

# **A Database Design for Storage and Retrieval of Snow Profiles on a PC.**

Peter Weir. P. Geo.  
Research Officer  
Snow Avalanche Programs,  
British Columbia Ministry of Transportation and Highways  
940 Blanshard St., Victoria, BC, Canada V8W 3E6  
Tel. (604) 387-7517, Fax (604) 356-8143  
E-mail: PWeir@Galaxy.gov.bc.ca

## **Overview**

An application for plotting snow profiles has been designed for the Windows Personal Computer operating environment. A Graphical User Interface allows the IASH / ICSI symbols to be presented in pick lists with accompanying textual descriptions. A mouse or similar pointing device is used to make selections wherever possible.

Provided that density measurements are made, the application automatically computes the equivalent water content of each layer and of the entire snowpack. A weighted average snowpack density is also computed. The application uses the graphical area on the left of the profile, traditionally reserved for plotting Ramsonde hardness, to display a plot of the results of a hand hardness analysis and / or a cumulative plot of slope parallel shear stress (a function of slope angle, layer thickness and density).

The application supports a display-only mode, a fixed corporate format and a mode that supports Dynamic Data Exchange. In its simplest form the application stores data from a single snow profile in a comma delimited ASCII file. This approach suits small sites and facilitates the transfer and digital exchange of snow profile information.

A database has been designed to accompany the application so that a variety of other PC based tools, including a Rule Based Expert System used for snow profile interpretation, can share its data. A logical data model is presented for use in the design of a relational database. Entity relationship diagrams are explained.

In a larger corporate application, as exists in the BC Ministry of Transportation and Highways, a database approach is warranted to facilitate the storage and retrieval of the large number of snow profiles collected each season. Queries can be run to identify and display snow profiles from a given date range, area or region, and / or profiles which satisfy any given set of snowpack conditions.

## **Introduction**

Many snow safety operations now use personal computers to store weather, snowpack and avalanche occurrence data. As seasons pass, managing this information becomes a challenge or a headache (depending on your perspective). In the worst case the computer becomes a sophisticated tool for losing data. The potential cost savings that may have been used to convince management to invest in Personal Computer (PC) technology can become large costs which, if left unchecked, will draw more and more financial and staff resources. Data consistency and quality control issues compound into major problems, especially as staff change or as new technologies are adopted.

Hourly weather observations from one or more electronic dataloggers can generate more data in a single season than may accumulate over a decade when only manually derived data are collected. Data are often stored as ASCII files which is then imported into a spreadsheet or graphics display package. Other data may be entered directly into a spreadsheet. Information stored this way may be loosely termed a database, most spreadsheets purport to have some database functions, but a file based approach fails to qualify as a database management system.

This paper will endeavour to establish a working vocabulary and overview of the methodology required to set up a relational database in a PC environment. A strict definition of a database will be established using snow profile data as an example of information to be encoded. The experience of Snow Avalanche Programs (SAP) of the British Columbia Ministry of Transportation and Highways (MoTH) in creating a database application is outlined.

## **Computing in the Ministry of Transportation and Highways Environment**

Snow Avalanche Programs of MoTH operate in 9 centres across the Province of British Columbia, Canada where some 1200 avalanche paths are monitored in 69 different avalanche areas. Each avalanche technician has at least one 486 class workstation and a 286 PC dedicated to polling a cluster of remote automatic weather stations. Each computer is connected to a file server to form a local area network (LAN). Servers are connected with routers to form a Province wide WAN (Wide Area Network).

Snow Avalanche Programs' headquarters in Victoria on Vancouver Island is co-located with the Ministry's Information System Branch which employs over 150 computer specialists, programmers, analysts, systems, network, PC support and help-desk staff, data and database administrators and project managers. Outside consulting companies are contracted for special projects. Thus we are perhaps in a much more fortunate position than many other safety organisations by having a richly skilled group of experts available to develop computer applications.

