Database Project
Free University of Bolzano

# Project Title: "Snowstop DB"

Student: Samuel Dalvai

01/09/2021

## Contents

# 1 Conceptual design

## 1.1 Structured requirements

**General statements:**
We want to realize a database for a retail company producing products for the retainment of snow on roofs. We want to represent the different products and provide the possibility to produce snowload computations that are based on the amount of snow on a given roof and are linked to a supply offer for a customer. The supply offers will be tailored around the values computed in the snowload computation, and the right type of product to be offered will be chosen accordingly.

**Statements concerning snowstop products:**
For the snowstop products, which are identified by a code, we are interested in the name and the retail price. The type of material is also of interest, this can be zink steel, stainless steel, painted steel, aluminium or copper. In the case of painted steel a color can also be specified, this attribute is therefore in some cases optional. In addition each snowstop product has to be one of three types: snow retainer, retainer holder or retainer accessory.

**Statements concerning snow retainers:**
For the snow retainers, which are a snowstop product, we are also interested in the linear resistance. In addition, each snow retainer can be one of two types: grid retainer, tube retainer. For the grid retainers we are also interested in the height and the profile of the grid, for the tube retainers we are also interested in the diameter.

**Statements concerning retainer holders:**
For the retainer holders we also want to represent the resistance, but in contrast to the snow retainers, rather than the resistance per meter we express the maximum load that a single element can withstand. In addition, we are also interested in the type of roof to which the system is compatible. There are five possible types of roofs, namely 'concrete tile', 'ondulated plate', 'trapezoidal sheet', 'standing seam sheet' or 'flat tile'. In addition, each RetainerHolder is compatible with one or more snow retainers.

**Statements concerning retainer accessories:**
For the retainer accessories we are interested in the measure and in the type, which can be either "connection" or "ice retainer". In addition, each accessory is compatible with one or more snow retainers.

**Statements concerning supply offers:**
For the supply offers we are interested in the code, which uniquely identifies them, the total price and the date in which they have been formulated, the total price depends on the quantity and the price of the associated snowstop products. In addition, a supply offer is made for one or more customers, since different customers could be competing for the same bidding. A supply offer must consist of the following snowstop products:

- One type of snow retainer
- One type of retainer holder
- Between zero and two types of accessories

During the formulation of a supply offer, the compatibility of the retainer holder and of the

accessories with the snow retainer must be ensured. In addition a supply offer must always be associated to one snowload computation.

**Statements concerning customers:**
For the customers we are interested in the code that uniquely identifies them, the name, the phone number (customers can have more than one phone number) and the discount reserved to them (from 0 to 30). In addition each customer is located in a specific city.

**Statements concerning snowload computations:**
For the snowload computation we are interested in the code that uniquely identifies it, the date in which it has been formulated, the computed ground load and the roof load. In addition each snowload computation is always associated to a specific building site and to zero or more supply offers, in the latter case the total resistance of the system as well as the number of rows needed and the maximum distance of the retainer holder are of interest. The total resistance in the association between supply offers and snowload computation must be higher or equal to the computed roof load.

**Statements concerning building sites:**
For the building sites we are interested in the name, that uniquely identifies them within the city in which they are located, and in the roof type, which is a composite attribute that comprises length, width, steepness and covering type, the possible roof types are the same listed for the retainer holder. The steeepness and the length of the roof, as well as the altitude and the zone in which it is located will affect the value of the snowload computation. In addition, each building site is located in a specific city and has exactly one snowload computation associated to it. Furthermore, the roof type of the retainer holder listed in a supply offer must be the same as the covering type of the building site associated to it through the snowload computation.

**Statements concerning cities:**
For the cities we are interested in the zip code that uniquely identifies them together with the name (some cities might have the same name). Also the altitude is of interest, specifically for the snow load computation. In addition each city is located in one specific province.

**Statements concerning provinces:**
For the provinces we are interested in the name , the shorthand made of two letters that uniquely identifies them, the climatic zone and the base snow fall. The last two parameters influence the amount of snow falling. There are 4 climatic zones with different coefficients for the computation of the snow load.

## 1.2 Glossary of terms

| Term | Description | Synonyms | Connections |
|---|---|---|---|
| **Snowstop product** | An article that is manufactured for sale, designed for the retainment of snow on the roof | Commodity | Supply offer |
| **Snow retainer** | A metal barrier to prevent snow from falling off the roof | Snow stopper, Snow barrier | Retainer holder, Retainer accessory |
| **Grid retainer** | A grid formed metal barrier to prevent snow from falling off the roof | Grid barrier | |
| **Tube retainer** | Same as Grid retainer but in the form of a tube | Tube barrier | |
| **Retainer holder** | A metal device that supports a Snow retainer and is anchored to the roof structure | Retainer support, Hook | Snow retainer |
| **Retainer accessory** | A metal element that can connect two pieces of Snow retainer or that can be installed to prevent the sliding of ice under the retainer | Junction, Ice retainer | Snow retainer |
| **Supply offer** | A list of Snowstop products with a price that is offered to a customer | Bid | Snowstop product, Snowload computation |
| **Customer** | A company or an enterprise that buys commodities from a retailer | Client, Consumer | City, Supply offer |
| **Snowload computation** | A mathematical computation that represents the average load of snow on a roof | Snow load, Snow estimation | Building site, Supply offer |
| **Building site** | The roof on which a snow retaining system has to be installed | Roof, Building | City, Snowload computation |
| **City** | A city located in Italy | Municipality, Location | Building site, Province |
| **Province** | A geographical area that comprises a group of municipalites | Department, District | City |

