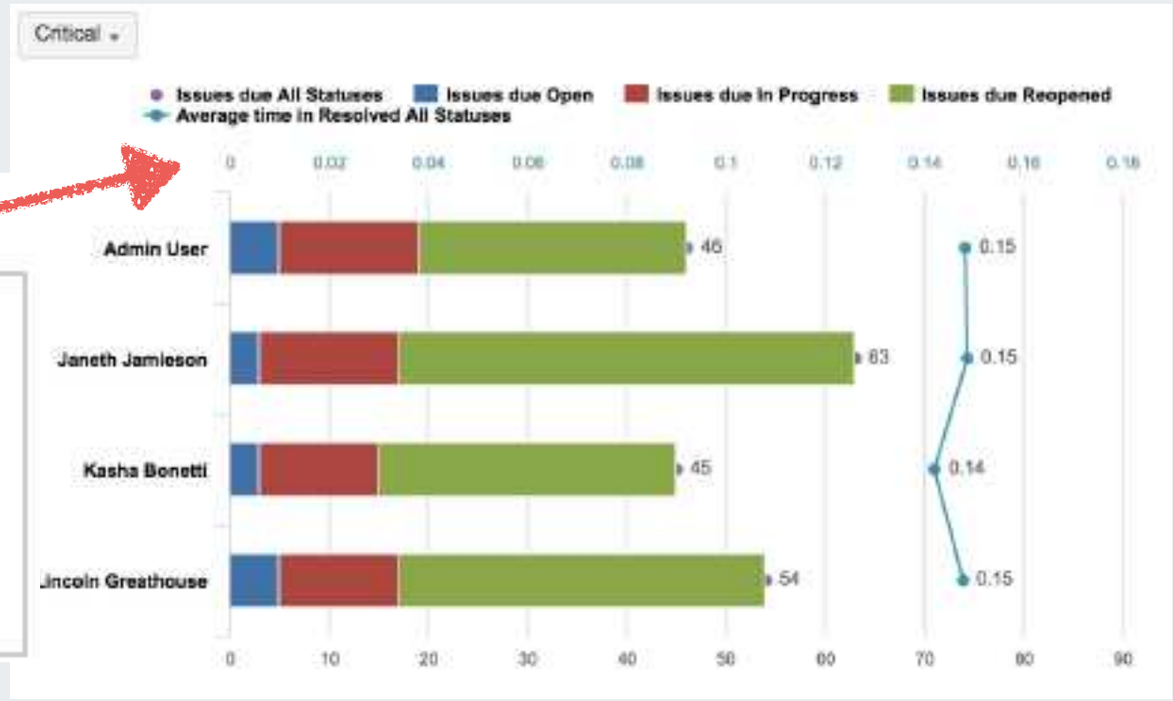
```
1  [select `jira_priorities`.`id` as `c0`, `jira_statuses`.`id` as `c1`, sum(`jira_issues_measures`.`issue_due`) as `m0` from
   `jira_priorities` as `jira_priorities`, `jira_issues_measures` as `jira_issues_measures`, `jira_statuses` as `jira_statuses` where
   `jira_issues_measures`.`priority_id` = `jira_priorities`.`id` and `jira_priorities`.`id` = 2 and `jira_issues_measures`.`status_id`
   = `jira_statuses`.`id` group by `jira_priorities`.`id`, `jira_statuses`.`id`]
2  [select `jira_priorities`.`id` as `c0`, sum(`jira_issues_measures`.`issue_due`) as `m0` from `jira_priorities` as
   `jira_priorities`, `jira_issues_measures` as `jira_issues_measures` where `jira_issues_measures`.`priority_id` =
   `jira_priorities`.`id` and `jira_priorities`.`id` = 2 group by `jira_priorities`.`id`]
3  [[Assignee].[User]]: executing sql [select `jira_assignees`.`name` as `c0`, `jira_assignees`.`display_name` as `c1`,
   `jira_assignees`.`display_name` as `c2`, `jira_assignees`.`name` as `c3`, `jira_assignees`.`groups` as `c4`,
   `jira_assignees`.`email` as `c5`, `jira_assignees`.`user_key` as `c6` from `jira_assignees` as `jira_assignees` group by
   `jira_assignees`.`name`, `jira_assignees`.`display_name` order by ISNULL(`c1`) ASC, `c1` ASC]
4  [select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`, sum(`jira_issues_measures`.`issue_due`) as `m0` from
   `jira_assignees` as `jira_assignees`, `jira_issues_measures` as `jira_issues_measures`, `jira_priorities` as `jira_priorities`
   where `jira_issues_measures`.`assignee_id` = `jira_assignees`.`id` and `jira_issues_measures`.`priority_id` =
   `jira_priorities`.`id` and `jira_priorities`.`id` = 2 group by `jira_assignees`.`name`, `jira_priorities`.`id`]
5  [select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`, `jira_statuses`.`id` as `c2`,
   sum(`jira_issues_measures`.`issue_due`) as `m0` from `jira_assignees` as `jira_assignees`, `jira_issues_measures` as
   `jira_issues_measures`, `jira_priorities` as `jira_priorities`, `jira_statuses` as `jira_statuses` where
   `jira_issues_measures`.`assignee_id` = `jira_assignees`.`id` and `jira_issues_measures`.`priority_id` = `jira_priorities`.`id` and
   `jira_priorities`.`id` = 2 and `jira_issues_measures`.`status_id` = `jira_statuses`.`id` group by `jira_assignees`.`name`,
   `jira_priorities`.`id`, `jira_statuses`.`id`]
```