Snow profiles entered at any one of the field operations are saved in a database on the local LAN and a copy appended into the SAP headquarters database. This information immediately becomes available to SAP's headquarters nine staff who fill advisory and support roles to the 35 operational field staff. Over 400 profiles per year are added to the database.

SAP was established in the mid 1970s and data collected were kept on a mainframe computer operated by a government systems provider. Four years ago the data management challenges facing SAP were quite different to those typically encountered by smaller organisations. A fundamental problem with the old system was that there was no way to relate weather, snow or avalanche events together. Furthermore we lacked interactive tools to browse or make ad hoc queries on the data.

In 1991 SAP made a decision to abandon its mainframe database (typically referred to as a "legacy" system) in order to build local databases connected by a WAN. The process of creating a new database in a PC environment was very similar to that faced by a small snow safety operation.

### **A Working Vocabulary**

Computing is dominated by jargon and acronyms but it becomes necessary to be literate in these areas in order to communicate with systems staff. A variety of database structure exist and an even wider variety of products purport to be Database Management Systems (DBMS). Many of the "impostors" store records in "flat" files rather than in indexed databases. The worst case is when data are stored in an application specific format that cannot be read by any other program. There is no universally accepted definition of a database but the following characteristics should exist:

- There is only one copy of each item of data.  
Data redundancy is minimised.
- The data can be used by a number of different applications.  
Data Flexibility is maximised.
- The Data are indexed and cross referenced in order to define all required relationships amongst the different data items.

Relational databases are based on a logical data model. They are easy to comprehend but an understanding of the underlying data model is essential for them to be employed effectively. They are easy to implement although a poor implementation is easier to produce than a good one! Data access is independent of the physical storage mechanism. Relational databases are adaptable. An extra table can be added if a new entity is introduced without impacting the existing tables; in the snow profile database it was no problem to add a table to store the results of shovel shear test and rutschblock test after we had produced the initial data model.

## Entities

In data base modeling an entity is used to describe items about which we collect data. It can be considered "A person, place, thing, event or concept" (Table 1). Entities are generally referred to in the singular. They are identified by phrases like noun verb noun i.e. entity relationship entity.

Example        A snow profile has one or more layers.

<b>ENTITY NAME</b>	RAM HARDNESS
<b>ENTITY IDENTIFIER</b>	F4.10
<b>DEFINITION</b>	A measure of the resistance to penetration offered by the snowpack to a 40 mm wide cone with a 60 degree pitch. The cone is driven vertically down by a blow from a hammer typically of kg mass which is dropped from heights of 5 to 50 cm. The principal action is one of disaggregation involving both shear and compression (ref. Keeler 1967). Ram Hardness is expressed as a force (in Newtons). This is the product of its mass (kg) multiplied by the acceleration due to gravity (approx. 10 m/sec). Ram hardness gives an index of snow strength.
<b>UNIQUE IDENTIFIER</b>	Ram Hardness Id = Snow Profile Observation Id + Ram depth
<b>VOLUME</b>	
<b>TRUSTEE</b>	Avalanche Technician
<b>ALIASES</b>	
<b>ATTRIBUTE LIST</b>	Ram Hardness Height Vertical Ram Hardness
<b>RELATIONSHIP LIST</b>	

Table 1.        Entity definition from data dictionary.

## Attributes

An attribute is considered to be any characteristic, property or description of an entity; it is data that concerns the entity. The term field, data field or data item are equivalent to an attribute. An attribute is referred to by a single name but it may have multiple aliases, which must be noted in the data dictionary (Table 2).

**Example**      A layer from a snow profile is characterised by its liquid water content, grain form, grain size, hardness and density.

