

# ជំពូកទី ១

# Introduction to Databases

## 1. Introduction

Database បច្ចុប្បន្ននេះបានក្លាយទៅជាផ្នែកមួយចាំបាច់ក្នុងជីវភាពប្រចាំថ្ងៃ ដែលយើងមិនបានដឹងថាយើងកំពុងប្រើប្រាស់វា ប៉ុន្តែមុននឹងចាប់ផ្តើមពិពណ៌នាអំពី database យើងគួរតែត្រួតពិនិត្យទៅលើ applications នៃ database systems ជាមុនសិន។

នៅពេលដែលយើងទិញទំនិញពី local supermarket ពេលនោះ checkout assistant នឹងដំណើរការ bar code reader ទៅលើទំនិញនោះដើម្បីទទួលបានតំលៃ ហើយវាធ្វើការតភ្ជាប់ជាមួយ database application program ដើម្បីកាត់បន្ថយចំនួន items នោះនៅក្នុង stock ។ ប៉ុន្តែប្រសិនបើចំនួននោះ ស្ថិតនៅក្រោម reorder level នោះប្រព័ន្ធនឹងធ្វើការកម្រងទិញដោយស្វ័យប្រវត្តដើម្បីទទួលបាន items ថែមទៀត។ ប្រសិនបើអតិថិជនទូរស័ព្ទទៅកាន់ supermarket ដើម្បីសាកសួរព័ត៌មានអំពី item ណាមួយនោះ checkout assistant នឹងមើលថាតើ items នោះនៅក្នុង stock មាននៅសល់ដែររឺទេ បើនៅសល់តើវាសល់ប៉ុន្មានដោយដំណើរការលើ application program ។

នៅពេលដែលយើងចង់ធ្វើដំណើរកំសាន្តទៅលេងប្រទេសណាមួយ ដើម្បីកំសាន្តនៅថ្ងៃឈប់សំរាក ពេលនោះយើងត្រូវសាកសួរព័ត៌មានពី travel agency ហើយបន្ទាប់មក travel agency ធ្វើការប្រើប្រាស់ database ដើម្បីពិនិត្យមើលជើងយន្តហោះ និងថ្លៃឈប់សំរាកលំអិត។ នៅពេលដែលយើងយល់ឃើញថា វាសមស្របទៅនឹងតម្រូវការរបស់យើងពេលនោះយើងធ្វើការអាចកក់កន្លែងកំសាន្តថ្ងៃនោះ ហើយ database system ធ្វើការរក្សាទិន្នន័យការកក់នោះ។ ក្នុងករណីនេះប្រព័ន្ធត្រូវតែប្រាកដថាគ្មាន agents 2 ផ្សេងគ្នាធ្វើការកក់ទៅកាន់ថ្ងៃឈប់សំរាក និងកៅអីដូចគ្នាបានទេ។ ឧទាហរណ៍: ប្រសិនបើមានកៅអីមួយគត់នៅសល់សំរាប់ការធ្វើដំណើរដោយយន្តហោះពី London ទៅ New York និងមាន agents 2 ធ្វើការកក់ទុកជាមុនកៅអីនោះក្នុងពេលតែមួយ ពេលនោះប្រព័ន្ធត្រូវតែពិនិត្យមើលលើហេតុការណ៍នេះដោយអនុញ្ញាតិអោយ agent តែម្នាក់គត់អាចកក់បាន ហើយផ្តល់ព័ត៌មានដល់អ្នកផ្សេងទៀតថា គ្មានកៅអីនៅសល់ទេ។

