An Introduction to Database Systems

Chapter 1. An Overview of Database Management

spark@dblab.sogang.ac.kr

1.1 An Introductory Example

- a Database System
  - a computerized record-keeping system
- database
  - a kind of electronic filing cabinet
  - a repository for a collection of computerized data files
- operations
  - add, insert, retrieve, update, delete, remove(existing files)
- tables, records(logical), SQL

spark@dblab.sogang.ac.kr

1.1 An Introductory Example

the Wine Cellar Database(Cellar file)

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardony	Buena Vista	2001	1	2003
3	Chardony	Geyser Peak	2001	5	2003
6	Chardony	Stonestreet	2000	4	2002
12	Jo. Riesling	Jekel	2002	1	2003
21	Fume Blanc	Ch. St. Jean	2001	4	2003
22	Fume Blanc	Robt. Mondavi	2000	2	2002
30	Gewurztraminer	Ch. St. Jean	2002	3	2003
43	Cab. Sauvignon	Windsor	1995	12	2004
45	Cab. Sauvignon	Geyser Peak	1998	12	2006
48	Cab. Sauvignon	Robt. Mondavi	1997	12	2008
50	Pinot Noir	Gary Farrell	2000	3	2003
51	Pinot Noir	Stemmler	1997	3	2004
52	Pinot Noir	Dehlinger	1999	2	2002
58	Merlot	Clos du Bois	1998	9	2004
64	Zinfandel	Lytton Spring	1998	9	2007
72	Zinfandel	Rafanelli	1999	2	2007

spark@dblab.sogang.ac.kr

1.1 An Introductory Example

- sample retrieval against the wine cellar database
  - retrieval :

```
SELECT    WINE, BIN, PRODUCER
FROM      CELLAR
WHERE     READY = 2004;
```

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

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

spark@dblab.sogang.ac.kr

1.1 An Introductory Example

- INSERT, UPDATE, DELETE examples

- inserting new data :

```
INSERT
INTO        CELLAR (BIN#, WINE, PRODUCER, YEAR, BOTTLES, READY)
            VALUES(53, 'Pinot Noir', 'Saintsbury', 2001, 6, 2005) ;
```

- changing existing data :

```
UPDATE      CELLAR
SET         BOTTLES = 4
WHERE       BIN#=3;
```

- deleting existing data :

```
DELETE
FROM        CELLAR
WHERE       BIN#=2;
```

- Tables (relational tables)
- Rows/columns
- Data type
- Primary key
- Operations (SELECT/INSERT/UPDATE/DELETE)