## 1.3 Conceptual schema

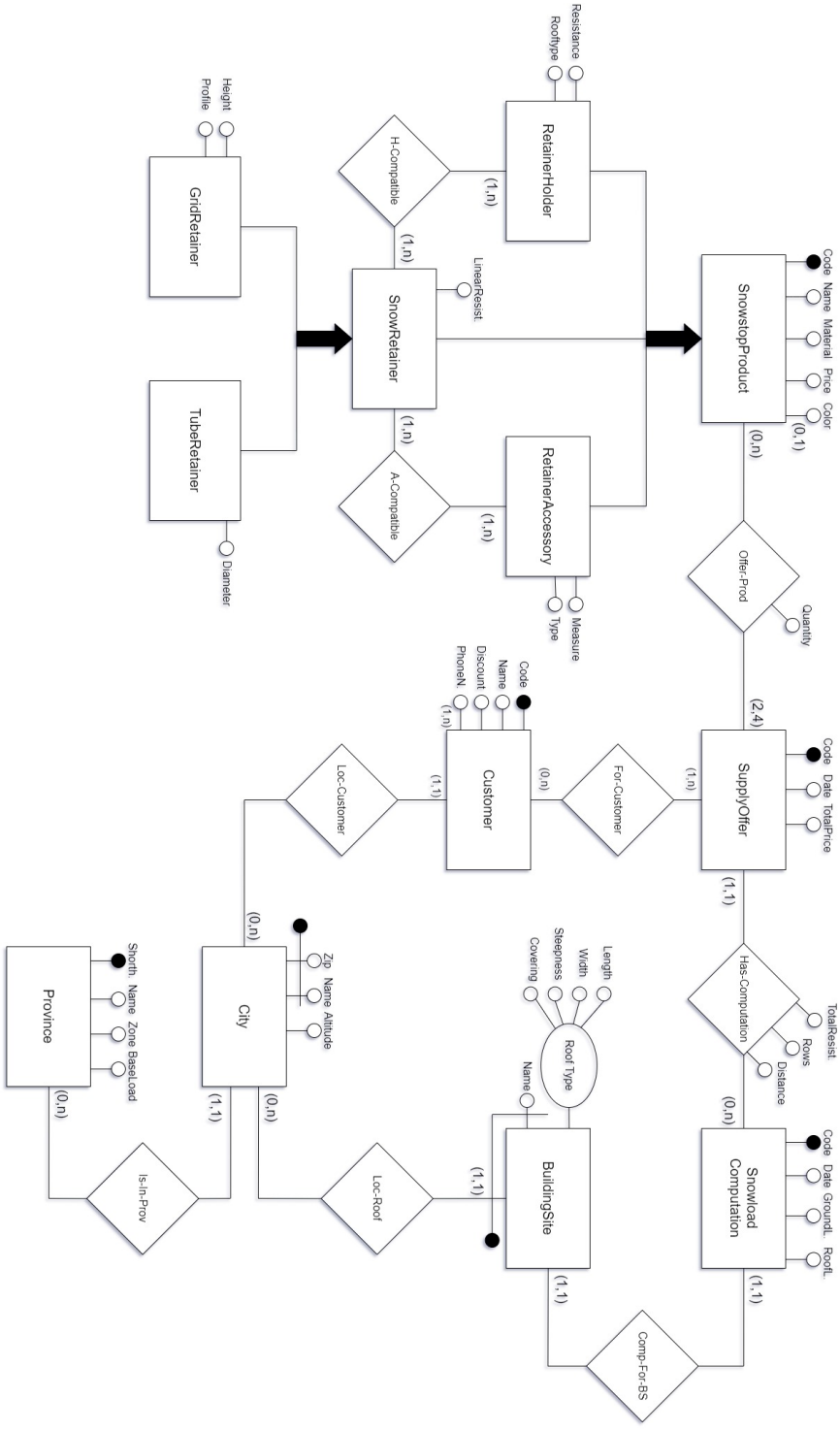


Figure 1: Conceptual schema

## 1.4 Data dictionary

**Data dictionary: Entities**

| Entity | Description | Attributes | Identifiers |
|---|---|---|---|
| **SnowstopProduct** | Snow retaining product for roofs | Code<br>Name<br>Material<br>Price<br>Color | {Code} |
| **SnowRetainer** | Retainer of snow on the roof | LinearResistance | {Code} |
| **GridRetainer** | Retainer in the form of a grid | Height<br>Profile | {Code} |
| **TubeRetainer** | Retainer in the form of a tube | Diameter | {Code} |
| **RetainerHolder** | Support for a retainer | Resistance<br>Rooftype | {Code} |
| **RetainerAccessory** | Accessory for a retainer | Measure<br>Type | {Code} |
| **SupplyOffer** | Snowstop products offered to customer | Code<br>Date<br>TotalPrice | {Code} |
| **Customer** | Company that is offered products to | Code<br>Name<br>PhoneNumber<br>Discount | {Code} |
| **SnowloadComputation** | Computation of the amount of snow load on bulding site | Code<br>Date<br>GroundLoad<br>RoofLoad | {Code} |
| **BuildingSite** | A roof for which a supply offer is made | Name<br>RoofType(Length, Width, Steepness, Covering) | {Name, City} |
| **City** | A municipality of interest for customers and snow fall | Zip<br>Name<br>Altitude | {Zip,Name} |
| **Province** | The province in which a city is located | Shorthand<br>Name<br>Zone<br>BaseLoad | {Shorthand} |

**Data dictionary: Relationships**

| Relationship | Description | Components | Attributes | Identifiers |
|---|---|---|---|---|
| **H-Compatible** | Compatibility between retainer and holder | RetainerHolder, SnowRetainer | | |
| **A-Compatible** | Compatibility between retainer and accessory | RetainerAccessory, SnowRetainer | | |
| **Offer-Prod** | Products in a supply offer | SnowstopProduct, SupplyOffer | Quantity | |
| **For-Customer** | Supply offer for a customer | SupplyOffer, Customer | | |
| **Loc-Customer** | Customer located in city | Customer, City | | |
| **Has-Computation** | Computation associated to SupplyOffer | SnowloadComputation, SupplyOffer | TotalResistance, Rows, Distance | |
| **Comp-For-BS** | Computation for a building site | SnowloadComputation, BuildingSite | | |
| **Loc-Roof** | Building site located in city | BuildingSite, City | | |
| **Is-In-Prov** | City located in province | City, Province | | |

**Data dictionary: External constraints**

| External integrity constraints | |
|---|---|
| **1** | The total resistance of the system in a supply offer must be higher or equal to the roof load of its associated snowload computation |
| **2** | The retainer holder and accessories listed in a supply offer must be compatible with the type of snow retainer listed in the same offer |
| **3** | The roof type of a retainer holder listed in a supply offer must be the same as the covering type of the building site associated to it through the snowload computation. |
| **4** | A supply offer has to be associated with exactly one retainer holder, one snow retainer and between zero and two different types of accessory |
| **5** | The total price of a supply offer must be equal to the sum of the prices of the associated snowstop products multiplied by their price |

## 1.5 Table of volumes and operations

**Table of Volumes**

| Concept | Construct | Volume |
|---|---|---|
| **SnowstopProduct** | Entity | 500 |
| **SnowRetainer** | Entity | 50 |
| **GridRetainer** | Entity | 25 |
| **TubeRetainer** | Entity | 25 |
| **RetainerHolder** | Entity | 400 |
| **RetainerAccessory** | Entity | 50 |
| **SupplyOffer** | Entity | 5000 |
| **Customer** | Entity | 2500 |
| **SnowloadComputation** | Entity | 4000 |
| **BuildingSite** | Entity | 4000 |
| **City** | Entity | 8000 |
| **Province** | Entity | 100 |
| **H-Compatible** | Relationship | 10000* |
| **A-Compatible** | Relationship | 1250* |
| **Offer-Prod** | Relationship | 15000 |
| **For-Customer** | Relationship | 6000 |
| **Loc-Customer** | Relationship | 2500 |
| **Has-Computation** | Relationship | 5000 |
| **Comp-For-BS** | Relationship | 4000 |
| **Loc-Roof** | Relationship | 4000 |
| **Is-In-Prov** | Relationship | 8000 |

* Each RetainerHolder is on average compatible with half of the SnowRetainer, same reasoning applies to the RetainerAccessory.