<b>ENTITY NAME</b>	LAYER
<b>ATTRIBUTE NAME</b>	HAND HARDNESS
<b>DEFINITION</b>	Subjective test that involves gently pushing various sized objects horizontally into the snow layer by hand (with a force of about 15N). Produces an index hardness value.
<b>VALUE RANGE</b>	1. Fist = Very Low 2. 4 fingers = Low 3. 1 finger = Moderate 4. Pencil = High 5. Knife Blade = Very High 6. Ice
<b>FORMAT</b>	Integer
<b>ALIASES</b>	Hardness, Resistance
<b>RETENTION PERIOD</b>	Indefinite

Table 2.      Attribute definition from data dictionary.

## Data Dictionary

A data dictionary is a vital part of any DBMS. It establishes accepted definitions for every entity in the database. To create a consistent database we must all agree that density is measured in units of Kg/m<sup>3</sup>, not percent water, or grams per cubic centimetre, or any other unit. The old term "specific gravity" will not appear in our data dictionary. Similarly we agree that grain forms will be classified according to the 1990 ICSI system and not its predecessor or some derivative thereof.

Most DBMS have "passive" data dictionaries. They must be manually updated to remain synchronised with the table structures which may be changed by programming staff as new entities are added. Keeping the data dictionary current becomes a vital task for the person charged with database administration. An outdated document rapidly changes status from being an asset to a liability. The best DBMS have an "active" data dictionary whereby any changes made to the table structure or attributes of those tables automatically updates the data dictionary. Oracle is a DBMS with an active dictionary.

## **Data Modeling**

Data must be viewed as an enterprise resource, a snow profile does not belong to the observer but to the snow safety organisation for whom s/he works. Its data are only of value when it is shared with others as information related to snowpack structure. Data should be considered stable through time whereas processes (i.e. PC applications) almost certainly change. For example, snow profile data collected 10 years ago remains valid but computer based tools that work with this data have and will probably continue to change. A data model is a tool to communicate the meaning of business data in business terms to computer specialists.

Three levels of data models are recognized:

A **Conceptual** model is a high level view of the major entities and their inter-relationships. Examples in our case are avalanche areas, avalanche paths, roads, weather stations, snow profiles and staff. This can be termed a corporate data model.

A **Logical** model shows the entities at a lower more detailed level. It contains the industry specific data and shows its inter-relationships. In the case of snow profiles we recognize five distinct entities:

- Snow profile location information
- Snow layer information
- Snow temperature profile
- Shear test results
- Ramsonde hardness profile

A **Physical** model contains the database design and is the most detailed. This model is dependent on the DBMS employed. It may contain changes made to the logical model to improve performance. Thus, it is the "as built" model. In the FoxPro DBMS each entity will be represented as a separate data table encoded as a distinct DOS file but in Access all tables are stored in a single DOS file.

In this discussion we focus on the logical data model.

Logical Term	Physical Term	Example
Entity	Table	Snw_layr.DBF
Attribute	Column	Hardness, Grain Form, Grain Size, Liquid water, Density
Entity Occurrence	Row	4 finger, 1b, 1 mm, Dry, 120 Kg/m <sup>3</sup>

Table 3. Nomenclature in Logical and Physical data models.

### Snow Profile Data Model

A normalised data model is proposed for efficient storage of snow profile information. A snow profile can be represented as five separate entities, each of which has a variety of attributes (figure 1).

An entity relationship diagram is used to describe the relationships between the five entities. The "crows foot" notation linking the entities indicates one-to-many relationships. The text surrounding the notation is read clockwise. Similar notations are available for indicating one-to-one or many-to-many relationships. The essential skill in data modeling is to eliminate these latter types of relationships in a process known as resolving the relationship to "third normal form" (3NF).

In our model, entities are linked by a system generated key field labeled Profile\_ID (underlined in figure 1). In our implementation the Profile\_ID is a system generated 8 character field that is repeated in all tables. In the table named Snow Profile, which contains the geographic and date information, Profile\_ID is the key field; it is unique and once assigned cannot be changed. The key includes a 2 character identifier which is assigned to each avalanche technician location. Thus, when the records are merged they retain their uniqueness. In the four other tables Profile\_ID is a foreign key which is used to associate the tables allowing the relational database to function.