eazyBI report

[Measures].[Average time in Resolved] =

```
Case when
  [Status].Currentmember is [Status].DefaultMember
Then
  ([Measures].[Days in transition status],
   [Transition status].[Resolved])
   /
   ([Measures].[Transitions from status issues count],
   [Transition status].[Resolved])
End
```

MDX calculated member



MDX query

```
select NON EMPTY Crossjoin({[Measures].[Issues due], [Measures].[Average time in Resolved]}, {[Status].[All Statuses],
[Status].[All Statuses].Children}) ON COLUMNS,
  NON EMPTY Generate(NonEmptyCrossJoin([Assignee].[User].Members, {[Measures].DefaultMember}),
  {[Assignee].CurrentMember}) ON ROWS
from [Issues]
where [Priority].[Critical]
```

SQL query

```
(select `jira_priorities`.`id` as `c0`, sum(`jira_issues_measures`.`issue_due`) as `m0` from `jira_priorities` as
`jira_priorities`, `jira_issues_measures` as `jira_issues_measures` where `jira_issues_measures`.`priority_id` =
`jira_priorities`.`id` and `jira_priorities`.`id` = 2 group by `jira_priorities`.`id`)
(select `jira_priorities`.`id` as `c0`, sum(`jira_issues_measures_c1`.`days_in_transition_status`) as `m0`, count(
distinct CASE WHEN `jira_issues_measures_c1`.`transition_from_status` IS NOT NULL THEN
`jira_issues_measures_c1`.`issue_id` END) as `m1` from `jira_priorities` as `jira_priorities`, `jira_issues_measures_c1`
as `jira_issues_measures_c1` where `jira_issues_measures_c1`.`priority_id` = `jira_priorities`.`id` and
`jira_priorities`.`id` = 2 group by `jira_priorities`.`id`)
(select `jira_priorities`.`id` as `c0`, `jira_statuses`.`id` as `c1`, sum(`jira_issues_measures`.`issue_due`) as `m0`
from `jira_priorities` as `jira_priorities`, `jira_issues_measures` as `jira_issues_measures`, `jira_statuses` as
`jira_statuses` where `jira_issues_measures`.`priority_id` = `jira_priorities`.`id` and `jira_priorities`.`id` = 2 and
`jira_issues_measures`.`status_id` = `jira_statuses`.`id` group by `jira_priorities`.`id`, `jira_statuses`.`id`)
(select `jira_priorities`.`id` as `c0`, `jira_statuses`.`id` as `c1`,
sum(`jira_issues_measures_c1`.`days_in_transition_status`) as `m0`, count(distinct CASE WHEN
`jira_issues_measures_c1`.`transition_from_status` IS NOT NULL THEN `jira_issues_measures_c1`.`issue_id` END) as `m1`
from `jira_priorities` as `jira_priorities`, `jira_issues_measures_c1` as `jira_issues_measures_c1`, `jira_statuses` as
`jira_statuses` where `jira_issues_measures_c1`.`priority_id` = `jira_priorities`.`id` and `jira_priorities`.`id` = 2
and `jira_issues_measures_c1`.`status_id` = `jira_statuses`.`id` group by `jira_priorities`.`id`, `jira_statuses`.`id`)
[[Assignee],[User]]: executing sql [select `jira_assignees`.`name` as `c0`, `jira_assignees`.`display_name` as `c1`,
`jira_assignees`.`display_name` as `c2`, `jira_assignees`.`name` as `c3`, `jira_assignees`.`groups` as `c4`,
`jira_assignees`.`email` as `c5`, `jira_assignees`.`user_key` as `c6` from `jira_assignees` as `jira_assignees` group by
`jira_assignees`.`name`, `jira_assignees`.`display_name` order by ISNULL(`c1`) ASC, `c1` ASC]