spark@dblab.sogang.ac.kr

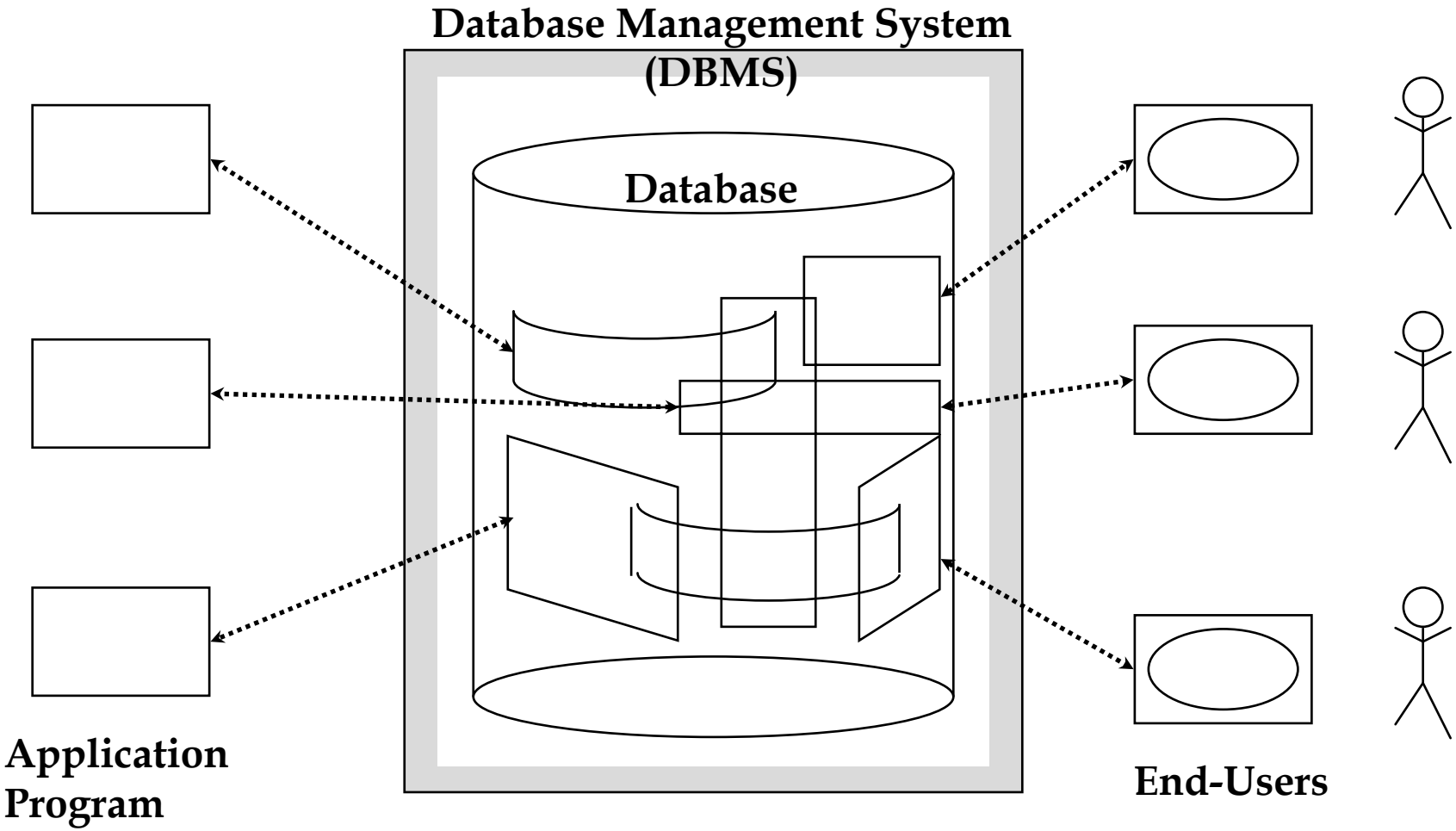
1.2 What is a Database System ?

- **Database System**
  - **a computerized record-keeping system**
  - **components**
    - **data**
    - **hardware**
    - **software**
    - **users**
- **Data**
  - **multi-user system ⁝ single-user system(visible)**
  - **data in database system**
    - ***integrated***
    - ***shared***
    - **EMPLOYEE and ENROLLMENT files**

spark@dblab.sogang.ac.kr

1.2 What is a Database System ?

Simplified Picture of a Database System



spark@dblab.sogang.ac.kr

1.2 What is a Database System ?

- **Hardware**
  - **secondary storage volumes(moving-head magnetic disks) that are used to hold the stored data, together with the associated I/O devices(drives, etc.), device controller, I/O channel,...**
  - **the processor(s) and associated main memory**
  - ***physical organization of Database***
- **Software**
  - **DBMS**

    ; shielding of database users from hardware-level details
- **User**
  - **application programmer(batch, online)**
  - **end-user**
    - **interface**
      - **query language processor(built-in) : command-driven**
      - **menu or form-driven**
  - **DBA**

spark@dblab.sogang.ac.kr

1.3 What is a Database?

- **Persistent Data (database data)**
  - **Differ from ephemeral data (I/O data, control statements, work queue, sw control blocks, immediate results, …)**
  - **Removed from the DB only by some explicit request to DBMS**
  - **Database**
    - **A collection of stored persistent data used by the application systems of some particular enterprise**
      - *manufacturing company*
      - *bank*
      - *hospital*
      - *university*
      - *government department*

