Flexible Data Acquisition and Analysis

Joe Conway
mail@joeconway.com

O'Reilly Open Source Convention
July 26–30, 2004

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Live example

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Scenario

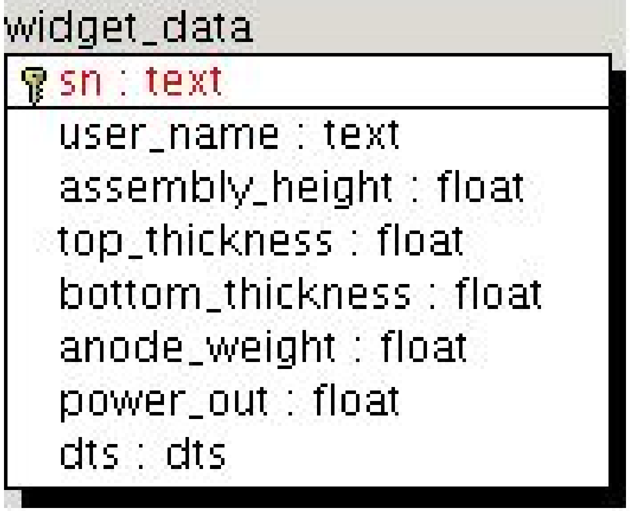
- Product "widget": assembly with three components
  - top shell
  - bottom shell
  - anode
- Critical attributes
  - assembly height
  - top/bottom shell thickness
  - anode weight
- Inspection test results
  - measured power output

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Simple Data Model

- Pros
  - Flat
  - Self Contained
  - Easily understood
  - Simple to extract from
- Cons
  - Specific
  - Inflexible
  - Manual maintenance
  - Doesn't scale



widget_data

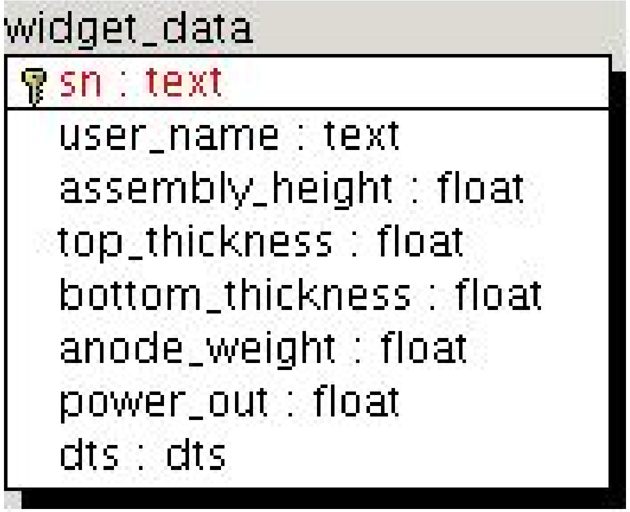
sn : text

user_name : text
assembly_height : float
top_thickness : float
bottom_thickness : float
anode_weight : float
power_out : float
dts : dts

Trade-off simplicity for higher long-term maintenance cost

Simple Data Model - widget_data

- Data collected for the widget assembly
  - sn: the serial number of this instance of the assembly
  - user_name: person recording the data
  - assembly_height: the height of the final assembly
  - top_thickness: the wall thickness of the "top" component of the assembly
  - bottom_thickness: the wall thickness of the "bottom" component of the assembly
  - anode_weight: the weight of the "anode" component
  - power_out: measured power output of the assembly
  - dts: date and time of data collection



widget_data
sn : text
user_name : text
assembly_height : float
top_thickness : float
bottom_thickness : float
anode_weight : float
power_out : float
dts : dts

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Data input: simple data model

```
INSERT INTO widget_data
VALUES ('wsn101', 'Jim', 7.251, 0.754,
        0.756,2.01, 18.123, 'today');
INSERT 3565581 1
```

Data extraction: simple data model

```
select * from widget_data where dts = 'today';
-[ RECORD 1 ]----+-----------------------
sn               | wsn101
user_name        | Jim
assembly_height  | 7.251
top_thickness    | 0.754
bottom_thickness | 0.756
anode_weight     | 2.01
power_out        | 18.123
dts              | 2004-06-19 00:00:00-07
```

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Scope changes

- Add subcomponents
  - e.g. a baffle and its height now needs to be tracked
- Add attributes
  - e.g. anode length is determined to be critical
- Add test results
  - e.g. assembly voltage test result (PASS/FAIL) must be recorded
- Other common scope changes:
  - New version of old product
  - New or additional products

All require data model and corresponding application code changes made by a developer. There has to be a better way. Fortunately there is ...

