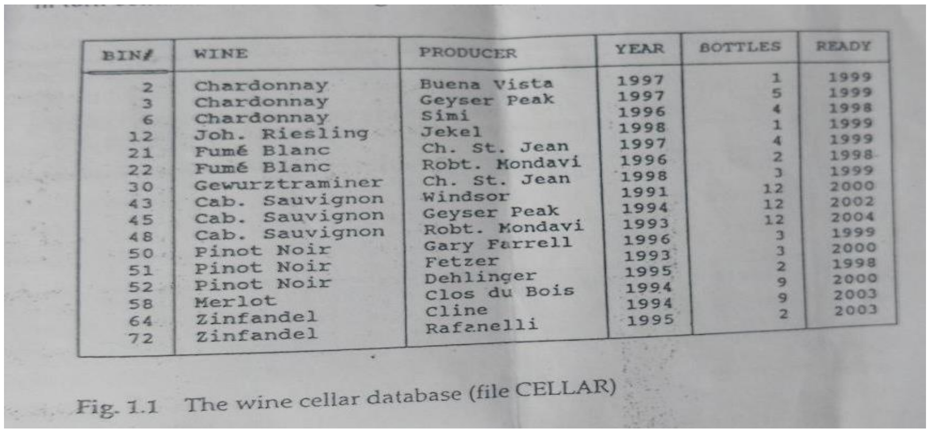
Overview of Database Management

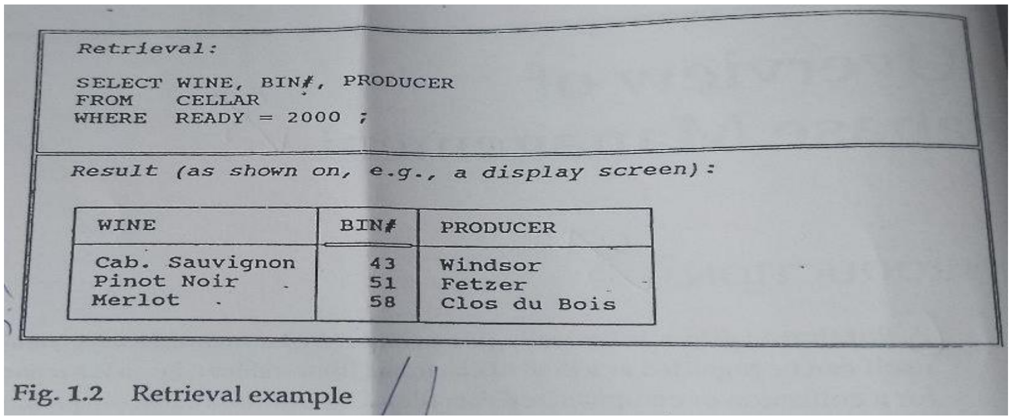
A database system is a computerized record-keeping system. The database can be regarded as a kind of electronic filing cabinet; i.e.. it is a repository or container for a collection of computerized data files. Users of the system can perform a variety of operations on such files-for example:

- Adding new, empty files to the database;
- Inserting data into existing files;
- Retrieving data from existing files;
- Changing data in existing files;
- Deleting data from existing files;
- Removing existing files from the database.



BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	1997	1	1999
3	Chardonnay	Geyser Peak	1997	5	1999
6	Chardonnay	Simi	1996	4	1998
12	Joh. Riesling	Jekel	1998	1	1999
21	Fumé Blanc	Ch. St. Jean	1997	4	1999
22	Fumé Blanc	Robt. Mondavi	1996	2	1998
30	Gewurztraminer	Ch. St. Jean	1998	3	1999
43	Cab. Sauvignon	Windsor	1991	12	2000
45	Cab. Sauvignon	Geyser Peak	1994	12	2002
48	Cab. Sauvignon	Robt. Mondavi	1993	12	2004
50	Pinot Noir	Gary Farrell	1996	3	1999
51	Pinot Noir	Fetzer	1993	3	2000
52	Pinot Noir	Dehlinger	1995	2	1998
58	Merlot	Clos du Bois	1994	9	2000
64	Zinfandel	Cline	1994	9	2003
72	Zinfandel	Rafanelli	1995	2	2003

Fig. 1.1 The wine cellar database (file CELLAR)



Retrieval:

```
SELECT WINE, BIN#, PRODUCER
FROM   CELLAR
WHERE  READY = 2000 ;
```

Result (as shown on, e.g., a display screen):

WINE	BIN#	PRODUCER
Cab. Sauvignon	43	Windsor
Pinot Noir	51	Fetzer
Merlot	58	Clos du Bois

Fig. 1.2 Retrieval example

What is Database?