      - *Product data*
      - *Account data*
      - *Patient data*
      - *Student data*
      - *Planning data*

spark@dblab.sogang.ac.kr

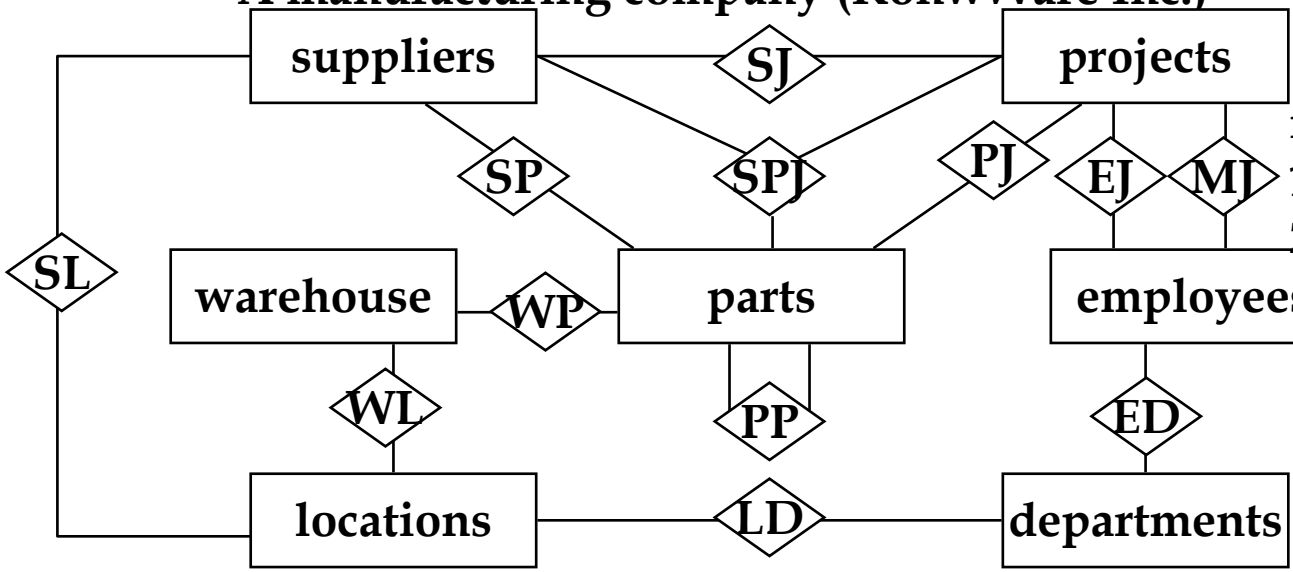
1.3 What is a Database?

- **Entities and Relationships**
  - **entities**
    - **independent**
    - **distinguishable object**
  - **relationship** **unary relationship(bill-of-materials)**
    binary relationship
    ternary relationship(suppliers-parts-projects)

spark@dblab.sogang.ac.kr

1.3 What is a Database?

- **Entities and Relationships**
  - **A manufacturing company (KonwWare Inc.)**



multiple relationship
1. assigned to
2. manager of

ex) SP relationship - bidirectional

- Given a supplier, get the parts supplied by that supplier
- Given a part, get the suppliers who supply that part

point : They are just as much a part of the data as are the basic entities.

- **ERD**

spark@dblab.sogang.ac.kr

1.3 What is a Database?

- **Entities and Relationships**
  - **connection trap**

    (a) "Snith supplies monkey wrenches to the Manhanttan project" tells us more than the following three statements do :

    (b) "Smith supplies monkey wrenches"

    (c) "Monkey wrenches are used in the Manhanttan project"

    (d) "The Manhanttan project is supplied by Smith"
  - **bill-of-material relationship**
  - **A relation can be regarded as an entity in its own right.**

- **Properties**
  - **Entities(relationships) can b regarded as having properties corresponding to the information we wish to record about them.**

    Ex) Suppliers – locations, name, city, …

spark@dblab.sogang.ac.kr

1.3 What is a Database?

- **Data and Data Models**
  - Data is really given facts. A given fact corresponds to what logicians called a true proposition.