Agenda

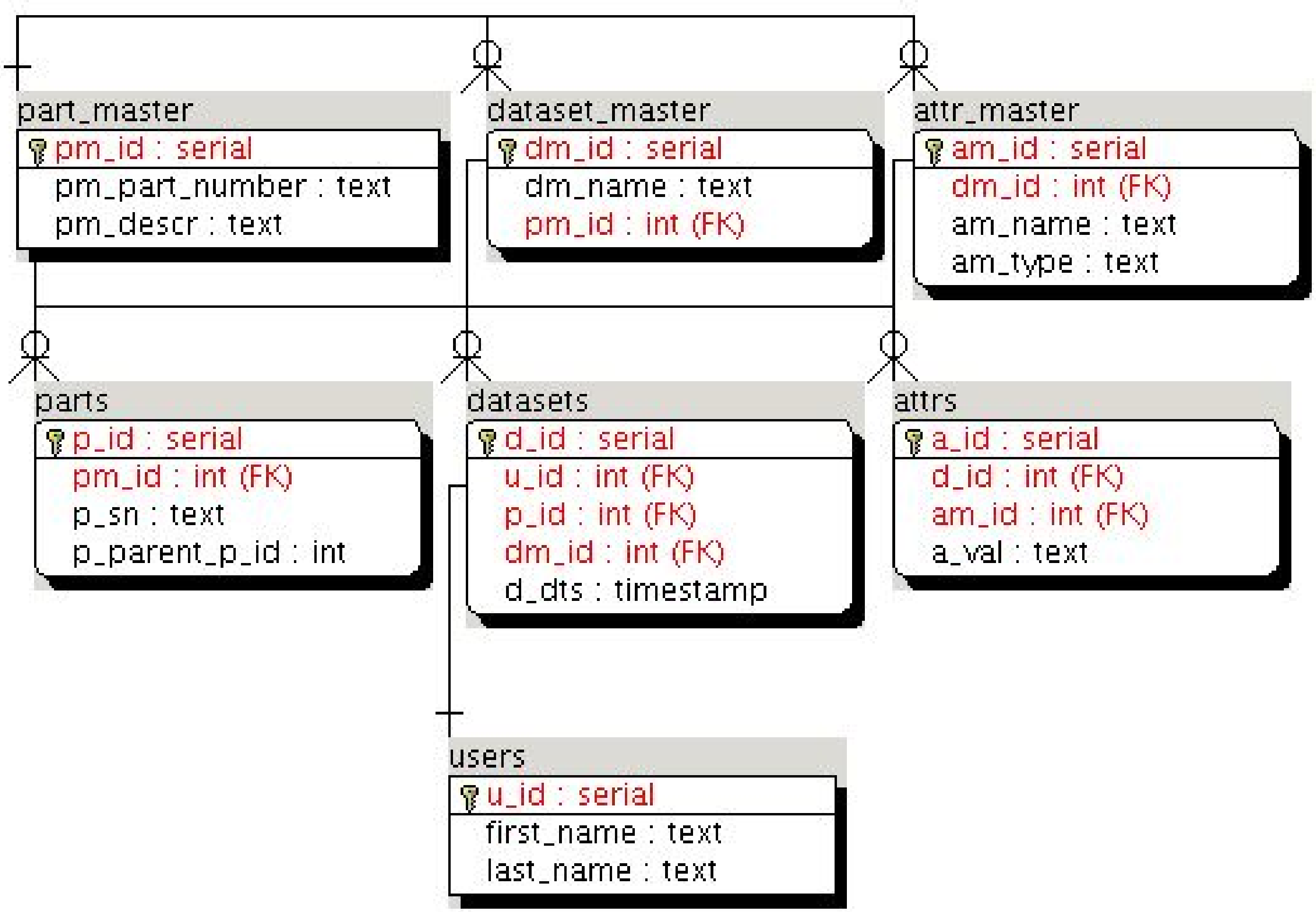
- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Flexible data model