Persistent data

In persistent, database data differs in kind from other more ephemeral data, such as input data, output data, control statements, work queues, software control blocks, intermediate results, and more generally any data that is transient in nature. Data in the database "persists" because once it has been accepted by the DBMS for entry in the database in the first place, it can subsequently be removed from the database only by some explicit request to the DBMS not as a mere side effect of some program completing execution.

A database is a collection of persistent data that is used by the application systems of some given enterprise.

The term "enterprise" here is simply a convenient generic term for any reasonably self-contained commercial, scientific, technical, or other organization. An enterprise might be a single individual or a complete corporation or similar large body or anything in between. Here are some examples:

1. A manufacturing company
2. A bank
3. A hospital
4. A university
5. A government department

Any enterprise must necessarily maintain a lot of data about its operation. Such data is the "persistent data" referred to above. The enterprises just mentioned would typically include the following among their persistent data

1. Product data
2. Account data
3. Patient data
4. Student data
5. Planning data

Data and Data Model

Relational database systems have become so dominant. They support the foregoing interpretation of data and databases very directly. Relational systems are based on a formal theory called the relational model of data, according to which

- **Data** is represented by means of rows in tables, and such rows can be directly interpreted as true propositions.
- **Operators** are provided for operating on rows in tables, and those operators directly sup-port the process of inferring additional true propositions from the given ones.

A data model is an abstract, self-contained, logical definition of the objects, operators, and so forth, that together constitute the abstract machine with which users interact. The objects allow us to model the structure of data. The operators allow us to model its behavior.

An implementation of a given data model is a physical realization on a real machine of the components of the abstract machine that together constitute that model.

The term data mode is used in the literature with two quite different meanings. The first is as described above. The second is as a model of the persistent data of some particular enterprise. The difference between the two meanings can be characterized thus:

- A data model in the **first sense** is like a programming language-albeit one that is somewhat abstract-whose constructs can be used to solve a wide variety of specific problems, but in and of themselves have no direct connection with any such specific problem.
- A data model in the **second sense** is like a specific program written in that language. In other words, a data model in the second sense takes the facilities provided by some model in the first sense and applies them to some specific problem. It can be regarded as a specific application of some model in the first sense.

The Conceptual Level

The conceptual view is a representation of the entire information content of the database again in a form that is somewhat abstract in comparison with the way in which the data is physically stored. It will also be quite different from the way in which the data is viewed by any particular user.

The conceptual view consist of many occurrences of each of many types of conceptual record. A conceptual record is not necessarily the same as either an external record on the one hand, or a stored record on the other,

The conceptual view is defined by means of the conceptual schema, which includes definitions of each of the various conceptual record types. The conceptual schema is written using another data definition language, the conceptual DDL. There must be no reference in the conceptual schema to stored field representation, stored record sequence, indexes, hashing schemes, pointers, or any other storage and access details.

The conceptual view, then, is a view of the total database content, and the conceptual schema is a definition of that view. The definitions in the conceptual schema are intended to include a great many additional features, such as the security and integrity constraints.

Mappings

The three levels per SE, the architecture involves certain mappings---one conceptual/internal mapping and several external/conceptual mappings, in general:

- **The conceptual/internal mapping** defines the correspondence between the conceptual view and the stored database; it specifies how conceptual records and fields are represented at the internal level. If the structure of the stored database is changed-i.e.. if a change is made to the storage structure definition-then the conceptual/internal map-ping must be changed accordingly, so that the conceptual schema can remain invariant.

- **An external/conceptual mapping** defines the correspondence between a particular ex-ternal view and the conceptual view. In general, the differences that can exist between these two levels are analogous to those that can exist between the conceptual view and the stored database. For example, fields can have different data types; field and record names can be changed; several conceptual fields can be combined into a single (virtual) external field; and so on. Any number of external views can exist at the same time; any number of users can share a given external view; different external views can overlap.

Utilities

Utilities are programs designed to help the DBA with various administration tasks. Some utility programs operate at the external level of the system, and thus are effectively nothing more than special-purpose applications. Other utilities, operate directly at the internal level and must be provided by the DBMS vendor.