**Operations of interest:**

1. Insert a new customer.
2. Insert a new snowstop product, defining also the type and the compatibility.
3. Insert a new city.
4. Create a snowload computation on a given building site.
5. Create a supply offer.
6. List all the supply offers made for a given customer.
7. Update prices of snowstop products.
8. Update zip code and name of cities.

**Table of Operations**

| Operation | Type | Frequency |
|---|---|---|
| 1 | Interactive | 20/day |
| 2 | Interactive | 10/month |
| 3 | Interactive | 5/day |
| 4 | Interactive | 40/day |
| 5 | Interactive | 50/day |
| 6 | Batch | 10/week |
| 7 | Interactive | 2/year |
| 8 | Batch | 1/month |

# 2 Restructuring of the conceptual schema

## 2.1 Restructured conceptual schema

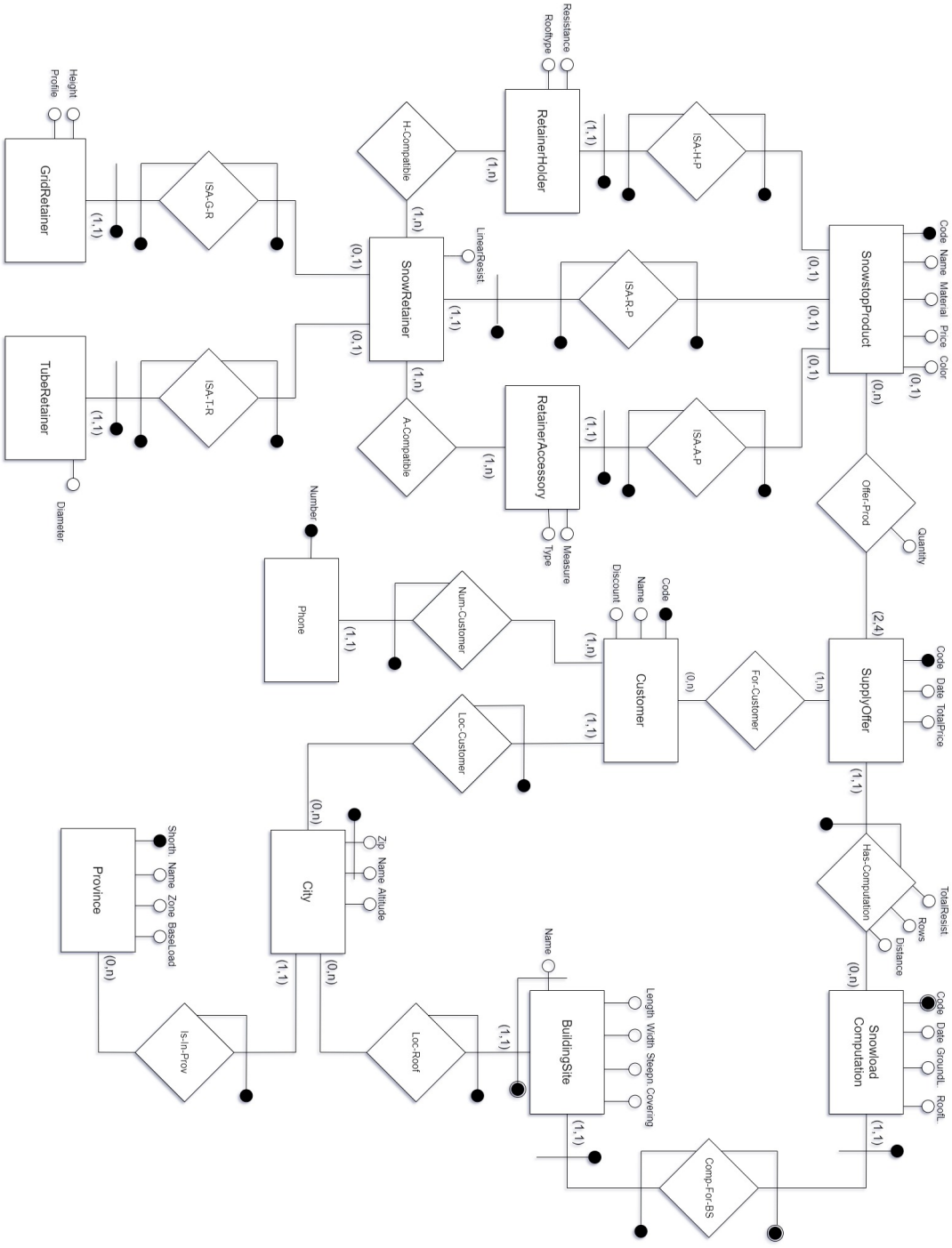


Figure 2: Restructured Conceptual schema

**Notes on the restructured conceptual schema diagram (2.1):**

- **Redundancy analysis:** The only apparent redundancy in the schema is the TotalPrice attribute in the SupplyOffer entity, which depends on the values of the Price attribute in the SnowstopProduct and the Quantity attribute in the Offer-Prod relationship. In this case we decide to keep the redundancy and express it through an external constraint, otherwise the schema would become less readable.
- **Elimination of multi-valued attributes:** We transform the Phone attribute of the Customer entity into an entity associated to Customer through the new relationship Num-Customer, adding the appropriate cardinalities.
- **Elimination of composite attributes:** In order to simplify the schema and reduce the load on operations, instead of creating a Roof Type entity to express the composite attribute, we incorporate the different attributes directly into the BuildingSite entity.
- **Elimination of ISA and generalization between entities:** We replace the complete generalizations on SnowstopProduct and SnowRetainer by the corresponding binary relations and by specifying the correct cardinaltites on the new relations. Additional constraint will need to be added in order to make sure that the generalization is complete.
- **Elimination of ISA and generalizations between relations:** No generalizations between relations were present in the first schema, nothing has to be done.
- **Choice of the primary identifiers of entities:** No apparent identification cycle was present in the first schema, thus we do not need to break one. For the entities having more than one identifier we give precedence to the internal identifiers where possible.
- **Choice of the primary identifiers of relationships:** Where we have to choose between primary identifiers for relationships we choose as primary identifier the simpler of the two entities. For example in the relation Comp-For-BS we choose the participation of the SnowloadComputation entity as primary identifier, since that specific entity, compared to the BuildingSite entity, has a primary identifier which does not involve any participation to a relation.

