Handling Inconsistencies in Multidimensional Data Warehouses

AMW Punta del Este, Uruguay

Monica Caniupan
mcaniupa@ubiobio.cl

Depto. Sistemas de Información
Universidad del Bío-Bío
Concepción-Chile

Joint work with Leopoldo Bertossi

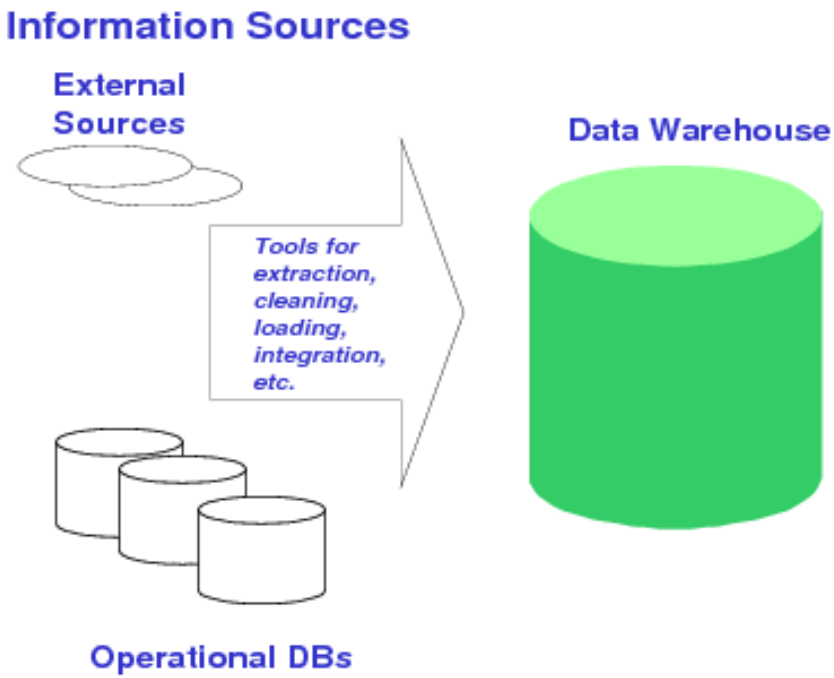
Octubre 2007

Table of contents

Multidimensional Data Warehouses

Multidimensional Data Warehouses (MDWs) are data repositories:

- Subject oriented
- Integrated
- Variable in time
- Non volatile

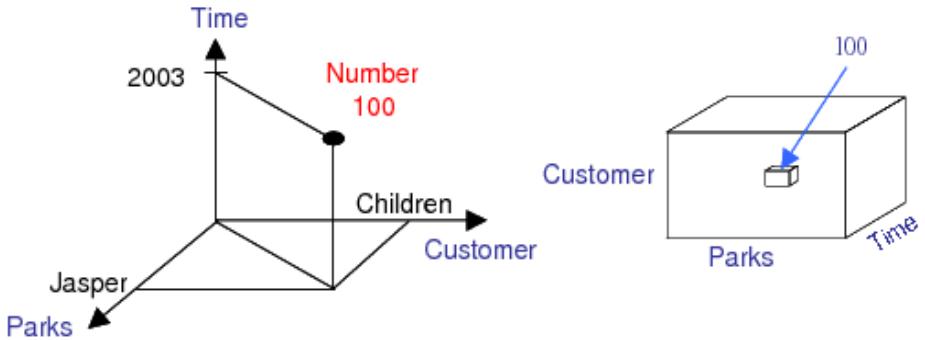


Dimensions and Facts

MDWs are queried by OLAP (On-Line Analytical Processing) systems, which require aggregation of data

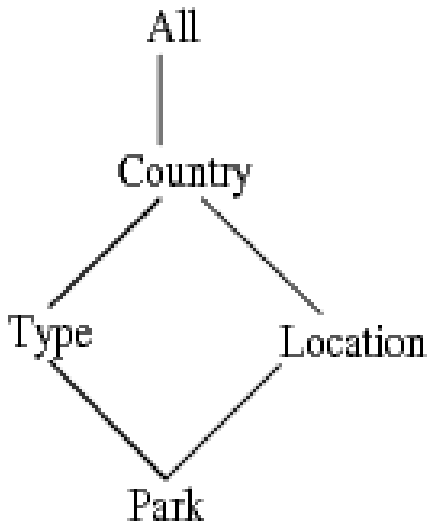
The MDWs consist mainly of dimension and facts:

- Dimensions: way in which data is organized
- Facts: numerical data related with dimensions



Hierarchical Dimension Schemas

Example: The National Parks' hierarchy $\mathcal{H} = (C, \nearrow)$:

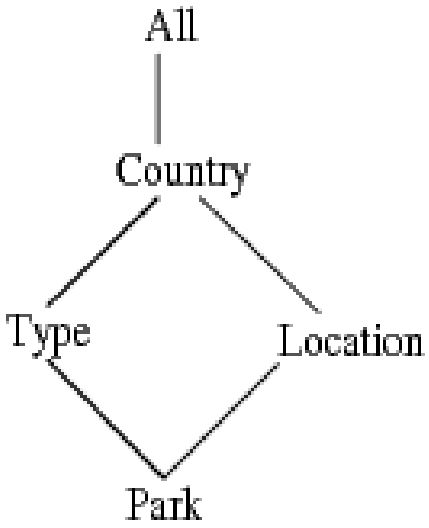


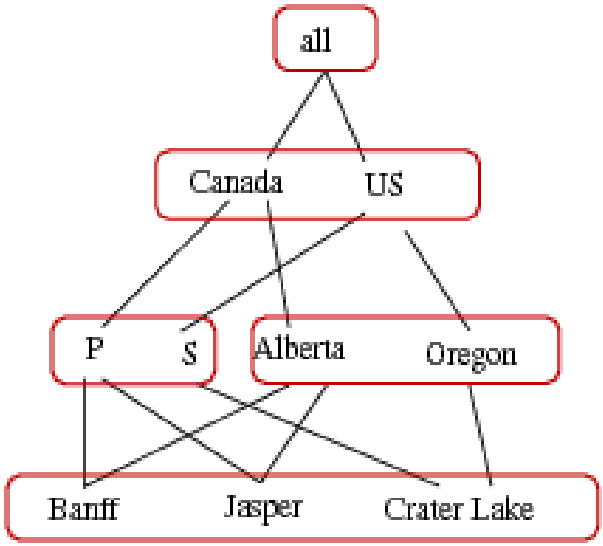
$C = \{Park, Type, Location, Country, All\}$

$\nearrow = \{(Park, Type), (Park, Location), (Type, Country), (Location, Country), (Country, All)\}$

$\nearrow^* = \nearrow \cup \{(Park, Park), (Type, Type), (Park, Country), ...\}$

Hierarchical Dimension Instances





All is the top category and *Park* is the bottom category

In an homogeneous dimension schema all the edges between categories are mandatory

A Dimension Instance as a First-Order Structure

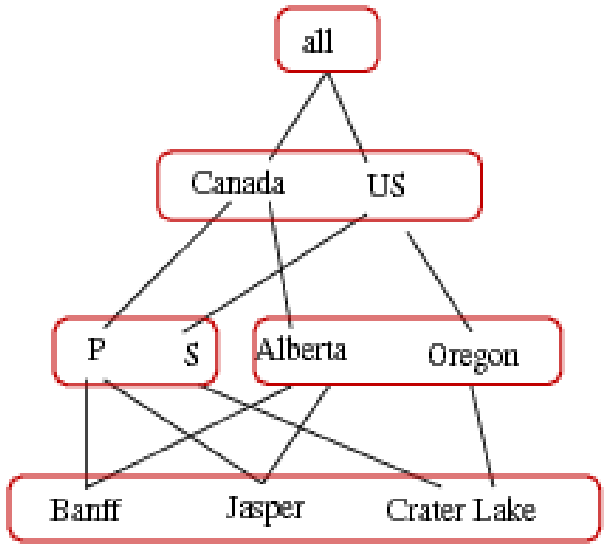
$$D = \langle \mathcal{U}, Park(\cdot)^{\mathcal{U}}, Type(\cdot)^{\mathcal{U}}, Location(\cdot)^{\mathcal{U}}, Country(\cdot)^{\mathcal{U}}, All(\cdot)^{\mathcal{U}}, all^{\mathcal{U}}, A^{\mathcal{U}}, <^{\mathcal{U}}, <^{*\mathcal{U}} \rangle$$