Here are some examples of the kind of utilities

- **Load routines**, to create the initial version of the database from one or more operating system files;
- **Unload/reload routines,** to unload the database, or portions thereof, to backup storage and to reload data from such backup copies
- **Reorganization routines**, to rearrange the data in the stored database for various rea-sons e.g., to cluster data together in some particular way on the disk, or to reclaim space occupied by data that has become obsolete;
- **Statistical routines**, to compute various performance statistics such as file sizes or value distributions or I/O counts, etc.;
- **Analysis routines,** to analyze the statistics just mentioned;

5.(a) 255.255.174.19 (prefix notation)

The first two octets have a value of 255 (represent 16 1s) and the third octet has a value of 174 (represents 5 1s) and the fourth octet has a value of 19 (represents 3 1s) and 2

16 bits + 5 bits + 3 bits = 24 bits

Thus, a dotted-decimal notation of 255.255.174.19 has an equivalent prefix notation of /24. 3

(b) /28 (dotted-decimal notation)

Given a subnet mask of /28, the first, second and third octets have 8 1s and fourth octet has 4 1s. All 1s has a decimal value of 255. So, the first, second and third octets have 255 and fourth octet has 240.

Thus, a subnet mask with a prefix notation of /28 has an equivalent dotted-decimal notation of 255.255.255.240.

(c) 192.168.64.30/29 subnet (number of created subnets) 3

Number of created subnets = 2^s, where s is the number of borrowed bits

Number of borrowed bits = Bits in custom subnet – Bits in classful subnet mask

$$= 29 - 24 = 5$$

Number of created subnets = 2^5 = 32 subnets

(d) 172.25.50.120 /19 (40 subnets) 3

To support 40 subnets, use 6 borrowed bits.

The first octet has a value of 172. It is a class B address and has 16 bits in its classful mask.

So, add 6 borrowed bits to 16 bits. The result is 16 + 6 = 22 bits subnet mask.

Thus, 255.255.252.0 should be used for the subnet mask of /22.

(e) 172.20.64.10 /16 (110 hosts) 3

To support 110 hosts, the number of hosts supported by a specific number of host bits is 7.

An IPv4 address has 32 bits.

Number of subnet bits – Number of host bits = 32 – 7 = 25 subnet mask

Thus, a subnet mask of /25 should be used and can be written as 255.255.255.128 subnet.

(f) 118 2

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	0

Answer = 1110110

192.168.87.0 /24 (27-bit subnet mask) 17

The steps for calculating created subnet masks are as follow:

(1) Determine the interesting octet by determining the last octet that contains the last a 1.
(2) Determine the block size by subtracting decimal value of interesting octet from 256. 4
(3) Determine the first subnet by setting all borrowed bits to 0.
(4) Determine the additional subnet by counting block size in the interesting octet.

A 27-bit subnet mask is applied to a given IP address of 192.168.87.0 /24. To calculate the created subnet masks, the following steps should be considered:

(1) The subnet mask (in binary) is 11111111.11111111.11111111.11100000. 4
(2) The decimal value of the fourth octet is 224 (11100000 in decimal). 3
The block size is 32(256 – 224 = 32).
(3) The first subnet is 192.168. 87.0 /27.
(4) Counting 32(block size) in the fourth octet (interesting octet) to calculate the remaining subnets.

192.168.87.0

192.168. 87.32

192.168. 87.64

192.168. 87.96 4

192.168. 87.128

192.168. 87.160

192.168. 87.192

192.168. 87.224

Subnet Address	Direct Broadcast Address	Usable IP Addresses
192.168.87.0	192.168.87.31	192.168.87.1 - 192.168.87.30
192.168.87.32	192.168.87.63	192.168.87.33 - 192.168.87.62
192.168.87.64	192.168.87.95	192.168.87.65 - 192.168.87.94
192.168.87.96	192.168.87.127	192.168.87.97 - 192.168.87126
192.168.87.128	192.168.87.159	192.168.87.129 - 192.168.87.158
192.168.87.160	192.168.87.191	192.168.87.161 - 192.168.87.190
192.168.87.192	192.168.87.223	192.168.87.193 - 192.168.87.222
192.168.87.224	192.168.87.255	192.168.87.225 - 192.168.87.254

6

Usable IP address ranges for the 192.168. 87.0 /27 subnets