- Pros
  - Higher degree of abstraction
  - Allows for "configuration" by a user-administrator instead of "customization" by a developer
  - Quickly adaptable; can collect arbitrary data
  - Scalable
- Cons
  - Complex data model
  - More difficult to understand
  - Initial application development complexity
  - Data extraction complexity

Trade-off developer complexity for long-term ease of maintainance

Flexible data model



part_master
- pm_id : serial
- pm_part_number : text
- pm_descr : text

dataset_master
- dm_id : serial
- dm_name : text
- pm_id : int (FK)

attr_master
- am_id : serial
- dm_id : int (FK)
- am_name : text
- am_type : text

parts
- p_id : serial
- pm_id : int (FK)
- p_sn : text
- p_parent_p_id : int

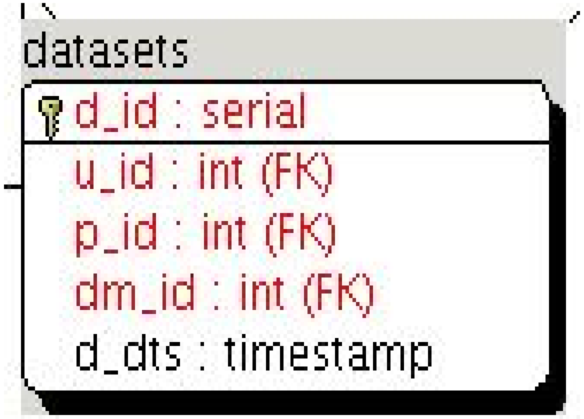
datasets
- d_id : serial
- u_id : int (FK)
- p_id : int (FK)
- dm_id : int (FK)
- d_dts : timestamp

attrs
- a_id : serial
- d_id : int (FK)
- am_id : int (FK)
- a_val : text

users
- u_id : serial
- first_name : text
- last_name : text

Flexible data model - datasets

- Primary record of a data collection event
  - d_id: primary key
  - u_id: the user that created the record
  - p_id: the serialized part to which the data relates
  - dm_id: the template this dataset instance derives from
  - d_dts: actual date and time (including time zone) of the data collection event
  - (u_id, p_id, dm_id, d_dts) is unique



datasets

d_id : serial

u_id : int (FK)

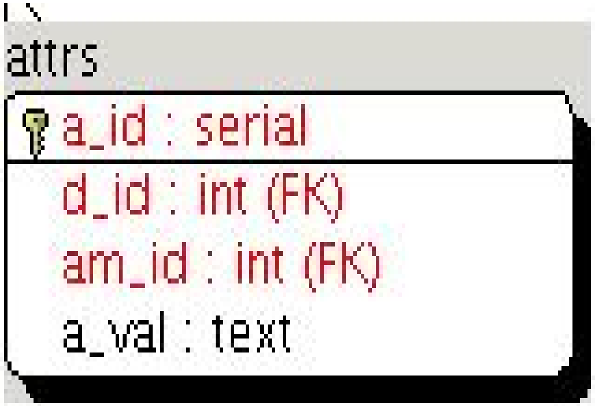
p_id : int (FK)

dm_id : int (FK)

d_dts : timestamp

Flexible data model - attrs

- Stores the actual attribute datums
  - a_id: primary key
  - d_id: the dataset to which the datum relates
  - am_id: the template this attribute instance derives from
  - a_val: the datum
  - (d_id, am_id) is unique



attrs

a_id : serial

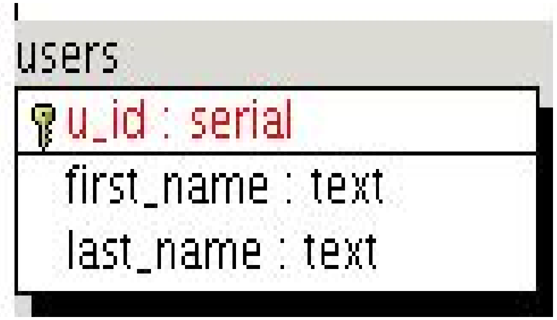
d_id : int (FK)