- $Park^{\mathcal{U}} = \{Banff, Jasper, CraterLake\}$, $Type^{\mathcal{U}} = \{P, S\}$
- $Location^{\mathcal{U}} = \{Alberta, Oregon\}$, $Country^{\mathcal{U}} = \{Canada, US\}$
- $All^{\mathcal{U}} = \{all^{\mathcal{U}}\}$
- $A^{\mathcal{U}} = \{p_{Park,Type}, p_{Park,Location}, p_{Type,Country}, p_{Location,Country}, p_{Country,All}\}$
- $<^{\mathcal{U}} = \{(Banff, P), (Banff, Alberta), (Alberta, Canada), (Canada, All), \ldots\}$
- $<^{*\mathcal{U}} = <^{\mathcal{U}} \cup \{(Banff, Banff), (Banff, Canada) \ldots\}$

Roll-up Functions

Mapping of relation $<^*$ between elements of two fixed categories:

$$\mathcal{R}_{C_i}^{C_j}(D) := \{(x, y) \mid C_i^{\mathcal{U}}(x) \wedge C_j^{\mathcal{U}}(y) \wedge x <^* y\}$$



$R_{Park}^{Type}(D) = \{(Banff, P), (Jasper, P), (CraterLake, S)\}$,
$R_{Park}^{Country}(D) = \{(Banff, Canada), (Jasper, Canada), (CraterLake, US)\}$

Fundamental for computing aggregation of data

Dimension Constraints

The following $\mathcal{L}$-sentences are also dimension constraints:

- Path Constraints: $\forall x(Park(x) \rightarrow \exists y(Type(y) \wedge x < y))$
- Equality Constraints:$\exists x(Country(x) \wedge x = Canada)$
- Partitioning Constraints (PCs): For two categories C_i and C_j a PC is:

$$\forall xyz(C_i(x) \wedge C_j(y) \wedge C_j(z) \wedge x <^* y \wedge x <^* z \rightarrow y = z)$$

PCs enforce that roll-up functions are functional

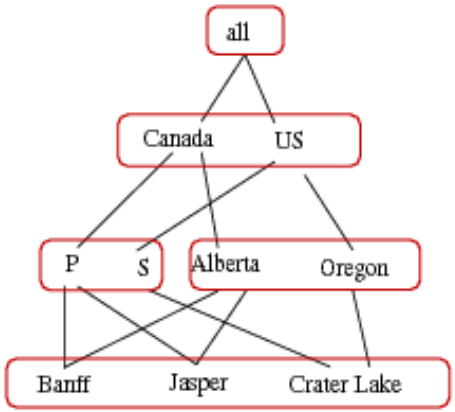
Summarizability

A category C_i is summarizable from a category C_j in a dimension instance D of a hierarchy schema $\mathcal{H}$ iff:

- $C_j \nearrow^* C_i$
- $R^{C_i}_{Cbottom}(D) = R^{C_j}_{Cbottom}(D) \bowtie R^{C_i}_{C_j}(D)$

Consistent Dimension Instances

$$\forall xyz(Park(x) \wedge Country(y) \wedge Country(z) \wedge x <^* y \wedge x <^* z \rightarrow y = z)$$



$R^{Country}_{Park}(D) = \{(Banff, Canada), (Jasper, Canada), (CraterLake, US)\}$
can be computed using the category *Type*, since:

$$R^{Country}_{Park}(D) = R^{Type}_{Park}(D) \bowtie R^{Country}_{Type}(D)$$

Aggregate Queries

We need numerical data, i.e. *facts*, which are stored in fact tables, and roll-up functions (treated as tables)

Example: $\mathcal{Q}$: Number of visitors grouped by country?

$V(Park, Number) = \{(Banff, 5000), (Jasper, 5000), (CraterLake, 10000)\}$

$R^{Country}_{Park}(D) = \{(Banff, Canada), (Jasper, Canada), (CraterLake, US)\}$

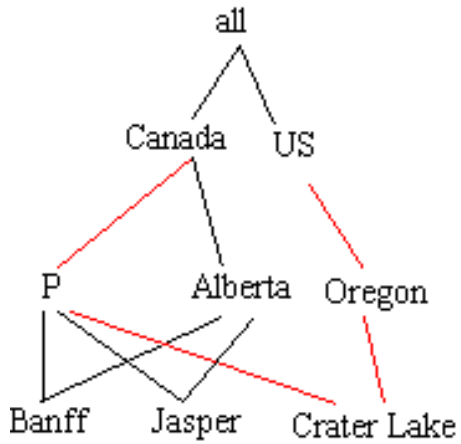
```
SELECT R.Country, SUM(V.Number)
FROM V, R^Country_Park R
WHERE V.Park = R.Park
GROUP BY R.Country
```