(select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`, sum(`jira_issues_measures`.`issue_due`) as `m0`
from `jira_assignees` as `jira_assignees`, `jira_issues_measures` as `jira_issues_measures`, `jira_priorities` as
`jira_priorities` where `jira_issues_measures`.`assignee_id` = `jira_assignees`.`id` and
`jira_issues_measures`.`priority_id` = `jira_priorities`.`id` and `jira_priorities`.`id` = 2 group by
`jira_assignees`.`name`, `jira_priorities`.`id`)
(select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`,
sum(`jira_issues_measures_c1`.`days_in_transition_status`) as `m0`, count(distinct CASE WHEN
`jira_issues_measures_c1`.`transition_from_status` IS NOT NULL THEN `jira_issues_measures_c1`.`issue_id` END) as `m1`
from `jira_assignees` as `jira_assignees`, `jira_issues_measures_c1` as `jira_issues_measures_c1`, `jira_priorities` as
`jira_priorities` where `jira_issues_measures_c1`.`assignee_id` = `jira_assignees`.`id` and
`jira_issues_measures_c1`.`priority_id` = `jira_priorities`.`id` and `jira_priorities`.`id` = 2 group by
`jira_assignees`.`name`, `jira_priorities`.`id`)
(select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`, `jira_statuses`.`id` as `c2`,
sum(`jira_issues_measures`.`issue_due`) as `m0` from `jira_assignees` as `jira_assignees`, `jira_issues_measures` as
`jira_issues_measures`, `jira_priorities` as `jira_priorities`, `jira_statuses` as `jira_statuses` where
`jira_issues_measures`.`assignee_id` = `jira_assignees`.`id` and `jira_issues_measures`.`priority_id` =
`jira_priorities`.`id` and `jira_priorities`.`id` = 2 and `jira_issues_measures`.`status_id` = `jira_statuses`.`id`
group by `jira_assignees`.`name`, `jira_priorities`.`id`, `jira_statuses`.`id`)
(select `jira_assignees`.`name` as `c0`, `jira_priorities`.`id` as `c1`, `jira_transition_statuses`.`id` as `c2`,
sum(`jira_issues_measures_c1`.`days_in_transition_status`) as `m0`, count(distinct CASE WHEN
`jira_issues_measures_c1`.`transition_from_status` IS NOT NULL THEN `jira_issues_measures_c1`.`issue_id` END) as `m1`
from `jira_assignees` as `jira_assignees`, `jira_issues_measures_c1` as `jira_issues_measures_c1`, `jira_priorities` as
`jira_priorities`, `jira_transition_statuses` as `jira_transition_statuses` where
`jira_issues_measures_c1`.`assignee_id` = `jira_assignees`.`id` and `jira_issues_measures_c1`.`priority_id` =
`jira_priorities`.`id` and `jira_priorities`.`id` = 2 and `jira_issues_measures_c1`.`transition_status_id` =
`jira_transition_statuses`.`id` and `jira_transition_statuses`.`id` = 5 group by `jira_assignees`.`name`,
`jira_priorities`.`id`, `jira_transition_statuses`.`id`)
```

MDX advantages

- eazyBI creates MDX queries automatically
- Users can extend reports with MDX calculated members
- Mondrian stores results in cache
- Mondrian runs SQL query when there are no cached results

MDX through sample

Multi-dimensional data cube 'Simpsons'

picture from: http://nerdist.com/fox-teases-a-simpsons-character-death-futurama-crossover/