    Ex) " Supplier S1 is located in London"
  - A Database is really a collection of such true propositions.
- **The relational model of data**
  - 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 support the process of inferring additional true propositions from the given ones.
- **What is a data model ?**
  - 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.**

spark@dblab.sogang.ac.kr

1.3 What is a Database?

- **Data and Data Models**
  - **The model is what users have to know about, the implementation is what users do not have to know about.**
  - **Data model : two meanings**
    - **Programming language**
    - **Specific program**

spark@dblab.sogang.ac.kr

1.4 Why Database ?

- advantages in single user database

  (over paper-based method of record-keeping)

  - compactness
  - speed
  - less drudgery
  - currency
- benefits in multi-user environment
  - centralized control of its data

spark@dblab.sogang.ac.kr

1.4 Why Database ?

- data administration and database administration
- A Manager :
  - Data Administrator(DA)
    - a person who understands the data, and the needs of the enterprise with respect to the data, at a senior management level
  - DA's job
    - to decide what data should be stored in the database in the first place, and to establish policies(e.g., data security) for maintaining and dealing with that data once it has been stored
- A Technician :
  - Database Administrator(DBA)
    - a person responsible for implementing the data administrator's decisions
  - DBA's job
    - to create the actual database and to implement the technical controls needed to enforce the various policy decisions made by the data administrator

spark@dblab.sogang.ac.kr

1.4 Why Database ?

- advantages that accrue from the notions of centralized control of data
  - Data sharing
    - not only existing applications, but also new application
  - Reduction of redundancy
    - integration by DBA
    - controlled redundancy
  - Avoidance of inconsistency
    - corollary of the reduction of redundancy
    - if redundancy is not removed but controlled
      - system : propagate update
  - Transaction support

spark@dblab.sogang.ac.kr

1.4 Why Database ?

- advantages that accrue from the notions of centralized(cont'd)
  - Integrity
    - data in the database is accurate
  - Security
    - Having the complete jurisdiction over the data, the DBA
      - ensure that the only means of access to the database is through the proper channels
      - define security rules to be checked for the sensitive data(read, write, modify)
  - Balancing the conflicting requirement
    - representation of data in storage
  - Standards
    - are observed in the representation of the data
    - aid to data interchange or migration of data between systems
  - Data independence

spark@dblab.sogang.ac.kr

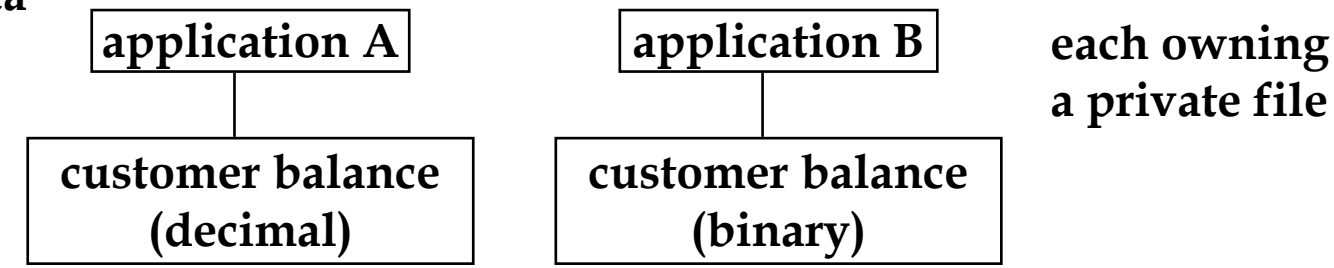
1.5 Data Independence

- **Data dependence**
  - **knowledge of that data organization and access technique is built into the application logic and code**
    - **storage structure (how the data is physically stored)**
    - **access strategy(how it is used)**

spark@dblab.sogang.ac.kr

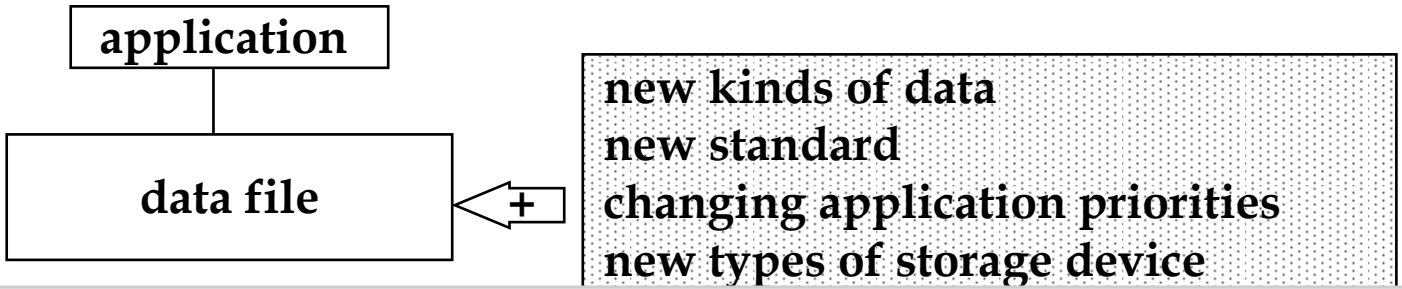
1.5 Data Independence

- the reason that "data dependence" is not proper
  - Different application will need different views of the same data