In an earlier implementation (described in the abstracts of ISSW 94) the entities were linked by a compound key composed of the profile location, date and time fields. However this combination failed to provide a unique label as a pair of observers sometimes undertook investigations at the same site at the same time. This revelation required a substantial update of our database structure to eliminate the possibility of storing profiles with the duplicate key fields.

True relational DBMS have "optimisers" that automatically determine the best way to join tables. This feature is rare in PC based DBMS at this time although Microsoft Access has a feature that approximates this process.

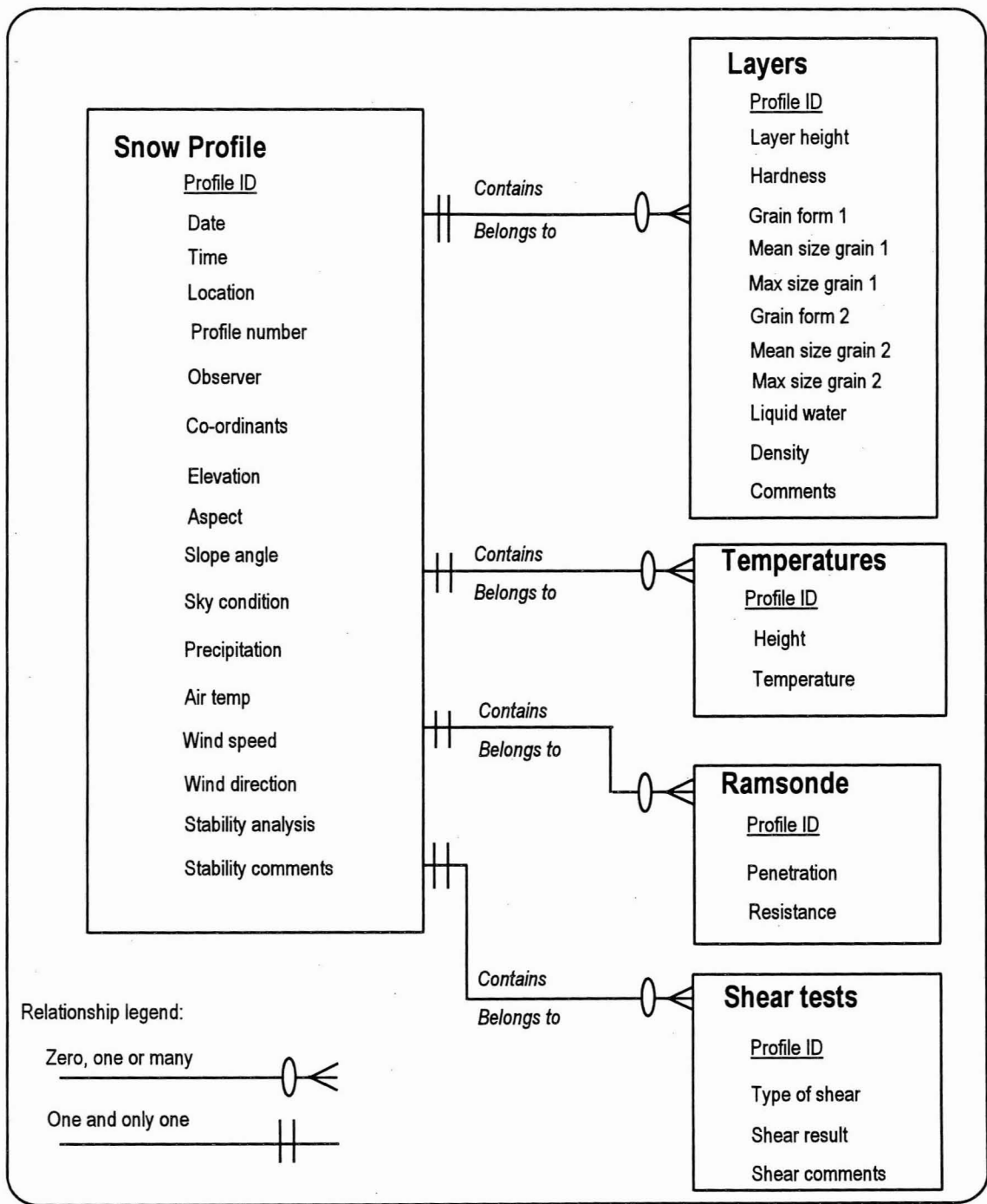


Fig 1. Logical data model for Snow Profile information (Entity Relationship Diagram).



Running a query that involve two or more entities is a true test of the data model. The query in figure 2 "Show all profiles where an easy or very easy shovel shear test occurred" requires the profile type entity to be related to the shear test entity.

### **Creating a Database Application**

Many months of effort went into defining our users' requirements, creating a data dictionary and designing the database using data modeling techniques before a single line of computer code was written. It is common to spend about 40 % of the total effort in the design phase in order to give programmers the clearest possible specifications. This ratio should be observed even in a small operation as the greatest paybacks follow effort expended at the planning stage.