Dimensions

Measures

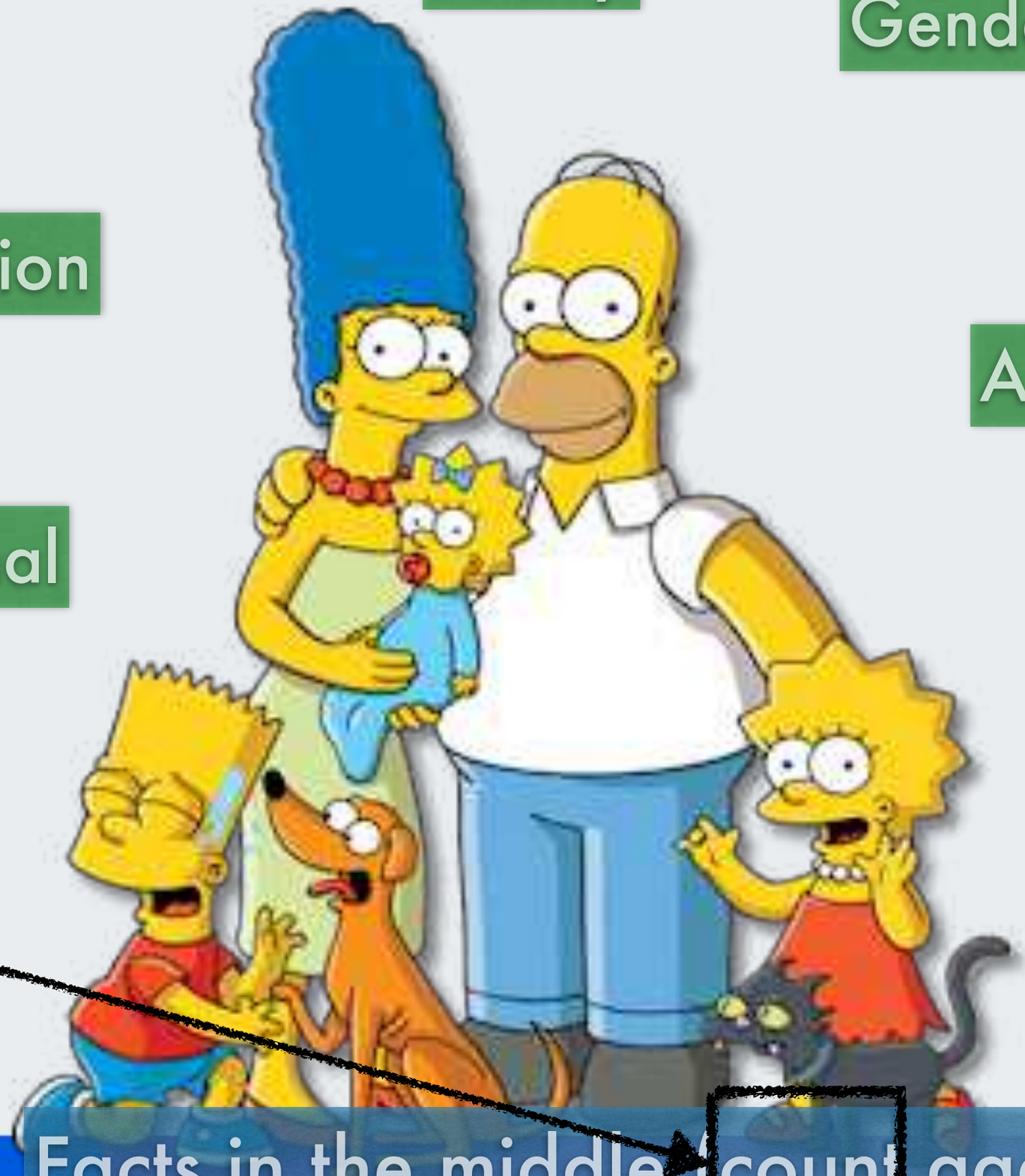
Family

Gender

Occupation

Adult/child

Human/animal

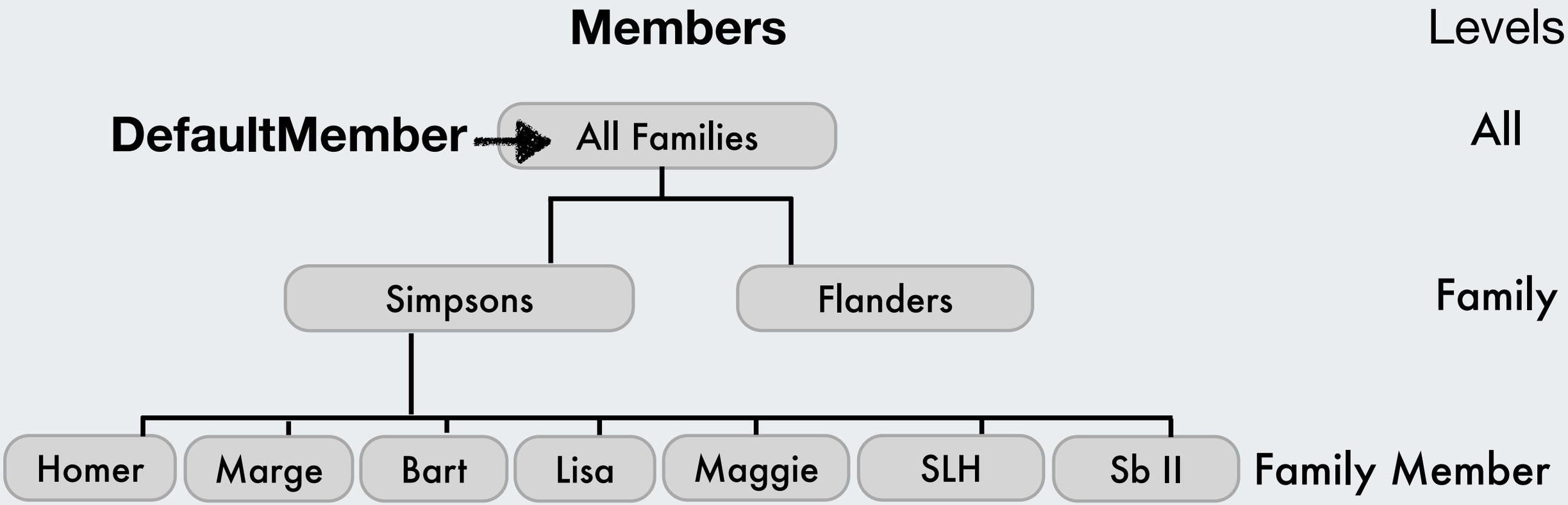


Default Measure

Facts in the middle (count, age)