The answers to $\mathcal{Q}$ are: $(Canada, 10000), (USA, 10000)$

Inconsistent Dimension Instances

The following dimension instance is inconsistent wrt the PC:

$$\forall xyz(Park(x) \wedge Country(y) \wedge Country(z) \wedge x <^* y \wedge x <^* z \rightarrow y = z)$$



$R^{Country}_{Park} = \{(Banff, Canada), (Jasper, Canada),$
$(CraterLake, Canada), (CraterLake, US)\}$ is not functional

Consequences on Aggregate Queries

Consider the fact table and roll-up function:

$$V(Park, Number) = \{(Banff, 5000), (Jasper, 5000), (CraterLake, 10000)\}$$

$$R^{Country}_{Park} = \{(Banff, Canada), (Jasper, Canada), (CraterLake, Canada), (CraterLake, US)\}$$

For the aggregate query:

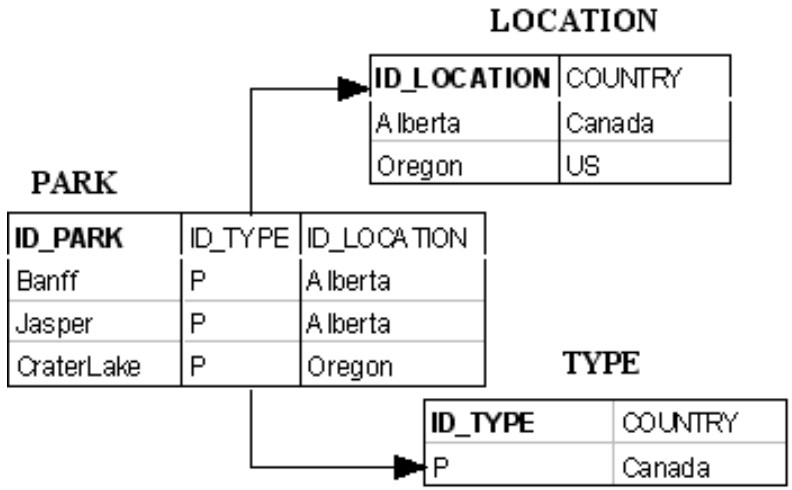
```
SELECT R.Country, SUM(V.Number)
FROM Visitors V, R^Country_Park R
WHERE V.Park = R.Park
GROUP BY R.Country
```

The answers are: $(Canada, 20000), (US, 10000)$

Visitors to *CraterLake* are added to *Canada* and also to *US*

Restoring Consistency

The notion of relational database repair (ABC,PODS99) does not capture the minimality required by MDWs



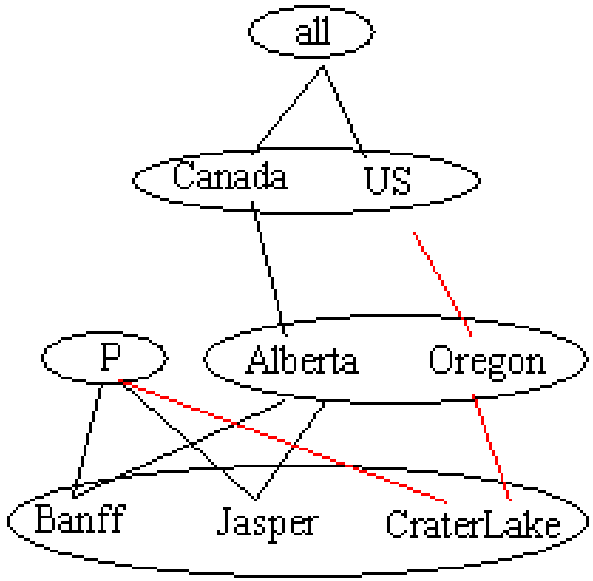
$\forall xyzwv(Park(x,y,z) \wedge Type(y,w) \wedge Location(z,v) \rightarrow w = v)$

Consistency can be restored by eliminating tuples:

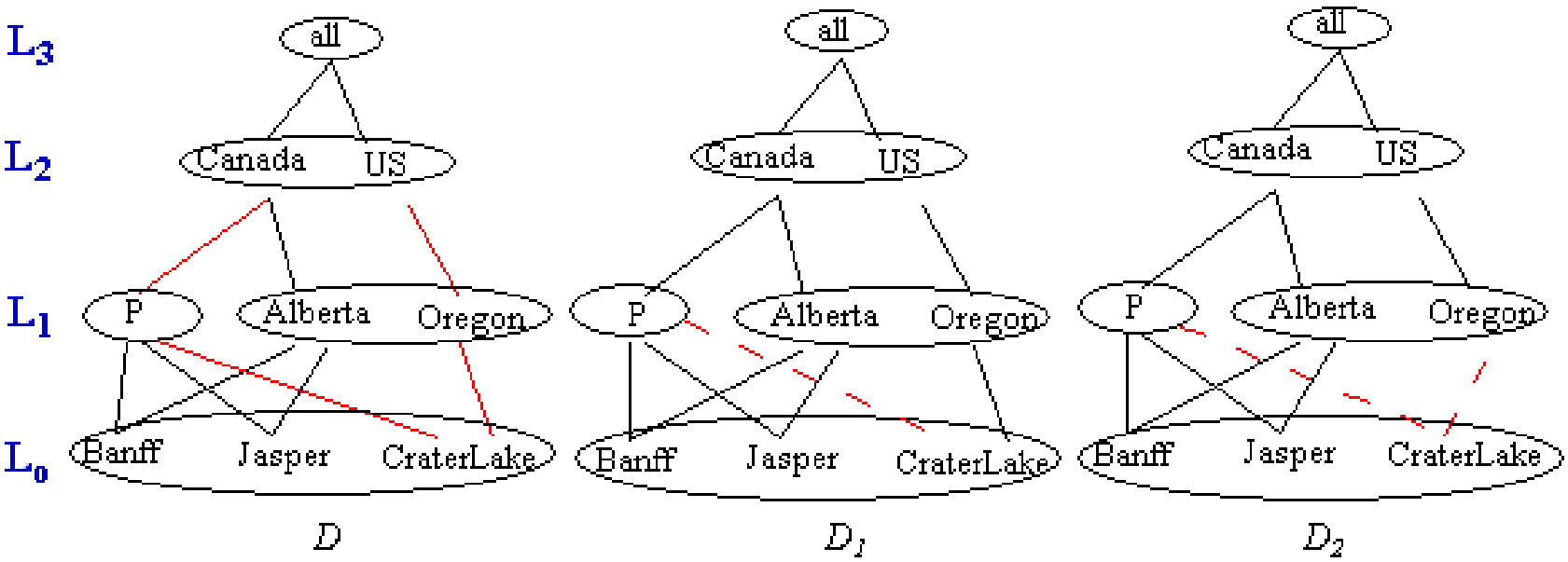
$Park(CraterLake, P, Oregon)$ or $Location(Oregon, US)$ or $Type(P, Canada)$

Restoring Consistency

- To restore consistency we should eliminate the minimum number of edges in dimension instances
- The repairs should not introduce new inconsistencies



Identification of Inconsistencies and Minimality



$R_0(D) = \{(Banff, P), (Banff, Alberta), (Jasper, P), (Jasper, Alberta), (CraterLake, P), (CraterLake, Oregon)\}$

$\Delta_0(\mathcal{D}, \mathcal{D}_1) = \{(CraterLake, P)\}$

$\Delta_0(\mathcal{D}, \mathcal{D}_2) = \{(CraterLake, P), (CraterLake, Oregon)\}$

$\mathcal{D}_1 \leq_{\mathcal{D},0} \mathcal{D}_2$: $\Delta_0(\mathcal{D}, \mathcal{D}_1) \subseteq \Delta_0(\mathcal{D}, \mathcal{D}_2)$