But, DBMS is able to perform all conversion

  - DBA must have the freedom to change the storage structure or access strategy in response to changing requirements without having to modify existing applications



such changes require corresponding changes to be made to programs

spark@dblab.sogang.ac.kr

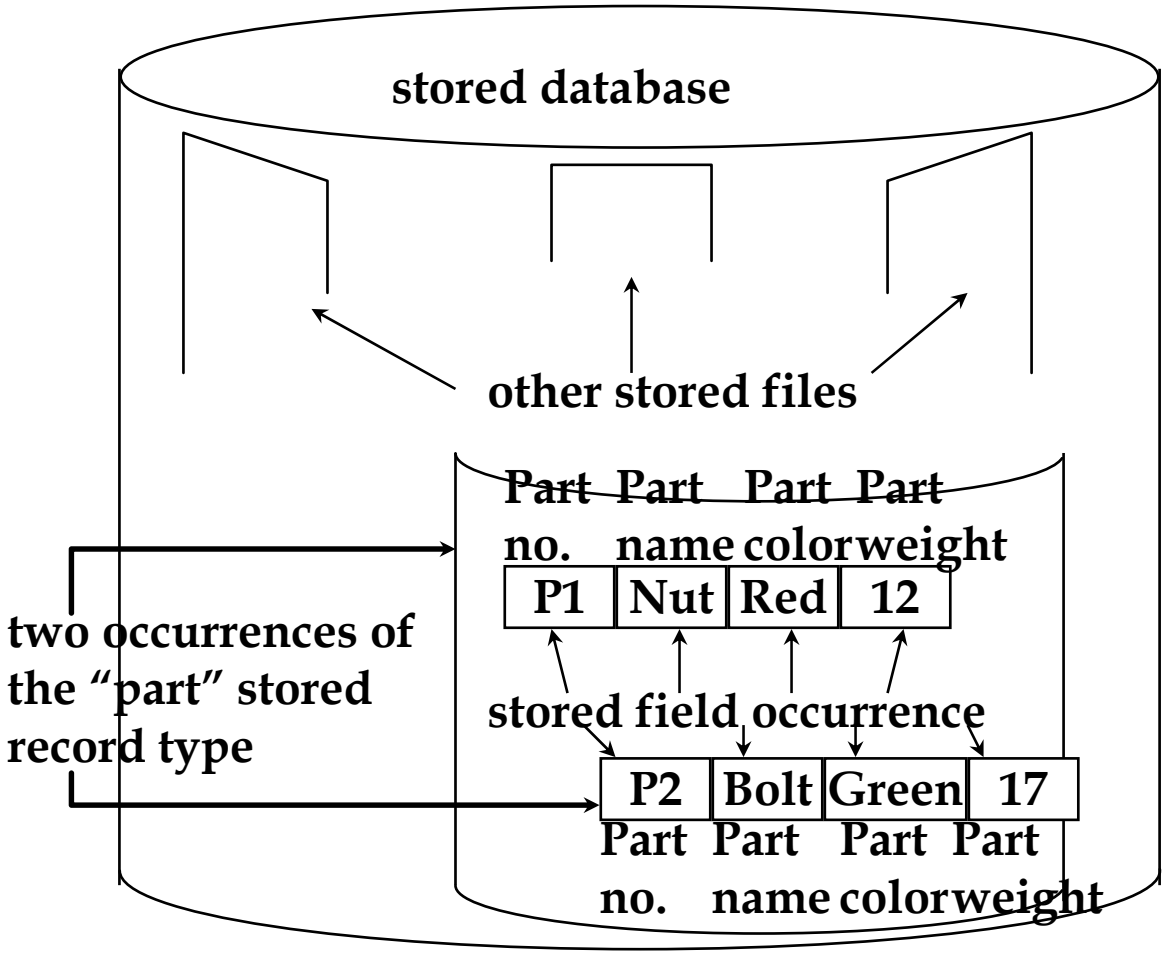
1.5 Data Independence

- Data Independence
  - the immunity of applications to change in physical representation (storage structure) and access technique
- some examples
  - of the types of change that DBA might wish to make
  - terminology
    - stored field
      - the smallest unit of stored data
    - stored record
      - a collection of related stored fields
    - stored file
      - the collection of all occurrences of one type of stored record