នៅពេលយើងចូលបណ្តាលយើងដើម្បីមើល រឺក៏ខ្លីសៀវភៅ នៅទីនោះប្រហែលជាអាចមាន database ផ្ទុកព័ត៌មានលំអិតរាល់សៀវភៅទាំងអស់នៅក្នុងបណ្តាលយើង, ព័ត៌មានលំអិតនៃ users, ការបម្រុងទុកជាមុន (reservation)...។ លើសពីនេះទៀតមាន computerized index ដែលធ្វើអោយ users អាចស្វែងរកសៀវភៅយ៉ាងលឿនតាមរយៈឈ្មោះអ្នកនិពន្ធ, ចំណងជើង រឺប្រភេទរបស់វា។ ប្រព័ន្ធធ្វើការគ្រប់គ្រងទៅលើការខ្ចីរបស់ users ហើយប្រសិនបើដល់ថ្ងៃកំណត់ ត្រូវប្រគល់សៀវភៅត្រលប់វិញ user នៅតែមិនទាន់ឃើញយកសៀវភៅមកប្រគល់ ប្រព័ន្ធត្រូវផ្ញើ reminder ទៅកាន់ user ដើម្បីរំលឹក។

ប្រសិនបើក្រលេកទៅមើលសាកលវិទ្យាល័យវិញ នៅទីនោះក៏មាន database system ផ្ទុក ព័ត៌មានអំពីនិស្សិតសិក្សានៅទីនោះទាំងអស់, វគ្គសិក្សា, ព័ត៌មានលំអិតបុគ្គលិក, មុខវិជ្ជាសិក្សាកាលពី ឆ្នាំមុនៗ, លទ្ធផលប្រលងបញ្ចប់ឆមាសនីមួយៗ...។ នៅពេលដែលចង់ធ្វើការស្វែងរកឈ្មោះសិស្ស ណាម្នាក់ វាមានលក្ខណៈងាយស្រួល ហើយចំណាយរយៈពេលខ្លីទៀត។

## 2. Traditional File-Based Systems

មានសៀវភៅ database ភាគច្រើនដែលបានសរសេរទៅលើ database system ក៏ប៉ុន្តែ មិនភ្លេចសរសេររំលឹកឡើងវិញពី file-based system ដែលប្រើប្រាស់យ៉ាងទូលំទូលាយមុនពេល កកើត database system។ ទោះបីជា file-based system វាហួសសម័យទៅហើយក៏ដោយ ប៉ុន្តែមានហេតុផលមួយចំនួនក្នុងការសិក្សាវាគឺ

- យល់ដឹងពីបញ្ហាកើតមានជាមួយ file-based system ហើយអាចការពារមិនអោយ បញ្ហានេះកើតឡើងជាមួយ database system ។
- ប្រសិនបើចង់ធ្វើការបំប្លែងពី file-based system ទៅ database system វាមានភាពចាំបាច់ត្រូវដឹងពីដំណើរការរបស់ file system ។

### 2. 1. File-Based Approach

**File-based system** A collection of application programs that perform services for the end-users such as the production of reports. Each program defines and manages its own data.

File-based system គឺជាបណ្តុំនៃ application programs ផ្តល់សេវាទៅអោយ end-users ដូចជាការបង្កើតរបាយការណ៍ជាដើម។ Application program នីមួយៗកំណត់ និងគ្រប់គ្រងទិន្នន័យរៀងៗខ្លួន។ កាលពីមុនគេប្រើប្រាស់ file-based system ដើម្បីបំប្លែងពី ប្រព័ន្ធប្រើប្រាស់ដោយដៃ (manual filing system) ទៅជាប្រព័ន្ធប្រើប្រាស់កុំព្យូទ័រ ឧទាហរណ៍:

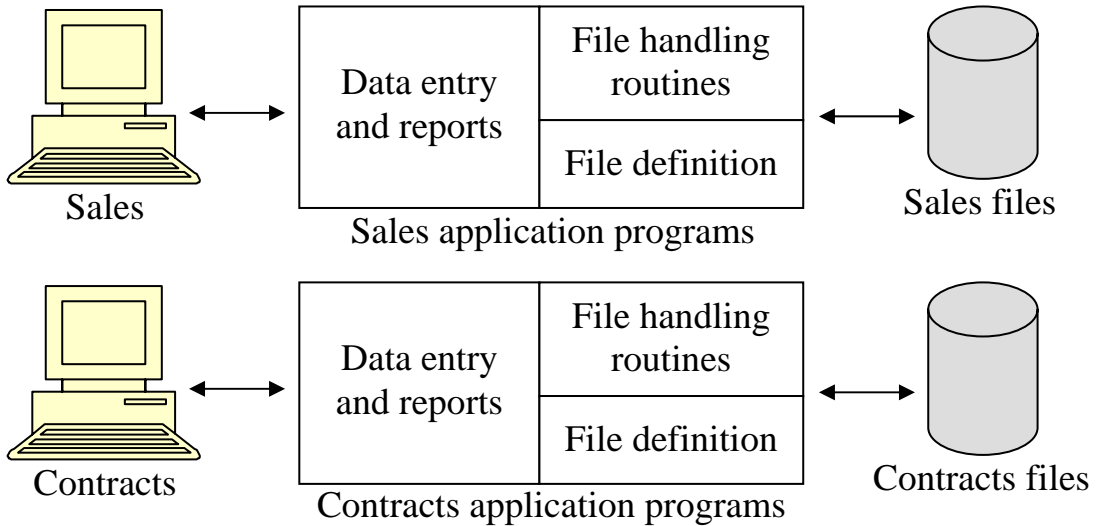
នៅក្នុង organization មួយគេបង្កើត manual files ជាច្រើនដើម្បីរក្សាទុកព័ត៌មានក្នុងការ  
ទំនាក់ទំនងក្នុងក្រៅចំពោះ project, product, task, client, or employee ។  
ដើម្បីការពារសុវត្ថិភាព files ជាច្រើននោះយើងធ្វើការបិទអ៊ីសំគាល់ និងរក្សាទុកក្នុងទូរជាច្រើន  
ហើយទូរទាំងនោះត្រូវតែចាក់សោរ វិស្វិតនៅក្នុងតំបន់ដែលមានសុវត្ថិភាព លើសពីនេះទៀត  
គេប្រហែលជាតំរៀប files ទាំងនោះរៀងគ្នាតាមលំដាប់កើន រឺចុះ ដើម្បីអោយមានភាពងាយស្រួល  
នៅពេលស្វែងរកអោយបានលឿន បើមិនដូចនោះទេយើងត្រូវស្វែងរកគ្រប់ files ទាំងអស់  
ធ្វើការអោយខាតពេលវេលា។ ជំនួសអោយវិធីសាស្ត្រប្រើប្រាស់ខាងលើ គេប្រើប្រាស់ index  
ដើម្បីជួយសំរួលលឿនដំណើរការស្វែងរកអោយកាន់តែលឿន។

Manual filing system ដំណើរការបានល្អនៅពេលដែលចំនួន items រក្សាទុកមានទំហំ  
តូច ហើយម្យ៉ាងទៀត manual filing system ក៏អាចដំណើរការបានដែរនៅពេលដែលចំនួន  
items មានទំហំធំ រឺច្រើនហើយយើងគ្រាន់តែរក្សា និងទទួលវា (store and retrieve)  
តែប៉ុណ្ណោះ។ ទោះបីជាយ៉ាងណាក៏ដោយ ការប្រើប្រាស់ manual filing system មានបញ្ហា  
នៅពេលដែលធ្វើការប្រើប្រាស់ទិន្នន័យរក្សាទុកក្នុង files ជាច្រើន ដែលមានទីតាំងខុសៗគ្នា  
ឧទាហរណ៍: Real estate agent's office ប្រហែលជាមាន file ផ្សេងៗគ្នាសំរាប់ property  
for sale or rent, buyer and renter និង staff ចូរឆ្លើយទៅនឹងសំណួរមួយចំនួន  
ដូចខាងក្រោមតើវាពាក់ព័ន្ធនឹងអ្វីខ្លះ