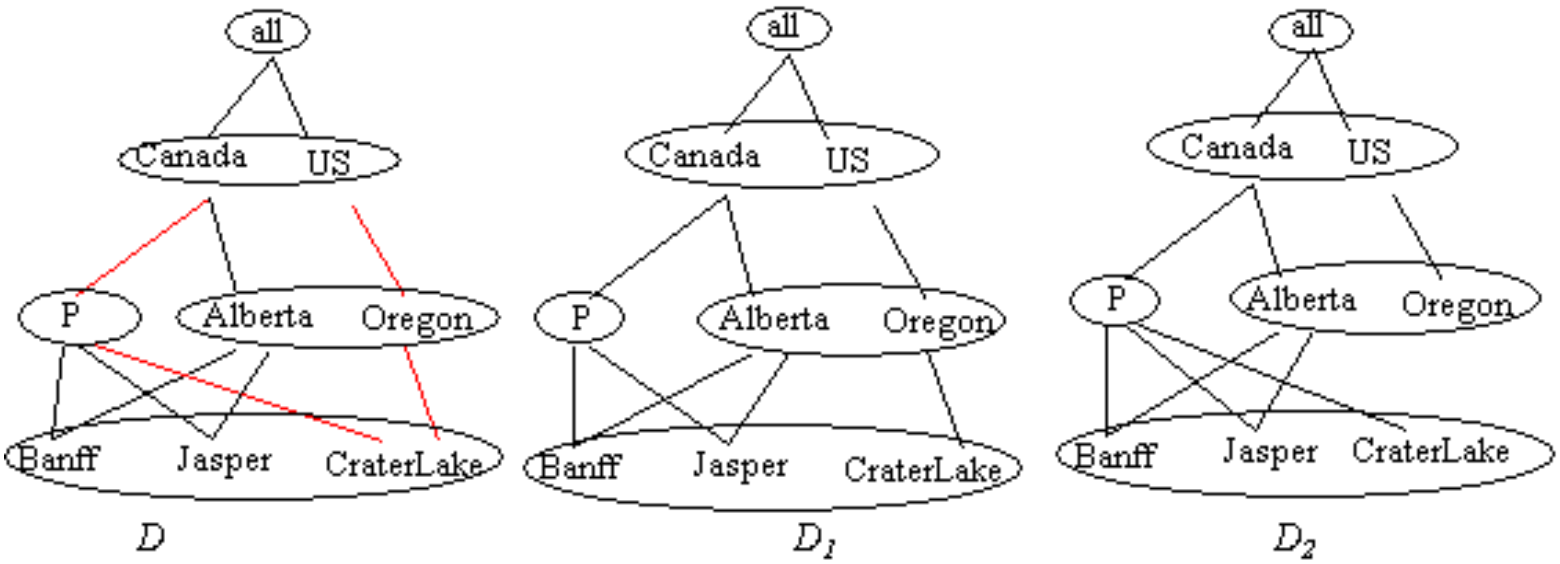
Repairs of Dimension Instances

For a dimension instance D with schema $\mathcal{H}$, and set of partitioning constraints PC, a repair of D wrt PC is:

- A dimension instance D' over the same dimension schema $\mathcal{H}$ and domain $\mathcal{U}$
- $D' \models PC$
- D' is $\leq_D$-minimal in the class of dimension instances that satisfy PC

  $D' \leq_D D''$ iff there exists an i such that for $k < i$, $\Delta_k(D, D') = \Delta_k(D, D'')$, and $D' \leq_{D,i} D''$

The Dimension Instances Repairs



Dimension instances repairs are not homogeneous instances anymore!

Consistent Answers to Aggregate Queries

$V(Park, Number) = \{(Banff, 5000), (Jasper, 5000), (CraterLake, 10000)\}$

$R^{Country}_{Park} = \{(Banff, Canada), (Jasper, Canada), (CraterLake, US)\}$

$R^{Country}_{Park} = \{(Banff, Canada), (Jasper, Canada), (CraterLake, Canada)\}$

For the aggregate query Q:

```
SELECT R.Country, SUM(V.Number)
FROM Visitors V, R^Country_Park R
WHERE V.Park = R.Park
GROUP BY R.Country
```

D' : $\{(Canada, 10000), (US, 10000)\}$, D'': $\{(Canada, 20000)\}$

The consistent answer to Q is $(Canada, [10000, 20000])$

Conclusions

- A repair semantics for homogeneous instances of MDWs, which captures the minimality required in MDWs
- The notion of repair can be improved by using knowledge from other dimension constraints, e.g. those that impose the existence of certain elements in the categories
- Future work:
  - Develop of a methodology for computing repairs (if necessary), and consistent answers in MDWs
  - Extension to heterogeneous dimension schemas
  - Consistent Query Answering wrt aggregation constraints