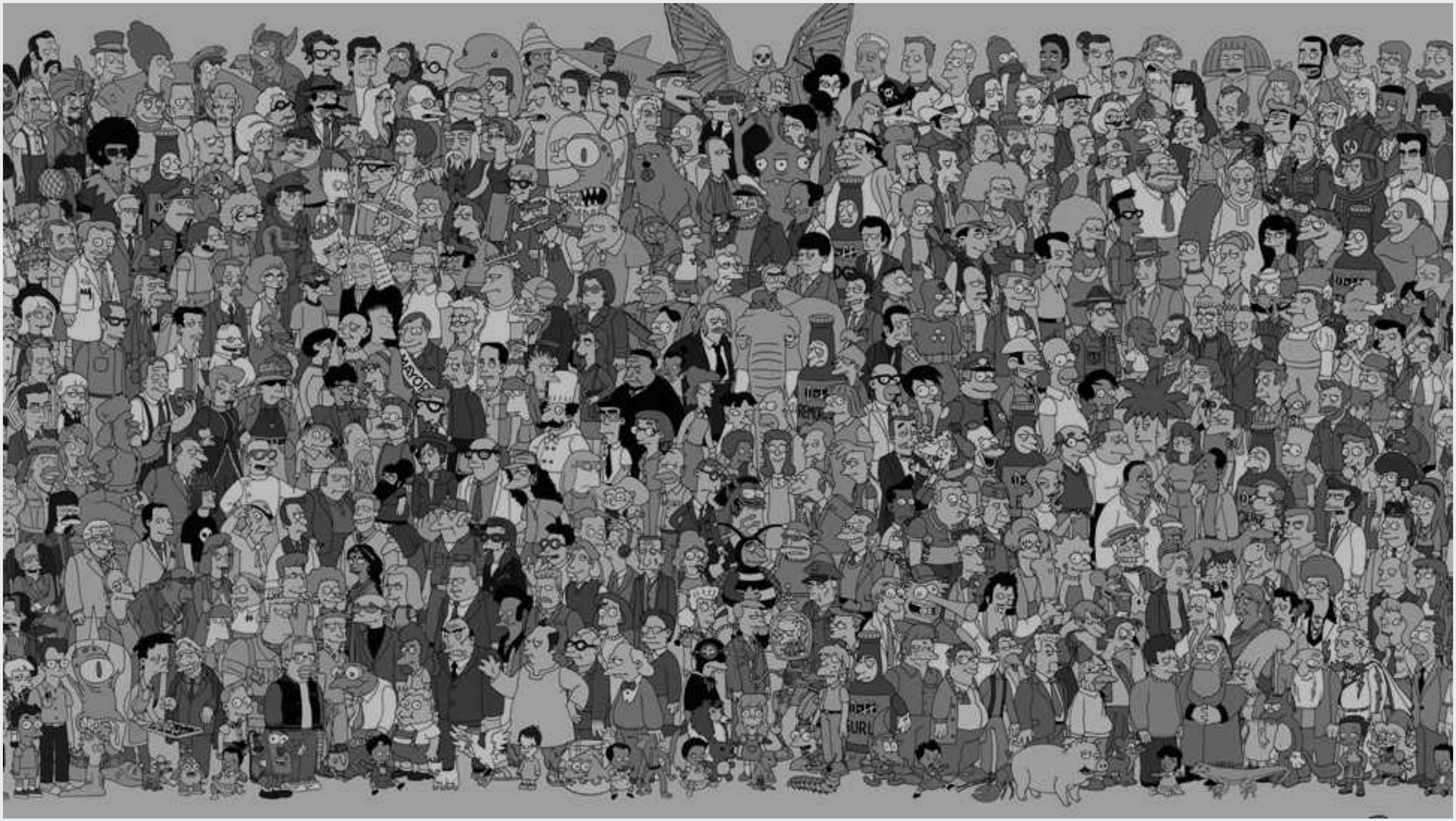
picture from: Wikipedia

Dimension ‘Family' has one hierarchy



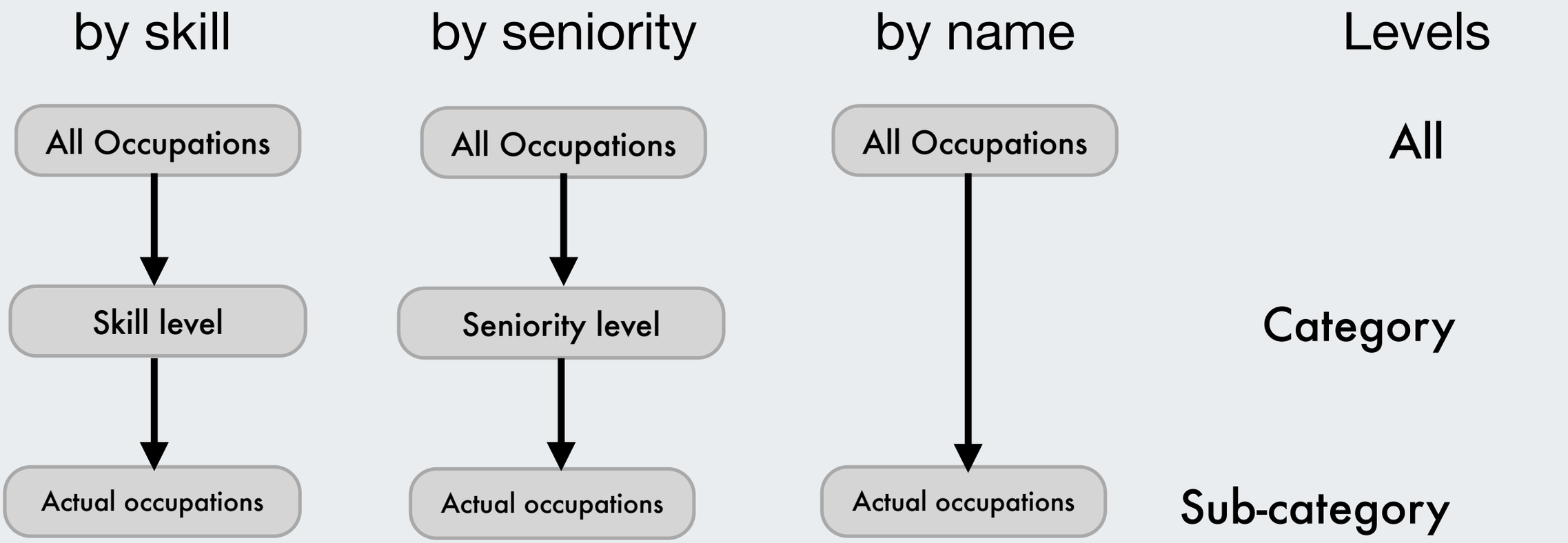
CurrentMember could be any member based on the context

[Family].DefaultMember



picture from: http://nerdist.com/fox-teases-a-simpsons-character-death-futurama-crossover/

Dimension can have several hierarchies defining category grouping



CurrentMember could be any member based on the context

Property

[Family].[Family].Members

[Family].[Simpsons]

Name: Simpsons
Address: 742 Evergreen Terrace, Springfield, United States
First appearance:
April 19, 1987

[Family].[Simpsons].Get('Address')