spark@dblab.sogang.ac.kr

1.5 Data Independence

Stored fields, records, and files

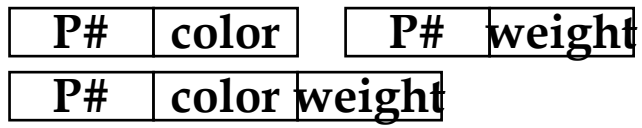


spark@dblab.sogang.ac.kr

1.5 Data Independence

- Differences between what the application sees and what is actually stored might be quite considerable
  - representation of numeric data
    - numeric as a string or a packed decimal
    - Choose a base(binary/decimal) scale(fixed/float)
      mode(real/complex)
      precision(number of digits)

    units for numeric data
    - inches or centimeter
  - Data encoding
    - 1=Red, 2=Blue, …
  - data materialization
    - direct or indirect(*virtual field*)
  - structure of stored records
  - structure of stored files



spark@dblab.sogang.ac.kr

1.6 Relational Systems and Others

- relational approach
  - developed since late 1970
  - the dominant trend in the market place today
  - based on aspects of mathematics
  - an ideal vehicle for teaching the concepts and principles of database systems
- What does it mean that a system is relational ?
  1. The data is perceived by the user as tables
  2. The operators at the user's disposal(for data retrieval) are operators that generate new tables from old

spark@dblab.sogang.ac.kr

1.6 Relational Systems and Others

- Data structure and operators in a relational system

(a) given table: CELLAR

WINE	YEAR	BOTTLES
Chardonnay	1996	4
Flume Blanc	1996	2
Pinet Noir	1993	3
Zinfandel	1994	9

(b) Operators(examples) :

1. Row subset :

```
SELECT WINE, YEAR, BOTTLES
FROM   CELLAR
WHERE  YEAR > 1995 ;
```

Result

WINE	YEAR	BOTTLES
Chardonnay	1996	4
Flume Blanc	1996	2

2. Column Subset :

```
SELECT WINE, BOTTLES
FROM   CELLAR ;
```

Result

WINE	BOTTLES
Chardonnay	4
Flume Blanc	2
Pinet Noir	3
Zinfandel	9

spark@dblab.sogang.ac.kr

1.6 Relational Systems and Others

- **Others**
  - **categorized according to(data structures and operators) they present to the user**
  - ***hierarchical system(IMS) : tree structure***
  - ***inverted systems(CA-DATACOM/DB)***
  - ***network systems(CA-IDMS/DB)***
  - ***Relational systems(DB2-IBM, Ingres II-CA, Informix Dymaic Server-Informix, SQL-Server-MS, Oracle 8i-Oracle, Sybase Adaptive Server-Sybase)***
  - ***Object, object/relational systems***
  - ***Multi-dimensional system(DW/OLAP)***
  - ***Logic-based system***

spark@dblab.sogang.ac.kr

DBMS

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use
- Database Applications:
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Online retailers: order tracking, customized recommendations
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives

spark@dblab.sogang.ac.kr

Purpose of Database Systems

- In the early days, database applications were built directly on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

spark@dblab.sogang.ac.kr

Purpose of Database Systems

- Drawbacks of using file systems (cont.)
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent accessed needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance and updating it at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data
- Database systems offer solutions to all the above problems

spark@dblab.sogang.ac.kr

History of Database Systems

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provide only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allow direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
  - High-performance (for the era) transaction processing

spark@dblab.sogang.ac.kr

History of Database Systems

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce
- 2000s:
  - XML and XQuery standards
  - Automated database administration

spark@dblab.sogang.ac.kr