## 2.2 Restructured data dictionary

**Restructured data dictionary: Entities**

| Entity | Description | Attributes | Identifiers |
|---|---|---|---|
| **SnowstopProduct** | Snow retaining product for roofs | Code<br>Name<br>Material<br>Price<br>Color | {Code} |
| **SnowRetainer** | Retainer of snow on the roof | LinearResistance | {Code} |
| **GridRetainer** | Retainer in the form of a grid | Height<br>Profile | {Code} |
| **TubeRetainer** | Retainer in the form of a tube | Diameter | {Code} |
| **RetainerHolder** | Support for a retainer | Resistance<br>Rooftype | {Code} |
| **RetainerAccessory** | Accessory for a retainer | Measure<br>Type | {Code} |
| **SupplyOffer** | Snowstop products offered to customer | Code<br>Date<br>TotalPrice | {Code} |
| **Customer** | Company that is offered products to | Code<br>Name<br>Discount | {Code} |
| **Phone** | Office or mobile phone number | Number | {Number} |
| **SnowloadComputation** | Computation of the amount of snow load on bulding site | Code<br>Date<br>GroundLoad<br>RoofLoad | {Code} |
| **BuildingSite** | A roof for which a supply offer is made | Name<br>Length<br>Width<br>Steepness<br>Covering | {Name, City} |
| **City** | A municipality of interest for customers and snow fall | Zip<br>Name<br>Altitude | {Zip,Name} |
| **Province** | The province in which a city is located | Shorthand<br>Name<br>Zone<br>BaseLoad | {Shorthand} |

**Restructured data dictionary: Relationships**

| Relationship | Description | Components | Attributes | Identifiers |
|---|---|---|---|---|
| **H-Compatible** | Compatibility between retainer and holder | RetainerHolder, SnowRetainer | | {Code,Code} |
| **A-Compatible** | Compatibility between retainer and accessory | RetainerAccessory, SnowRetainer | | {Code,Code} |
| **Offer-Prod** | Products in a supply offer | SnowstopProduct, SupplyOffer | Quantity | {Code,Code} |
| **For-Customer** | Supply offer for a customer | SupplyOffer, Customer | | {Code,Code} |
| **Loc-Customer** | Customer located in city | Customer, City | | {Code} |
| **Num-Customer** | Phone number of customer | Customer, Phone | | {Num} |
| **Has-Computation** | Computation associated to SupplyOffer | SnowloadComputation, SupplyOffer | TotalResistance, Rows, Distance | {Code} |
| **Comp-For-BS** | Computation for a building site | SnowloadComputation, BuildingSite | | {Code} |
| **Loc-Roof** | Building site located in city | BuildingSite, City | | {Name, City} |
| **Is-In-Prov** | City located in province | City, Province | | {Zip, Name} |
| **ISA-H-P** | Is a retainer holder snowstop product | SnowstopProduct, RetainerHolder | | {Code} |
| **ISA-R-P** | Is a snow retainer snowstop product | SnowstopProduct, SnowRetainer | | {Code} |
| **ISA-A-P** | Is a retainer accessory snowstop product | SnowstopProduct, RetainerAccessory | | {Code} |
| **ISA-G-R** | Is a grid snow retainer | SnowRetainer, GridRetainer | | {Code} |
| **ISA-T-R** | Is a tube snow retainer | SnowRetainer, TubeRetainer | | {Code} |

**Restructured data dictionary: External constraints**

| External integrity constraints | |
|---|---|
| **1** | The total resistance of the system in a supply offer must be higher or equal to the roof load of its associated snowload computation |
| **2** | The retainer holder and accessories listed in a supply offer must be compatible with the type of snow retainer listed in the same offer |
| **3** | The roof type of a retainer holder listed in a supply offer must be the same as the covering type of the building site associated to it through the snowload computation. |
| **4** | A supply offer has to be associated with exactly one retainer holder, one snow retainer and between zero and two different types of accessory |
| **5** | The total price of a supply offer must be equal to the sum of the prices of the associated snowstop products multiplied by their price |
| **6** | Each instance of SnowstopProduct must participate either to ISA-H-P, ISA-R-P or ISA-A-P, but not to more than one of them |
| **7** | Each instance of SnowRetainer must participate either to ISA-G-R or ISA-A-P, but not to both of them |

## 2.3 Restructured table of volumes and operations

**Restructured table of volumes**

| Concept | Construct | Volume |
|---|---|---|
| **SnowstopProduct** | Entity | 500 |
| **SnowRetainer** | Entity | 50 |
| **GridRetainer** | Entity | 25 |
| **TubeRetainer** | Entity | 25 |
| **RetainerHolder** | Entity | 400 |
| **RetainerAccessory** | Entity | 50 |
| **SupplyOffer** | Entity | 5000 |
| **Customer** | Entity | 2500 |
| **Phone** | Entity | 3750 |
| **SnowloadComputation** | Entity | 4000 |
| **BuildingSite** | Entity | 4000 |
| **City** | Entity | 8000 |
| **Province** | Entity | 100 |
| **H-Compatible** | Relationship | 10000* |
| **A-Compatible** | Relationship | 1250* |
| **Offer-Prod** | Relationship | 15000 |
| **For-Customer** | Relationship | 6000 |
| **Loc-Customer** | Relationship | 2500 |
| **Num-Customer** | Relationship | 3750 |
| **Has-Computation** | Relationship | 5000 |
| **Comp-For-BS** | Relationship | 4000 |
| **Loc-Roof** | Relationship | 4000 |
| **Is-In-Prov** | Relationship | 8000 |
| **ISA-H-P** | Relationship | 400 |
| **ISA-R-P** | Relationship | 50 |
| **ISA-A-P** | Relationship | 50 |
| **ISA-G-R** | Relationship | 25 |
| **ISA-T-R** | Relationship | 25 |

* Each RetainerHolder is on average compatible with half of the SnowRetainer, same reasoning applies to the RetainerAccessory.

**Operations of interest:**

1. Insert a new customer.
2. Insert a new snowstop product, defining also the type and the compatibility.
3. Insert a new city.
4. Create a snowload computation on a given building site.
5. Create a supply offer.
6. List all the supply offers made for a given customer.
7. Update prices of snowstop products.
8. Update zip code and name of cities.

**Table of Operations**

| Operation | Type | Frequency |
|---|---|---|
| 1 | Interactive | 20/day |
| 2 | Interactive | 10/month |
| 3 | Interactive | 5/day |
| 4 | Interactive | 40/day |
| 5 | Interactive | 50/day |
| 6 | Batch | 10/week |
| 7 | Interactive | 2/year |
| 8 | Batch | 1/month |

## 2.4 Access tables

In the total cost evaluation we assume that a write access costs like two read accesses.