am_id : int (FK)

a_val : text

Flexible data model - users

- Stores the application users list
  - u_id: primary key
  - first_name: the user's first name
  - last_name: the user's last name
  - (first_name, last_name) is unique



users

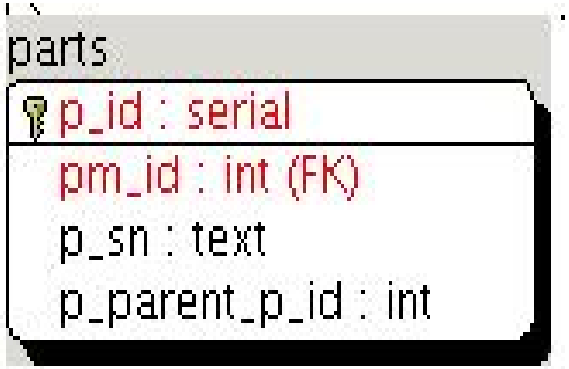
u_id : serial

first_name : text

last_name : text

Flexible data model - parts

- Tracks serialized instances of parts being measured or tested
  - p_id: primary key
  - pm_id: refers to the part_master record that this part derives from
  - p_sn: the identifying serial number for this part instance
  - p_parent_p_id: the parts p_id for the parent assembly that this part belongs to
  - (pm_id, p_sn) is unique



parts

p_id : serial

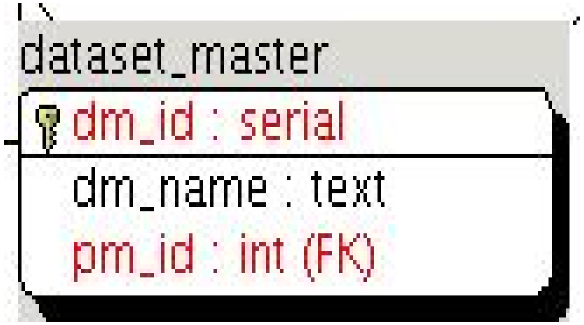
pm_id : int (FK)

p_sn : text

p_parent_p_id : int

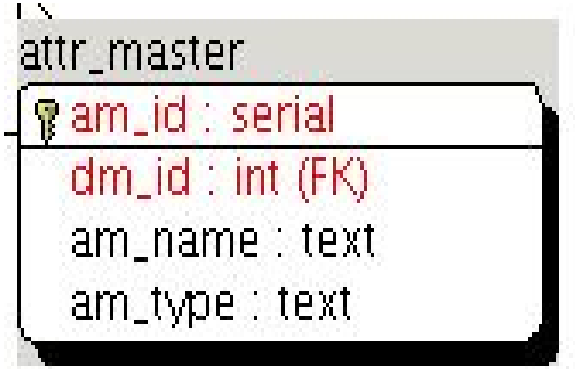
Flexible data model - dataset_master

- Master template record used to link a part number to a set of attributes that are required to be collected
  - dm_id: primary key
  - dm_name: common name for the dataset (e.g. "final inspection")
  - pm_id: reference to the part_master record for the part number requiring this dataset
  - (dm_name, pm_id) is unique



Flexible data model - attr_master

- Master template record used to describe a collected attribute
  - am_id: primary key
  - dm_id: link to the dataset template record requiring this attribute
  - am_name: the attribute name
  - am_type: the attribute data type
  - (dm_id, am_name) is unique



attr_master

am_id : serial

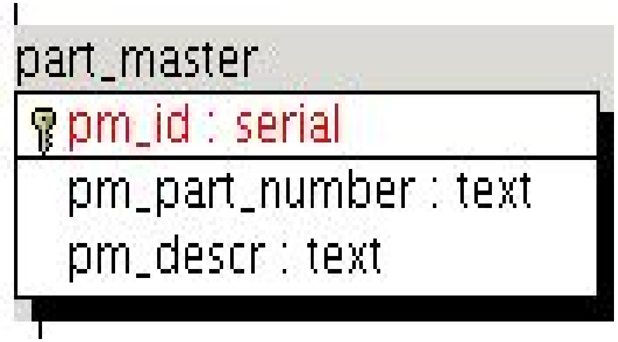
dm_id : int (FK)