About 40 % of the effort went into writing the code to create our database application. Most modern database languages (FoxPro for Windows in our case) are forth generation languages (4GLs). I.E. They are high level programming environments that contain a database engine and can be used to compile a run time executable program.

About 20 % of the effort went into commissioning, installing training users and debugging the code. Regrettably a further 30% of the projected effort went into "tuning and enhancing" the new database program after its first season in production. Alarming as it seems, this appears to be a common feature of many computer applications. A contingency fund should always be held over for this all important final phase. In our experience user satisfaction increases significantly following this final re-tuning phase. Our final product was a database application that we named SAWS, an acronym for Snow, Avalanche and Weather System.

Considerable effort was required to convert existing data into the new format. For example SAP had archived over 500 snow profiles on its mainframe but the text oriented computing technology of the 1980s had not provided programmers the tools to produce graphical output. Archiving snow profile information had ceased in 1985 as there appeared little hope of producing graphical output. The emergence of windows based graphical computing in the early 1990s enabled us to build the SnowPro application, something that would have been almost impossible 5 years earlier in a PC environment (figure 3). In 1990 the International Commission on Snow and Ice revised its snow classification scheme. Rules had to be established to translate grain forms classified according to the UNESCO/IASH/WMO 1970 scheme to those of ICSI 1990. The effort required to translate existing data are often underestimated when re-designing a system.