**Access table for Operation 1**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | W |
| Phone | Entity | 1.5* | W |
| Num-Customer | Relationship | 1.5 | W |
| Loc-Customer | Relationship | 1 | W |

* We assume that a customer has normally between 1 and 2 phone numbers, thus 1.5 on average

Total: 5*20 write accesses = 200 accesses per day

**Access table for Operation 2a (RetainerHolder or RetainerAccessory)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| RetainerHolder / RetainerAccessory | Entity | 1 | W |
| ISA-H-P / ISA-A-P | Relationship | 1 | W |
| SnowRetainer | Entity | 50 | R |
| H-Compatible / A-Compatible | Relationship | 25 | W |

Total: 28*10 write accesses + 50*10 read accesses per month = 1.060 accesses per month

**Access table for Operation 2b (SnowRetainer)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| SnowRetainer | Entity | 1 | W |
| ISA-R-P | Relationship | 1 | W |
| GridRetainer / TubeRetainer | Entity | 1 | W |
| ISA-G-R / ISA-T-R | Relationship | 1 | W |
| RetainerHolder | Entity | 400 | R |
| RetainerAccessory | Entity | 50 | R |
| H-Compatible | Relationship | 200 | W |
| A-Compatible | Relationship | 25 | W |

Total: 230*10 write accesses + 450*10 read accesses = 9.100 accesses per month

**Access table for Operation 3**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 1 | W |
| Is-In-Prov | Relationship | 1 | W |

Total: 2*5 write accesses = 20 accesses per day

**Access table for Operation 4**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| BuildingSite | Entity | 1 | W |
| Loc-Roof | Relationship | 1 | W |
| SnowloadComputation | Entity | 1 | W |
| Comp-For-BS | Relationship | 1 | W |

Total: 4*40 write accesses = 320 accesses per day

**Access table for Operation 5**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowloadComputation | Entity | 1 | R |
| Comp-For-BS | Relationship | 1 | R |
| BuildingSite | Entity | 1 | R |
| RetainerHolder | Entity | 1 | R |
| GridRetainer / TubeRetainer | Entity | 1 | R |
| ISA-G-R / ISA-T-R | Relationship | 1 | R |
| SnowRetainer | Entity | 1 | R |
| RetainerAccessory | Entity | 1 | R |
| H-Compatible | Relationship | 1 | R |
| A-Compatible | Relationship | 1 | R |
| ISA-H-P | Relationship | 1 | R |
| ISA-R-P | Relationship | 1 | R |
| ISA-A-P | Relationship | 1 | R |
| SnowstopProduct | Entity | 3 | R |
| SupplyOffer | Entity | 1 | W |
| Offer-Prod | Relationship | 3* | W |
| For Customer | Relationship | 1 | W |

* On average three products offered

Total: 5*50 write accesses + 16*50 read accesses = 1.300 accesses per day

**Access table for Operation 6**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | R |
| For-Customer | Relation | 5 | R |
| SupplyOffer | Entity | 5* | R |

* We suppose on average 5 supply offers for customer

Total: 11*10 read accesses = 110 read accesses per week

**Access table for Operation 7**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 500 | R |
| SnowstopProduct | Entity | 500 | W |

Total: 500*2 read accesses + 500*2 write accesses = 3.000 accesses per year

**Access table for Operation 8**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 8000 | R |
| City | Entity | 50* | W |

* We assume only a small percentage of city have their zip code changed

Total: 8000*1 read accesses + 50*1 write accesses = 8.100 accesses per month

# 3 Direct translation

## 3.1 Relational schema

SnowstopProduct(Code,Name,Material,Price,Color*)
  generalization constraint: SnowstopProduct[Code] ⊆ SnowRetainer[Code] ∪ RetainerHolder[Code] ∪ RetainerAccessory[Code]
  constraint: Material is 'Zink Steel' or 'Stainless Steel' or 'Painted Steel' or 'Aluminium' or 'Copper'
  constraint: Color is NULL if and only if Material is not 'Painted Steeel'

SnowRetainer(Code,LinearResistance)
  foreign key: SnowRetainer[Code] ⊆ SnowstopProduct[Code]
  inclusion: SnowRetainer[Code] ⊆ H-Compatible[RetainerCode]
  inclusion: SnowRetainer[Code] ⊆ A-Compatible[RetainerCode]
  generalization constraint: SnowRetainer[Code] ∩ RetainerHolder[Code] ∩ RetainerAccessory[Code] = ∅
  generalization constraint: SnowRetainer[Code] ⊆ GridRetainer[Code] ∪ TubeRetainer[Code]

GridRetainer(Code,Height,Profile)
  foreign key: GridRetainer[Code] ⊆ SnowRetainer[Code]
  generalization constraint: GridRetainer[Code] ∩ TubeRetainer[Code] = ∅

TubeRetainer(Code,Diameter)
  foreign key: TubeRetainer[Code] ⊆ SnowRetainer[Code]
  generalization constraint: TubeRetainer[Code] ∩ GridRetainer[Code] = ∅

RetainerHolder(Code,Resistance,RoofType)
  foreign key: RetainerHolder[Code] ⊆ SnowstopProduct[Code]
  inclusion: RetainerHolder[Code] ⊆ H-Compatible[HolderCode]
  generalization constraint: RetainerHolder[Code] ∩ SnowRetainer[Code] ∩ RetainerAccessory[Code] = ∅
  constraint: Rooftype is 'Concrete Tile' or 'Ondulated Plate' or 'Trapezoidal Sheet' or 'Standing Seam Sheet' or 'Flat Tile'

H-Compatible(HolderCode,RetainerCode)
  foreign key: H-Compatible[HolderCode] ⊆ RetainerHolder[Code]
  foreign key: H-Compatible[RetainerCode] ⊆ SnowRetainer[Code]

RetainerAccessory(Code,Measure,Type)
  foreign key: RetainerAccessory[Code] ⊆ SnowstopProduct[Code]
  inclusion: RetainerAccessory[Code] ⊆ A-Compatible[AccessoryCode]
  generalization constraint: RetainerAccessory[Code] ∩ SnowRetainer[Code] ∩ RetainerHolder[Code] = ∅
  constraint: Type is 'Connection' or 'Ice Retainer'

A-Compatible(AccessoryCode,RetainerCode)
  foreign key: A-Compatible[AccessoryCode] ⊆ RetainerAccessory[Code]
  foreign key: A-Compatible[RetainerCode] ⊆ SnowRetainer[Code]

SupplyOffer(Code,Date,TotalPrice)
  foreign key: SupplyOffer[Code] ⊆ Has-Computation[OfferCode]
  inclusion: SupplyOffer[Code] ⊆ For-Customer[OfferCode]

For-Customer(OfferCode,CustomerCode)
  foreign key: For-Customer[OfferCode] ⊆ SupplyOffer[Code]
  foreign key: For-Customer[CustomerCode] ⊆ Customer[Code]

Offer-Prod(OfferCode,ProductCode,Quantity)
  foreign key: Offer-Prod[OfferCode] $\subseteq$ SupplyOffer[Code]
  foreign key: Offer-Prod[ProductCode] $\subseteq$ SnowstopProduct[Code]

Has-Computation(OfferCode,ComputationCode,TotalResistance,Rows,Distance)
  foreign key: Has-Computation[OfferCode] $\subseteq$ SupplyOffer[Code]
  foreign key: Has-Computation[ComputationCode] $\subseteq$ SnowloadComputation[Code]

Customer(Code,Name,Discount)
  foreign key: Customer[Code] $\subseteq$ Loc-Customer[CustomerCode]
  inclusion: Customer[Code] $\subseteq$ Num-Customer[CustomerCode]
  constraint: Discount $\geqslant 0$ and $\leqslant 30$