am_name : text

am_type : text

Flexible data model - part_master

- Master record used to describe a part
  - pm_id: primary key
  - pm_part_number: the part number for this part
  - pm_descr: a short description of the part
  - (pm_part_number) is unique



part_master

pm_id : serial

pm_part_number : text

pm_descr : text

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Data input: flexible data model

- Setup: tables, constraints, indexes, convenience functions
- Configuration data
  - application users (users)
  - valid part numbers (part_master)
  - data collection set master data (dataset_master)
  - data collection set detail data (attr_master)
- Data collection
  - create record of specific part by SN (parts)
  - update part hierarchies
  - create data collection event record (datasets)
  - store actual attribute data (attrs)
- Sample transactions are in flex.sql
  - see www.joeconway.com after OSCON 2004

Data extraction flexible data model

- Use application code to recurse, and drill down
- Pre-materialize views of data that look like the simple model
- Dynamically link hierarchies and transpose attributes for just-in-time analysis: see contrib/tablefunc:
  - connectby()
  - crosstab()

Data extraction: connectby()

```
select p.p_id, pm.pm_part_number
from parts p
 join datasets d on p.p_id = d.p_id
 join part_master pm on p.pm_id = pm.pm_id
where d.d_dts::date = '2004-Jun-20'
and p.p_parent_p_id is null;
 p_id | pm_part_number
------+-----------------
    7 | widget
(1 row)
```

Data extraction: connectby() (cont.)

```
select * from
connectby('parts','p_id','p_parent_p_id','7',0,'~')
AS t(p_id int, p_parent_p_id int, level int, branch text);

 p_id | p_parent_p_id | level | branch
------+---------------+-------+--------
    7 |               |     0 | 7
    1 |             7 |     1 | 7~1
    3 |             7 |     1 | 7~3
    5 |             7 |     1 | 7~5
(4 rows)
```