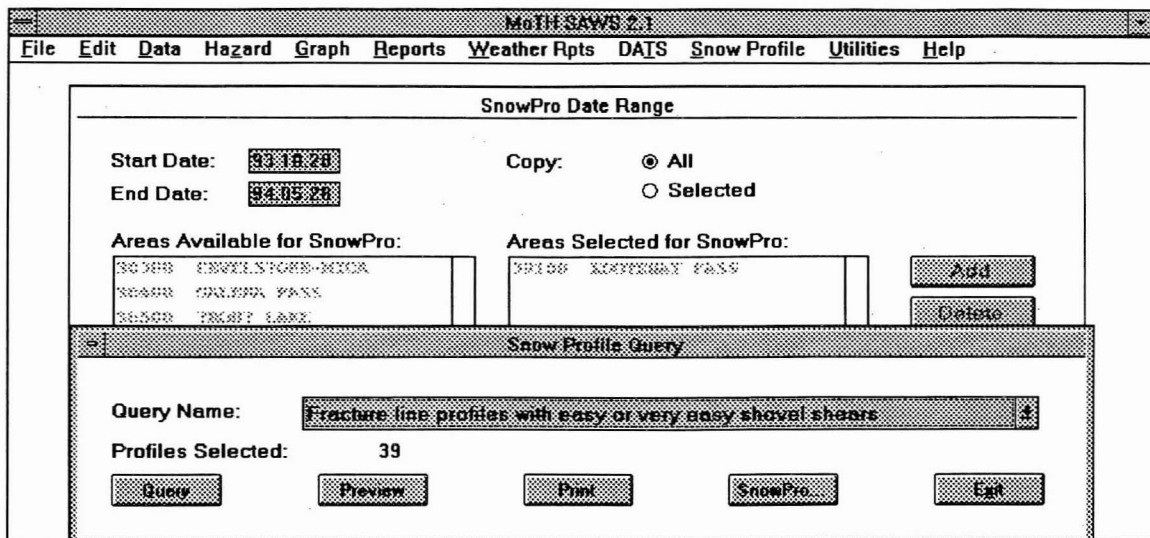


Fig 2. Query on snow profile database run from MoTH database application "SAWS".

Location		Campion low precip site				Area		Red Pass					
H.A.S.L.	1950	m	Co-ords			Sky Condition		⊗ Obscured					
Aspect	S	Slope		18		Precipitation		Snow -					
HS	171	HSW	549	<i>p</i>	325	R	-99	N	Air Temp	-3	Wind	Light - N	
R	1000	800	600	400	200	N							
T	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2	0		
	H	θ	F	E	R	HW	<i>p</i>	Comments					
	200												
	190												
	180						10						
	170						1		140		0.35		
	160						0.5				.05	← SM Just below ice crust	
	150						0.5		36 200		0.74		
	140						0.8		18 230				
	130						0.3		60 300				
	120						2				0.75	Surface hoar is rounded ← SH old surface hoar	
	110						0.5		41 270		.05		

Fig 3. Graphical output from SnowPro for Windows.  
(A Rule Based Expert System has highlighted layers most likely to fail).

Snow Avalanche Programs followed the computer industry's standard seven step systems development life cycle when designing its new DBMS. These steps are as follows:

1. Project Initiation
  - establish scope, feasibility, project plan. Management and user buy-in.
2. General Requirements
  - consider alternatives, recommend solutions
3. Detailed Requirements
  - functional requirements, implementation plan. Build conceptual data model.
4. Design
  - data dictionary, build logical data models, screens, report format, programs
5. Development
  - build tables, write code, testing
6. Implementation
  - user acceptance testing, training, data conversions
7. Post Implementation Review
  - were objectives met? List required enhancements

Snow Avalanche Programs are now entering a low input maintenance phase and plan to release minor enhancements at the start of each winter. We hope to get seven years production from our current application before contemplating restarting the development life cycle.

The standard system design process is expensive and time consuming. A Rapid System Design (RSD) approach may suit small PC applications for smaller operations. The first four steps are rolled into an intense two week period that typically commences with a two day JAD (Joint Application Design) session where a quick data model is built with the aid of CASE (Computer Aided Software Design) tools, a process that may take a further three days. In the second week users reconvene to examine the Entity Relationship diagrams. Coding then commences. There are much greater risks associated with RSD, especially for large applications, typically the data model may not support a query that may be required in the future. However, initial results are obtained much more quickly than when the standard approach is employed.

## Summary

This paper is intended to introduce the concept of logical data modeling which is the foundation on which a relational database is built. It mentions many of the key issues that must be considered when building a database or selecting a DBMS. Professional advice should be sought before any code is written. The payback for sound initial planning will come in savings in programming, in maintenance and in performance.

## References

- ICSI, 1990. *The International Classification for Seasonal Snow on the Ground*. International Commission on Snow and Ice of the International Association of Scientific Hydrology, 23p.
- McClung, D.M., Lakeman, G., Joseph, S. and Weir P.L. 1994. *An Expert System to Analyze Snow Profiles*. in preparation for Journal of Glaciology.
- UNESCO/IASH/WMO; 1970. *Seasonal Snow Cover*. United Nations Educational Scientific and Cultural Organisation, Paris; 38 p.