Phone(Number)
  foreign key: Phone[Number] $\subseteq$ Num-Customer[PhoneNumber]

Num-Customer(PhoneNumber,CustomerCode)
  foreign key: Num-Customer[PhoneNumber] $\subseteq$ Phone[Number]
  foreign key: Num-Customer[CustomerCode] $\subseteq$ Customer[Code]

Loc-Customer(CustomerCode,Zip,City)
  foreign key: Loc-Customer[CustomerCode] $\subseteq$ Customer[Code]
  foreign key: Loc-Customer[Zip,City] $\subseteq$ City[Zip,Name]

SnowloadComputation(Code,Date,GroundLoad,RoofLoad)
  foreign key: SnowloadComputation[Code] $\subseteq$ Comp-For-BS[ComputationCode]

Comp-For-BS(ComputationCode,RoofName,Zip,City)
  foreign key: Comp-For-BS[ComputationCode] $\subseteq$ SnowloadComputation[Code]
  foreign key: Comp-For-BS[RoofName,Zip,City] $\subseteq$ BuildingSite[Name,Zip,City]
  key: RoofName,Zip,City

BuildingSite(Name,Zip,City,Length,Width,Steepness,Covering)
  foreign key: BuildingSite[Zip,City] $\subseteq$ City[Zip,Name]
  foreign key: BuildingSite[Name,Zip,City] $\subseteq$ Comp-For-BS[RoofName,Zip,City]
  constraint: Covering is 'Concrete Tile' or 'Ondulated Plate' or 'Trapezoidal Sheet' or
    'Standing Seam Sheet' or 'Flat Tile'

City(Zip,Name,Altitude)
  foreign key: City[Zip,Name] $\subseteq$ Is-In-Prov[Zip,Name]

Province(Shorthand,Name,Zone,BaseLoad)
  constraint: Zone is 'I-A' or 'I-M' or 'II' or 'III
  constraint: BaseLoad $> 0$

Is-In-Prov(Zip,City,Province)
  foreign key: Is-In-Prov[Zip,City] $\subseteq$ City[Zip,Name]
  foreign key: Is-In-Prov[Province] $\subseteq$ Province[Shorthand]

| External integrity constraints in terms of the relational schema | |
|---|---|
| **1** | The TotalResistance attribute in the Has-Computation relationship must be higher than the RoofLoad attribute of its associated SnowloadComputation |
| **2a** | The Code of the RetainerHolder and the Code of the SnowRetainer associated to the same SupplyOffer through the Offer-Prod relationship must appear as an instance in the H-Compatible relationship |
| **2b** | The Code of the RetainerAccessory and the Code of the SnowRetainer associated to the same SupplyOffer through the Offer-Prod relationship must appear as an instance in the A-Compatible relationship |
| **3** | The RoofType attribute of a RetainerHolder associated to a SupplyOffer must have the same value as the Covering attribute of the BuildingSite associated to it following the relationships path by passing through SnowloadComputation entity. |
| **4** | The Code of the SupplyOffer must participate between 2 and 4 times to the Offer-Prod relationship. In addition, the same Code being present as an instance in the Offer-Prod relationship must be associated with exactly one RetainerHolder, one SnowRetainer and between zero and two different types of RetainerAccessory |
| **5** | The TotalPrice attribute of a SupplyOffer must be equal to the sum of the prices of the SnowstopProduct entities associated to it through the Offer-Prod relationship multiplied by their price |

## 3.2 Application load in terms of the relational schema

### 3.2.1 Table of volumes and operations

**Table of volumes after the direct translation**

| Concept | Construct | Volume |
|---|---|---|
| **SnowstopProduct** | Entity | 500 |
| **SnowRetainer** | Entity | 50 |
| **GridRetainer** | Entity | 25 |
| **TubeRetainer** | Entity | 25 |
| **RetainerHolder** | Entity | 400 |
| **RetainerAccessory** | Entity | 50 |
| **SupplyOffer** | Entity | 5000 |
| **Customer** | Entity | 2500 |
| **Phone** | Entity | 3750 |
| **SnowloadComputation** | Entity | 4000 |
| **BuildingSite** | Entity | 4000 |
| **City** | Entity | 8000 |
| **Province** | Entity | 100 |
| **H-Compatible** | Relationship | 10000* |
| **A-Compatible** | Relationship | 1250* |
| **Offer-Prod** | Relationship | 15000 |
| **For-Customer** | Relationship | 6000 |
| **Loc-Customer** | Relationship | 2500 |
| **Num-Customer** | Relationship | 3750 |
| **Has-Computation** | Relationship | 5000 |
| **Comp-For-BS** | Relationship | 4000 |
| **Loc-Roof** | Relationship | 4000 |
| **Is-In-Prov** | Relationship | 8000 |

* Each RetainerHolder is on average compatible with half of the SnowRetainer, same reasoning applies to the RetainerAccessory.

**Operations of interest:**

1. Insert a new customer.
2. Insert a new snowstop product, defining also the type and the compatibility.
3. Insert a new city.
4. Create a snowload computation on a given building site.
5. Create a supply offer.
6. List all the supply offers made for a given customer.
7. Update prices of snowstop products.
8. Update zip code and name of cities.

**Table of Operations**

| Operation | Type | Frequency |
|---|---|---|
| 1 | Interactive | 20/day |
| 2 | Interactive | 10/month |
| 3 | Interactive | 5/day |
| 4 | Interactive | 40/day |
| 5 | Interactive | 50/day |
| 6 | Batch | 10/week |
| 7 | Interactive | 2/year |
| 8 | Batch | 1/month |

### 3.2.2 Access tables

In the total cost evaluation we assume that a write access costs like two read accesses.

**Access table for Operation 1**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | W |
| Phone | Entity | 1.5* | W |
| Num-Customer | Relationship | 1.5 | W |
| Loc-Customer | Relationship | 1 | W |

* We assume that a customer has normally between 1 and 2 phone numbers, thus 1.5 on average

Total: 5*20 write accesses = 200 accesses per day

**Access table for Operation 2a (RetainerHolder or RetainerAccessory)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| RetainerHolder / RetainerAccessory | Entity | 1 | W |
| SnowRetainer | Entity | 50 | R |
| H-Compatible / A-Compatible | Relationship | 25 | W |

Total: 27*10 write accesses + 50*10 read accesses per month = 1.040 accesses per month

**Access table for Operation 2b (SnowRetainer)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| SnowRetainer | Entity | 1 | W |
| GridRetainer / TubeRetainer | Entity | 1 | W |
| RetainerHolder | Entity | 400 | R |
| RetainerAccessory | Entity | 50 | R |
| H-Compatible | Relationship | 200 | W |
| A-Compatible | Relationship | 25 | W |

Total: 228*10 write accesses + 450*10 read accesses = 9.060 accesses per month

**Access table for Operation 3**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 1 | W |
| Is-In-Prov | Relationship | 1 | W |