picture from: wikipedia

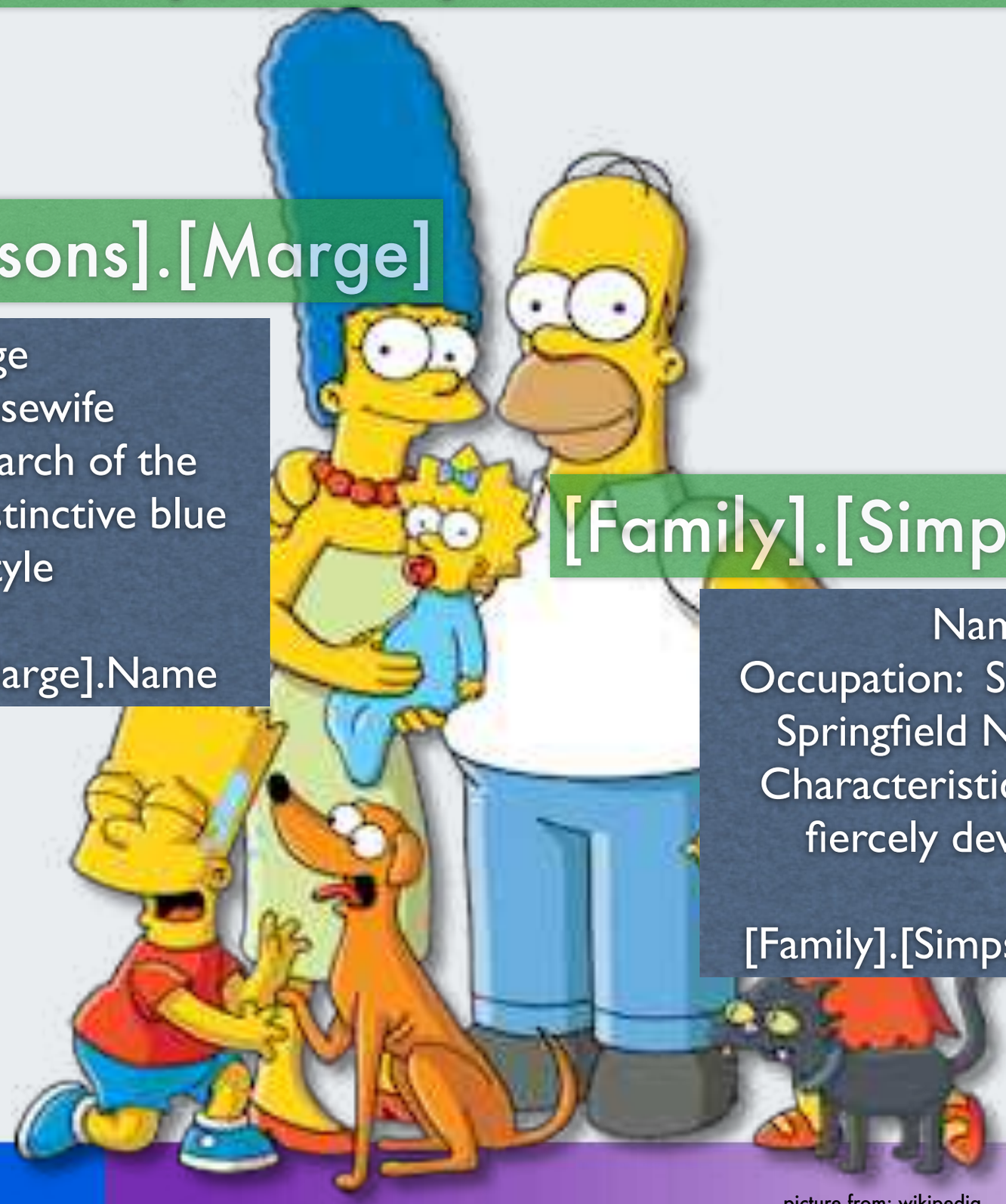
Property

[Family].[Family Member].Members

[Family].[Simpsons].[Marge]

Name: Marge
Occupation: Housewife
Characteristics: Matriarch of the Simpson family with distinctive blue beehive hairstyle

[Family].[Simpsons].[Marge].Name



[Family].[Simpsons].[Homer]

Name: Homer
Occupation: Safety Inspector at the Springfield Nuclear Power Plant
Characteristics: a decent man and fiercely devoted to his family

[Family].[Simpsons].[Homer].Name

picture from: wikipedia

Dimensions, Measures, Properties

- **Dimensions** categorize your data
- Dimensions use **hierarchical** structure to organise members
- **Measures** glue dimensions together
- **Properties** are attributes for **Members**

[Family].[Simpsons]

Gender

Occupation

Adult/child

Human/animal

[Measures].[count] = 7

picture from: wikipedia

[Family].[Family Member].Members

Gender

[Measures].[count] = 1

Occupation

Adult/child

[Human/animal].[Human]

[Measures].[count] = 1

[Measures].[count] = 1

[Measures].[count] = 1

picture from: wikipedia

Tuple

- Tuple is a way of addressing a cross-join of dimensions
- Tuple works within and can override existing report context

[Family].[Simpsons]

[Gender].[Male]

[Measures].[count] = 2

Occupation

Adult/child

[Human/animal].[Human]

[Measures].[children] = 1

[Measures].[children]
=
([Measures].[count],
[Human/animal].[Human],
[Adult/child].[child])

picture from: wikipedia

[Family].[Simpsons]