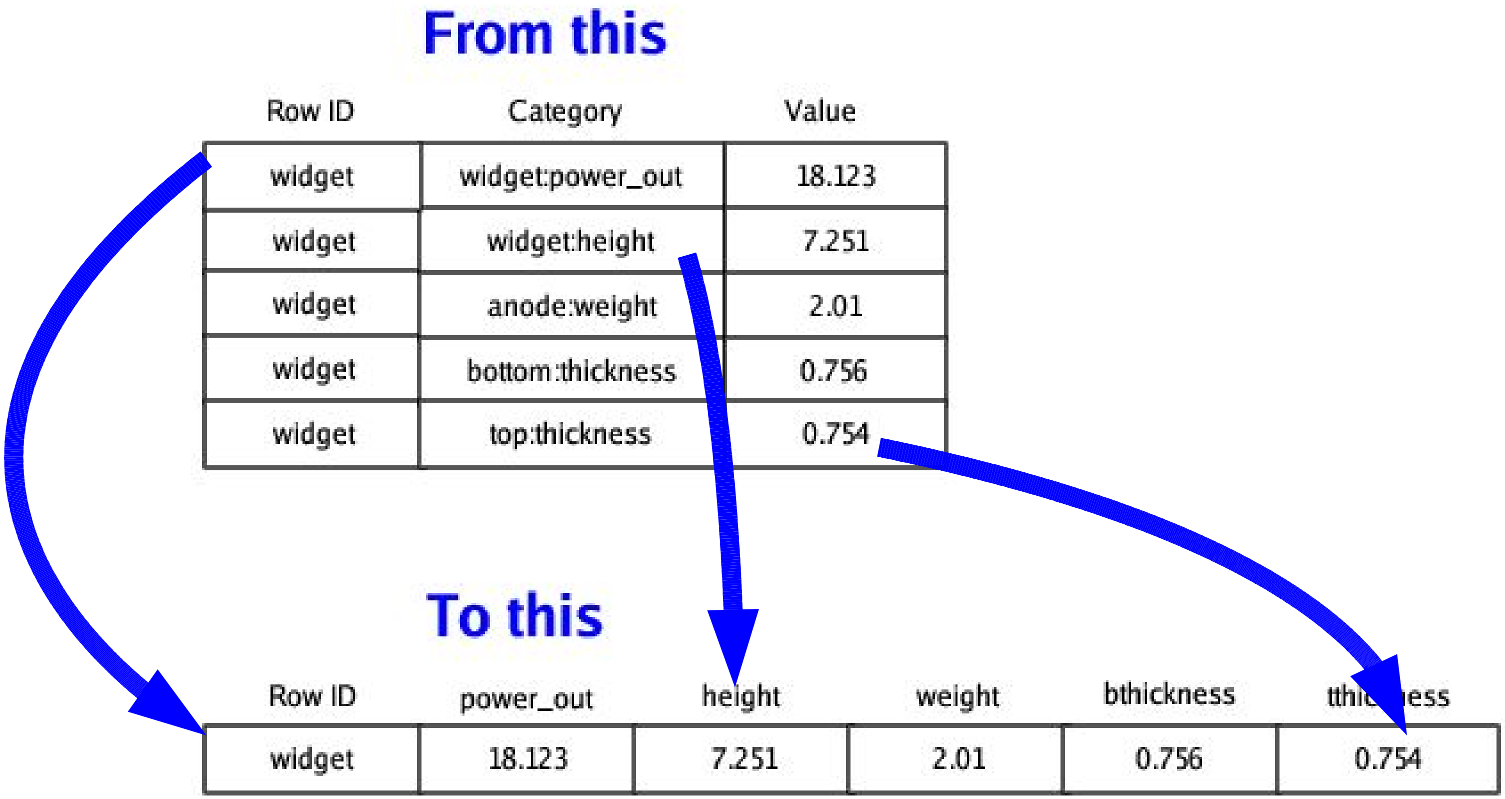
Data extraction: connectby() (cont.)

```
select
 pm.pm_part_number as pnum, p.p_sn, dm.dm_name as dset,
 u.first_name as fn, am.am_name as attr, a.a_val
from
 connectby('parts','p_id','p_parent_p_id','7',0,'~')
 AS t(p_id int, p_parent_p_id int, level int, branch text)
 join parts p on t.p_id = p.p_id
 join part_master pm on p.pm_id = pm.pm_id
 join datasets d on p.p_id = d.p_id
 join dataset_master dm on d.dm_id = dm.dm_id
 join attrs a on d.d_id = a.d_id
 join attr_master am on a.am_id = am.am_id
 join users u on d.u_id = u.u_id;
```

Data extraction: connectby() (cont.)

```
  pnum  |  p_sn  |     dset     | fn  |   attr    | a_val
--------+--------+--------------+-----+-----------+--------
 widget | wsn101 | widget attrs | Jim | power_out | 18.123
 widget | wsn101 | widget attrs | Jim | height    | 7.251
 anode  | asn101 | anode attrs  | Jim | weight    | 2.01
 bottom | bsn101 | bottom attrs | Jim | thickness | 0.756
 top    | tsn101 | top attrs    | Jim | thickness | 0.754
(5 rows)
```

Data extraction: crosstab()



From this

Row ID	Category	Value
widget	widget:power_out	18.123
widget	widget:height	7.251
widget	anode:weight	2.01
widget	bottom:thickness	0.756
widget	top:thickness	0.754

To this

Row ID	power_out	height	weight	bthickness	tthickness
widget	18.123	7.251	2.01	0.756	0.754

Data extraction: crosstab() (cont.)