Total: 2*5 write accesses = 20 accesses per day

**Access table for Operation 4**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| BuildingSite | Entity | 1 | W |
| SnowloadComputation | Entity | 1 | W |
| Comp-For-BS | Relationship | 1 | W |

Total: 3*40 write accesses = 240 accesses per day

**Access table for Operation 5**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowloadComputation | Entity | 1 | R |
| Comp-For-BS | Relationship | 1 | R |
| BuildingSite | Entity | 1 | R |
| RetainerHolder | Entity | 1 | R |
| GridRetainer / TubeRetainer | Entity | 1 | R |
| SnowRetainer | Entity | 1 | R |
| RetainerAccessory | Entity | 1 | R |
| H-Compatible | Relationship | 1 | R |
| A-Compatible | Relationship | 1 | R |
| SnowstopProduct | Entity | 3 | R |
| SupplyOffer | Entity | 1 | W |
| Offer-Prod | Relationship | 3* | W |
| For Customer | Relationship | 1 | W |

* On average three products offered

Total: 5*50 write accesses + 12*50 read accesses = 1.100 accesses per day

**Access table for Operation 6**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | R |
| For-Customer | Relation | 5 | R |
| SupplyOffer | Entity | 5* | R |

* We suppose on average 5 supply offers for customer

Total: 11*10 read accesses = 110 read accesses per week

**Access table for Operation 7**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 500 | R |
| SnowstopProduct | Entity | 500 | W |

Total: 500*2 read accesses + 500*2 write accesses = 3.000 accesses per year

**Access table for Operation 8**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 8000 | R |
| City | Entity | 50* | W |

* We assume only a small percentage of city have their zip code changed

Total: 8000*1 read accesses + 50*1 write accesses = 8.100 accesses per month

# 4 Restructuring of the relational schema

## 4.1 Restructured relational schema

SnowstopProduct(Code,Name,Material,Color*,Price)
- generalization constraint: SnowstopProduct[Code] $\subseteq$ SnowRetainer[Code] $\cup$ RetainerHolder[Code] $\cup$ RetainerAccessory[Code]
- constraint: Material is 'Zink Steel' or 'Stainless Steel' or 'Painted Steel' or 'Aluminium' or 'Copper'
- constraint: Color is NULL if and only if Material is not 'Painted Steeel'

SnowRetainer(Code,LinearResistance,RetainerType,Measure,Profile*)
- foreign key: SnowRetainer[Code] $\subseteq$ SnowstopProduct[Code]
- generalization constraint: SnowRetainer[Code] $\cap$ RetainerHolder[Code] $\cap$ RetainerAccessory[Code] $= \varnothing$
- constraint: RetainerType is 'Grid' or 'Tube'
- constraint: Profile is NULL if RetainerType is 'Tube', otherwise Profile is not NULL

RetainerHolder(Code,Resistance,RoofType,RetainerType)
- foreign key: RetainerHolder[Code] $\subseteq$ SnowstopProduct[Code]
- generalization constraint: RetainerHolder[Code] $\cap$ SnowRetainer[Code] $\cap$ RetainerAccessory[Code] $= \varnothing$
- constraint: Rooftype is 'Concrete Tile' or 'Ondulated Plate' or 'Trapezoidal Sheet' or 'Standing Seam Sheet' or 'Flat Tile'
- constraint: RetainerType is 'Grid' or 'Tube'

RetainerAccessory(Code,Measure,Type,RetainerType)
- foreign key: RetainerAccessory[Code] $\subseteq$ SnowstopProduct[Code]
- generalization constraint: RetainerAccessory[Code] $\cap$ SnowRetainer[Code] $\cap$ RetainerHolder[Code] $= \varnothing$
- constraint: Type is 'Connection' or 'Ice Retainer'
- constraint: RetainerType is 'Grid' or 'Tube'

SupplyOffer(Code,ComputationCode,Date,TotalPrice,TotalResistance,Rows,Distance)
- foreign key: SupplyOffer[ComputationCode] $\subseteq$ SnowloadComputation[Code]
- inclusion: SupplyOffer[Code] $\subseteq$ For-Customer[OfferCode]

For-Customer(OfferCode,CustomerCode)
- foreign key: For-Customer[OfferCode] $\subseteq$ SupplyOffer[Code]
- foreign key: For-Customer[CustomerCode] $\subseteq$ Customer[Code]

Offer-Prod(OfferCode,ProductCode,Quantity)
- foreign key: Offer-Prod[OfferCode] $\subseteq$ SupplyOffer[Code]
- foreign key: Offer-Prod[ProductCode] $\subseteq$ SnowstopProduct[Code]

Customer(Code,Name,Zip,City,Discount)
- foreign key: Customer[Zip,City] $\subseteq$ City[Zip,Name]
- inclusion: Customer[Code] $\subseteq$ Phone[CustomerCode]
- constraint: Discount $\geqslant 0$ and $\leqslant 30$

Phone(Number,CustomerCode)
- foreign key: Phone[CustomerCode] $\subseteq$ Customer[Code]

SnowloadComputation(Code,Date,GroundLoad,RoofLoad)
- foreign key: SnowloadComputation[Code] $\subseteq$ BuildingSite[ComputationCode]

BuildingSite(Name,Zip,City,ComputationCode,Length,Width,Steepness,Covering)
  foreign key: BuildingSite[Zip,City] ⊆ City[Zip,Name]
  foreign key: BuildingSite[ComputationCode] ⊆ SnowloadComputation[Code]
  constraint: Covering is 'Concrete Tile' or 'Ondulated Plate' or 'Trapezoidal Sheet' or 'Standing Seam Sheet' or 'Flat Tile'
  key: ComputationCode

City(Zip,Name,Province,Altitude)
  foreign key: City[Province] ⊆ Province[Shorthand]

Province(Shorthand,Name,Zone,BaseLoad)
  constraint: Zone is 'I-A' or 'I-M' or 'II' or 'III
  constraint: BaseLoad > 0

| **External integrity constraints in terms of the restructured relational schema** | |
|---|---|
| **1** | The TotalResistance attribute of SupplyOffer must be higher than the RoofLoad attribute of its associated SnowloadComputation |
| **2a** | The RetainerType attribute of RetainerHolder and SnowRetainer associated to the same SupplyOffer through the Offer-Prod relationship must have the same value |
| **2b** | The RetainerType attribute of RetainerAccessory and SnowRetainer associated to the same SupplyOffer through the Offer-Prod relationship must have the same value |
| **3** | The RoofType attribute of RetainerHolder associated to a SupplyOffer must have the same value as the Covering attribute of the BuildingSite associated to it following the relationships path by passing through SnowloadComputation entity. |
| **4** | The Code of SupplyOffer must participate between 2 and 4 times to the Offer-Prod relationship. In addition, the same Code being present as an instance in the Offer-Prod relationship must be associated with exactly one RetainerHolder, one SnowRetainer and between zero and two different types of RetainerAccessory |
| **5** | The TotalPrice attribute of a SupplyOffer must be equal to the sum of the prices of the SnowstopProduct entities associated to it through the Offer-Prod relationship multiplied by their price |