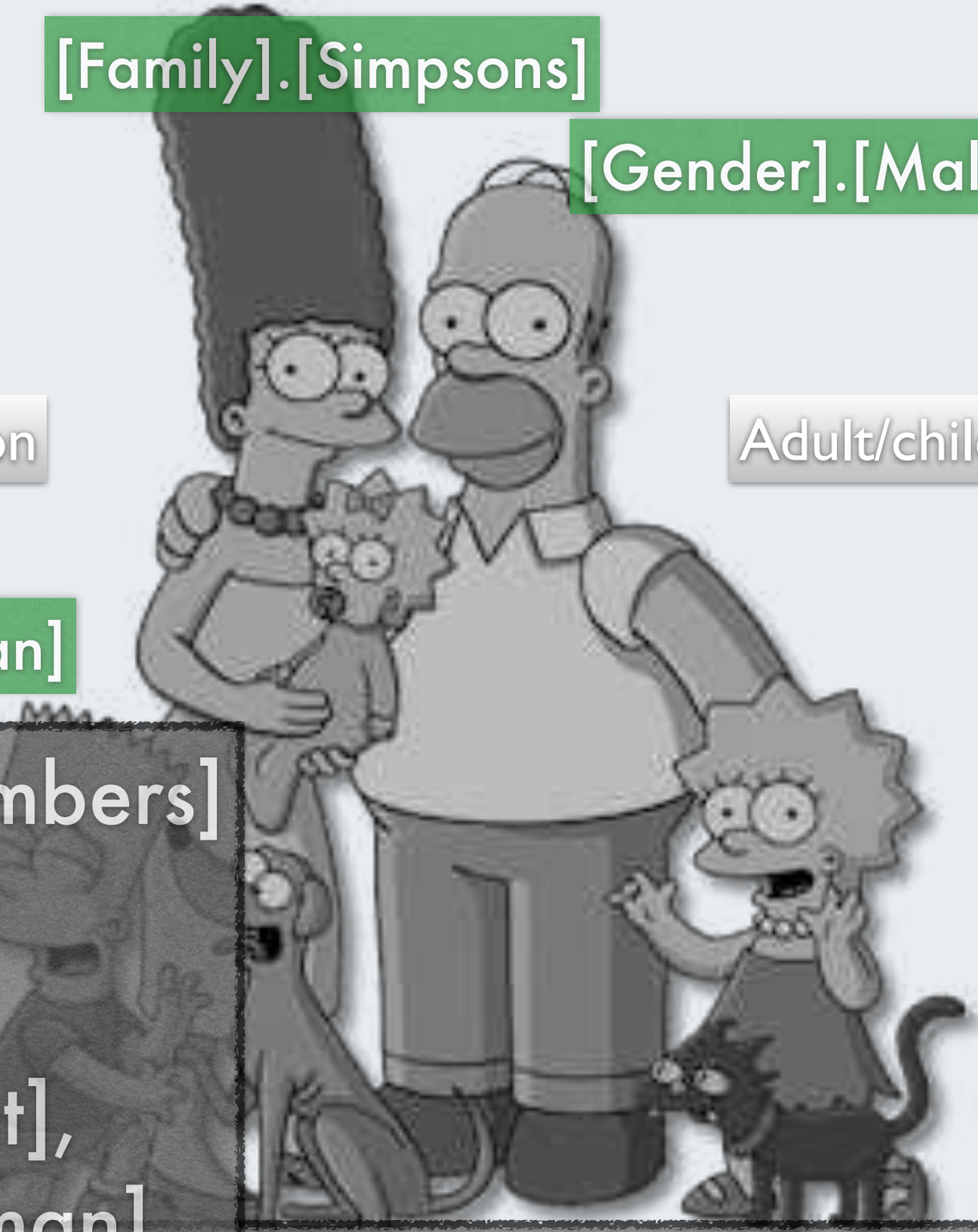
[Gender].[Male]

[Measures].[count] = 2

Occupation

Adult/child

[Human/animal].[Human]



[Measures].[family members]
=
DefaultContext
(([Measures].[count],
[Human/animal].[Human],
[Family].CurrentMember))

[Measures].[family members] = 5

picture from: wikipedia

Reports, Tuples, Sets

- Any **Measure** works in report context
- When using a **Member** there is always a **Measure** (default Measure if no measure specified)
- With a **Tuple** you could override report context
- **Set** has no context to report, use measures to get the context

Sum it up

- MDX query and MDX calculated members
- Measures glue dimensions together
- CurrentMember represents a member in report / set context
- There is no such object as CurrentSet
- Tuple takes into account report context and you could modify this
- Set does not take into account report context, use measure over Set as a filter

Questions?

community.eazybi.com
support@eazybi.com

Thank you!

eazybi.com