- select * from crosstab(row_sql, category_sql);
- row_sql
  - produces the source rows for the crosstab
  - must have at least 3 columns (row_id, category, value)
  - column 1 is always taken as row_id; the last two columns are always taken as category and value
  - additional columns added between row_id and category are copied from first row of each row_id group into the result
  - rows must be ordered by the row_id column
- category_sql
  - must produce a single column result, containing the distinct list of categories

Data extraction: crosstab() (cont.)

```
select * from crosstab(
 'select ''widget'' as assembly, p.p_sn,
   dm.dm_name, u.first_name as user_name,
   d.d_dts, pm.pm_part_number || '':'' || am.am_name,
   a.a_val
  from connectby(''parts'',''p_id'',''p_parent_p_id'',''7'',0,''~'')
   AS t(p_id int, p_parent_p_id int, level int, branch text)
   join parts p on t.p_id = p.p_id
   join part_master pm on p.pm_id = pm.pm_id
   join datasets d on p.p_id = d.p_id
   join dataset_master dm on d.dm_id = dm.dm_id
   join attrs a on d.d_id = a.d_id
   join attr_master am on a.am_id = am.am_id
   join users u on d.u_id = u.u_id
 ',
```

Data extraction: crosstab() (cont.)

```
 '
  select pm.pm_part_number || '':'' || am.am_name
  from connectby(''parts'',''p_id'',''p_parent_p_id'',''7'',0,''~'')
   AS t(p_id int, p_parent_p_id int, level int, branch text)
   join parts p on t.p_id = p.p_id
   join part_master pm on p.pm_id = pm.pm_id
   join datasets d on p.p_id = d.p_id
   join attrs a on d.d_id = a.d_id
   join attr_master am on a.am_id = am.am_id
  group by pm.pm_part_number || '':'' || am.am_name
  order by 1
 '
) as (assembly text, sn text, dataset_name text, user_name text, dts
   timestamp, anode_weight float8, bottom_thickness float8, top_thickness
   float8, widget_height float8, widget_power_out float8);
```

Data extraction: crosstab() (cont.)

```
-[ RECORD 1 ]----+--------------------
assembly         | widget
sn               | wsn101
dataset_name     | widget attrs
user_name        | Jim
dts              | 2004-06-20 00:00:00
anode_weight     | 2.01
bottom_thickness | 0.756
top_thickness    | 0.754
widget_height    | 7.251
widget_power_out | 18.123
```

Data extraction: simple data model (review)

```
-[ RECORD 1 ]----+-----------------------
sn               | wsn101
user_name        | Jim
assembly_height  | 7.251
top_thickness    | 0.754
bottom_thickness | 0.756
anode_weight     | 2.01
power_out        | 18.123
dts              | 2004-06-19 00:00:00-07
```

Data extraction: get_widget_data()

```
CREATE OR REPLACE FUNCTION
 get_widget_data(timestamptz, timestamptz, text)
RETURNS setof record AS '
```

See flex.sql

Data extraction: get_widget_data() (cont.)

```
select * from get_widget_data(
 '2004-Jun-20',
 '2004-Jun-20',
 'assembly text, sn text, dataset_name text,
  user_name text, dts timestamp, anode_weight
  float8, bottom_thickness float8, top_thickness
  float8, widget_height float8, widget_power_out
  float8'
) as (
  assembly text, sn text, dataset_name text,
  user_name text, dts timestamp, anode_weight
  float8, bottom_thickness float8, top_thickness
  float8, widget_height float8, widget_power_out
  float8);
```

Data extraction: get_widget_data() (cont.)

```
-[ RECORD 1 ]----+--------------------
assembly         | widget
sn               | wsn101
dataset_name     | widget attrs
user_name        | Jim
dts              | 2004-06-20 00:00:00
anode_weight     | 2.01
bottom_thickness | 0.756
top_thickness    | 0.754
widget_height    | 7.251
widget_power_out | 18.123
```

Agenda

- Scenario
- Simple data model
- Data input and extraction
- Scope changes
- Flexible data model
- Data input and extraction
- Implementation example

Implementation example

- Production inspection data collection system - POD
  - Used by 8 workcenters in 2 geographical locations for about 2 years now
  - Millions of attribute values collected
  - Thousands of master data sets
  - Hundreds of part numbers
- Demo
  - Create simple data set
  - Adding an attribute to existing data set
  - Create simple report from a data set

O'REILLY®

OPEN SOURCE CONVENTION™

JULY 26–30, 2004 • PORTLAND, OR