## 4.2 Reasons for the restructuring steps

- By carefully analysing the domain of the database, it follows that a SnowRetainer can only be of two types, namely Grid or Tube, we can avoid to represent the GridRetainer and TubeRetainer relations and instead add an Attribute to SnowRetainer with the constraint of it being either 'Grid' or 'Tube'. Instead of having the attributes 'Height' and 'Diameter' (which we have on GridRetainer and TubeRetainer) we can create a new attribute to represent them both and call it 'Measure', in addition we can also add the attribute 'Profile' (previously on GridRetainer) and make it optional, by also adding a constraint on it being null only if the attribute 'Type' has the value 'Tube'. In this way we avoid one write in operation 2b.
- By extending the same concept explained in the last point also to RetainerHolder and Retainer-Accessory we can add the same 'Type' attribute to these two relations. In this way we can avoid creating the relationships H-Compatible and A-Compatible by modifying the external constraint nr 4 on the SupplyOffer to check that the Type attributes correspond. As a result, operation 2a will need respectively 25 and 225 less writes and in addition 50 and 450 less reads. Operation 5 will need 2 less reads and in addition the satisfaction of the external constraint on SupplyOffer will become easier to manage.
- We merge Customer with Loc-Customer relationship, by adding Zip and City attributes to the relation Customer. This avoids one write in operation 1.
- We merge Phone with Num-Customer relationship, by adding CustomerCode attribute to the relation Phone. Similarly as above, this decreases the number of writes needed for operation 1 by an average of 1.5.
- We merge City with Is-In-Prov relationship, by adding new attribute Province to the relation City, in this way reduce the numbers of joins needed to obtain the climatic zone of a city from 3 to 2.
- We merge SupplyOffer with Has-Computation relationship, by adding all the attributes of the relationship to the relation SupplyOffer. In this way we avoid creating an additional relationship when creating a SupplyOffer and we need to perform one less join operation when accessing a Snowload-Computation that is associated to a SupplyOffer.
- We merge BuildingSite with Comp-For-BS relationship, by adding the Code of SnowloadComputation relation to the attributes of BuildingSite relation. Since the two relations are usually accessed together we can merge them in order to have one less write on operation 4.

## 4.3 Application load after the restructuring

### 4.3.1 Restructured table of volumes and operations

**Table of volumes after the restructuring of the relational schema**

| Concept | Construct | Volume |
|---|---|---|
| SnowstopProduct | Entity | 500 |
| SnowRetainer | Entity | 50 |
| RetainerHolder | Entity | 400 |
| RetainerAccessory | Entity | 50 |
| SupplyOffer | Entity | 5000 |
| Customer | Entity | 2500 |
| Phone | Entity | 3750 |
| SnowloadComputation | Entity | 4000 |
| BuildingSite | Entity | 4000 |
| City | Entity | 8000 |
| Province | Entity | 100 |
| Offer-Prod | Relationship | 15000 |
| For-Customer | Relationship | 6000 |

**Operations of interest:**

1. Insert a new customer.
2. Insert a new snowstop product, defining also the type and the compatibility.
3. Insert a new city.
4. Create a snowload computation on a given building site.
5. Create a supply offer.
6. List all the supply offers made for a given customer.
7. Update prices of snowstop products.
8. Update zip code and name of cities.

**Table of Operations**

| Operation | Type | Frequency |
|---|---|---|
| 1 | Interactive | 20/day |
| 2 | Interactive | 10/month |
| 3 | Interactive | 5/day |
| 4 | Interactive | 40/day |
| 5 | Interactive | 50/day |
| 6 | Batch | 10/week |
| 7 | Interactive | 2/year |
| 8 | Batch | 1/month |

### 4.3.2 Restructured access tables

In the total cost evaluation we assume that a write access costs like two read accesses.

**Access table for Operation 1**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | W |
| Phone | Entity | 1.5* | W |

* We assume that a customer has normally between 1 and 2 phone numbers, thus 1.5 on average

Total: 2.5*20 write accesses = 100 accesses per day

**Access table for Operation 2a (RetainerHolder or RetainerAccessory)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| RetainerHolder / RetainerAccessory | Entity | 1 | W |

Total: 2*10 write accesses = 40 accesses per month

**Access table for Operation 2b (SnowRetainer)**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 1 | W |
| SnowRetainer | Entity | 1 | W |

Total: 2*10 write accesses = 40 accesses per month

**Access table for Operation 3**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 1 | W |

Total: 1*5 write accesses = 10 accesses per day

**Access table for Operation 4**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| BuildingSite | Entity | 1 | W |
| SnowloadComputation | Entity | 1 | W |

Total: 2*40 write accesses = 160 accesses per day

**Access table for Operation 5**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowloadComputation | Entity | 1 | R |
| BuildingSite | Entity | 1 | R |
| RetainerHolder | Entity | 1 | R |
| SnowRetainer | Entity | 1 | R |
| RetainerAccessory | Entity | 1 | R |
| SnowstopProduct | Entity | 3 | R |
| SupplyOffer | Entity | 1 | W |
| Offer-Prod | Relationship | 3* | W |
| For Customer | Relationship | 1 | W |

* On average three products offered

Total: 5*50 write accesses + 8*50 read accesses = 1000 accesses per day

**Access table for Operation 6**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| Customer | Entity | 1 | R |
| For-Customer | Relation | 5 | R |
| SupplyOffer | Entity | 5* | R |

* We suppose on average 5 supply offers for customer

Total: 11*10 read accesses = 110 read accesses per week

**Access table for Operation 7**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| SnowstopProduct | Entity | 500 | R |
| SnowstopProduct | Entity | 500 | W |

Total: 500*2 read accesses + 500*2 write accesses = 3.000 accesses per year

**Access table for Operation 8**

| Concept | Construct | Accesses | Type |
|---|---|---|---|
| City | Entity | 8000 | R |
| City | Entity | 50* | W |

* We assume only a small percentage of city have their zip code changed

Total: 8000*1 read accesses + 50*1 write accesses = 8.100 accesses per month

# 5 Conclusion

## 5.1 Final remarks

By starting with the specifications of the domain and the idea for the project, we applied the four main phases of the design of a database. The restructured relational schema will be translated to an SQL specification and will be used by a Java Application that will interact with it. The SQL and Java files will be sent along with this document.

The database definition that has been described in the previous pages has been developed in response to a real need of a company environment based in Bolzano which operates in the retail field of snow retainment systems.

The aim of this project is to develop a database that could enable the automatization of the companies processes involved in the production of supply offers and the calculation of snow retaining systems. The "so called" Snowstop database will enable employees of the company to easily perform calculations on the snow load of a particular building site and propose to customers a supply offer that will provide a tailored snow retaining system based on such calculations. The altitude and the dimensions of the roof for which a snow load calculation is perfomed will affect the actual amount of snow that will deposit on it and the type and resistance of the retaining system needed.