- តើមាន three-bedroom properties ណាខ្លះសំរាប់លក់ដែលមានសួនច្បារ  
និងយានជួន?
- តើមាន flats ណាខ្លះសំរាប់ជួលដែលមានទីតាំងឆ្ងាយពីទីក្រុង 3 Kilometers?
- តើតំលៃលក់ផ្ទះជាមធ្យមប៉ុន្មាន?
- តើតំលៃជួលជាមធ្យមចំពោះ two-bedroom flat ថ្លៃប៉ុន្មាន?
- តើប្រាក់ខែបុគ្គលិកប្រចាំឆ្នាំសរុបប៉ុន្មាន?

ដោយយោងទៅលើការរីកចំរើនជារៀងរាល់ថ្ងៃ គេតែងតែចង់បានព័ត៌មានបន្ថែមទៀត  
ឧទាហរណ៍: របាយការណ៍ការលក់ រឺជួលប្រចាំឆ្នាំ, ប្រចាំថ្ងៃ, ប្រចាំត្រីមាសជាដើម ដែល manual  
filing system មិនមានលក្ខណៈគ្រប់គ្រាន់សំរាប់ការងារនេះ។

ឧបមាថា យើងមាន sale department ទទួលបន្ទុកលក់ និងជួល properties ព្រមទាំង  
contract department ទទួលបន្ទុកធ្វើកុងត្រាជួលលើ property for rent ដូចរូបខាងក្រោម  
ដែលមានផ្ទុកទិន្នន័យស្តួនជាច្រើនក្នុង departments ទាំងនេះ។



*Sales Files*

**Property\_for\_Rent**(Property Number, Street, Area, City, Post Code, Property Type, Number of Rooms, Monthly Rent, Owner Number)

**Owner**(Owner Number, First Name, Last Name, Address, Telephone Number)

**Renter**(Renter Number, First Name, Last Name, Address, Telephone Number, Preferred Type, Maximum Rent)

*Contracts Files*

**Lease**(Lease Number, Property Number, Renter Number, Monthly Rent, Payment Method, Deposit, Paid, Rent Start Date, Rent Finish Date, Duration)

**Property\_for\_Rent**(Property Number, Street, Area, City, Post Code, Monthly Rent)

**Renter**(Renter Number, First Name, Last Name, Address, Telephone Number)

(រូប 1.1)

**2. 2. Limitations of the File-Based Systems**

ការប្រើប្រាស់ File-Based System មានកំរិតដូចខាងក្រោម:

- Separation and isolation of data. វាមានភាពលំបាកក្នុងការប្រើប្រាស់ទិន្នន័យ (access data) ដែលស្ថិតនៅក្នុង files ផ្សេងៗគ្នា។
- Duplication of data. ឧបមាថា ក្រុមហ៊ុនមួយមាន 2 departments គឺ Sales និង Contacts ដែលត្រូវការប្រើប្រាស់ file-based system ដើម្បីគ្រប់គ្រង យើងសង្កេតឃើញថា ទាំង Sales និង Contacts សុទ្ធតែផ្ទុក Property និង Renter ដូចគ្នាជាហេតុនាំអោយមាន ភាពស្ងួតទិន្នន័យ។ ការគ្រប់គ្រងភាពស្ងួតទិន្នន័យមិនបាន បណ្តាលអោយមានបញ្ហាដូចជា:

- + Duplication is wasteful. ភាពស្ងួតទិន្នន័យបានធ្វើអោយខាត ផលប្រយោជន៍ជាច្រើនដូចជា កំលែងថ្លៃ, ខាតពេលវេលាបញ្ចូលទិន្នន័យ, space ត្រូវការធំ។
- + ភាពស្ងួតទិន្នន័យអាចនាំទៅអោយមាន loss of data integrity ជាហេតុនាំ អោយមិនយូរប៉ុន្មានទិន្នន័យនឹងមានភាពមិនស៊ីសង្វាក់គ្នា (inconsistency) ។
  - Data Dependence. Physical structure និង storage នៃ data files និង records ត្រូវបានកំណត់ក្នុង application code ដូចនេះនៅពេលធ្វើការកែប្រែ structure មានភាពលំបាក។
  - Incompatible file format. Structure នៃ files ត្រូវបានបង្កប់ក្នុង application programs ដូចនេះ structures អាស្រ័យទៅនឹង application programming languages មានន័យថា structure នៃ files សរសេរក្នុង Pascal programming language ប្រហែលជា អាចខុសគ្នាជាមួយ structure នៃ files សរសេរក្នុង C programming language ដែល បណ្តាលអោយមានភាពលំបាកក្នុងការធ្វើជាក្រុម។
  - Fixed queries/proliferation of application programs. File-based system អាស្រ័យលើ application programmers ដែលបានសរសេរ queries ទុកអោយ ប្រើប្រាស់ ប៉ុន្តែបើសិនជាមាន query ថ្មីកើតឡើងភ្លាមពេលនោះយើងត្រូវរក application programmers សរសេរអោយ។

### 3. Database Approach

ដោយឃើញថា file-based system មានភាពលំបាកទើបធ្វើអោយក៏កើត database system ឡើងដើម្បីជួយសំរួលបញ្ហាទាំងនេះ។

#### 3. 1. The Database

**Database** A shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.

Database គឺជាបណ្តុំទិន្នន័យរក្សាទុកក្នុងទម្រង់មួយងាយស្រួលប្រើប្រាស់ ហើយត្រូវបាន គេកសាងសំរាប់បំពេញតំរូវការព័ត៌មាននៃអង្គភាពណាមួយ។ Database ជា a single, large repository of data ដែលត្រូវគេបង្កើតតែម្តង ហើយត្រូវបានគេប្រើប្រាស់ក្នុងពេលតែមួយដោយ departments និង users ជាច្រើន។ រាល់ទិន្នន័យទាំងអស់ត្រូវបានច្របាច់បញ្ចូលក្នុង database ដោយផ្អែកទិន្នន័យស្តួនអប្បបរិមា ហើយលើសពីនេះទៀត database មិនមែនជាកម្មសិទ្ធិរបស់

department ណាមួយទេ តែវាជា shared corporate resource ។ Database មិនគ្រាន់តែផ្ទុក organization's operational data ថែមទាំងផ្ទុក description of data (Description of data called **System catalog** or **data dictionary** or **meta data**.) ។

ម្យ៉ាងវិញទៀត database system បានបំបែក definition of data (data structures) ចេញអោយដាច់ពី application programs ហេតុដូច្នេះប្រសិនបើប្រែទម្រង់ data structures ចំពោះ database វាមិនប៉ះពាល់ដល់ application programs ទេ។ ឧទាហរណ៍: ប្រសិនបើយើងកែប្រែទំហំកូអក្សរនៃ Student\_Name field ពី 20 ទៅ 30 កូអក្សរ វិបផ្តិត file ថ្មី នោះ application programs កំពុងធ្វើការប្រើប្រាស់ គ្មានផលប៉ះពាល់ទេ ទោះបីជាយ៉ាងណាក៏ដោយ ប្រសិនបើយើងលុប field ចេញពី file ដែល application program ប្រើប្រាស់ នោះ application program ទទួលរងការផលប៉ះពាល់ដោយសារការកែប្រែនេះ ហើយត្រូវតែធ្វើការកែប្រែតាមដែរ។

នៅពេលដែលយើងធ្វើការវិភាគព័ត៌មានអំពី organization ណាមួយ យើងមានបំណង កំណត់នូវ entities, attributes និង relationships ដូចនេះគេអាចនិយាយថា database ធ្វើការបង្ហាញពី entities, attributes និង relationships between entities ។

**3. 2. The Database Management Systems (DBMS)**

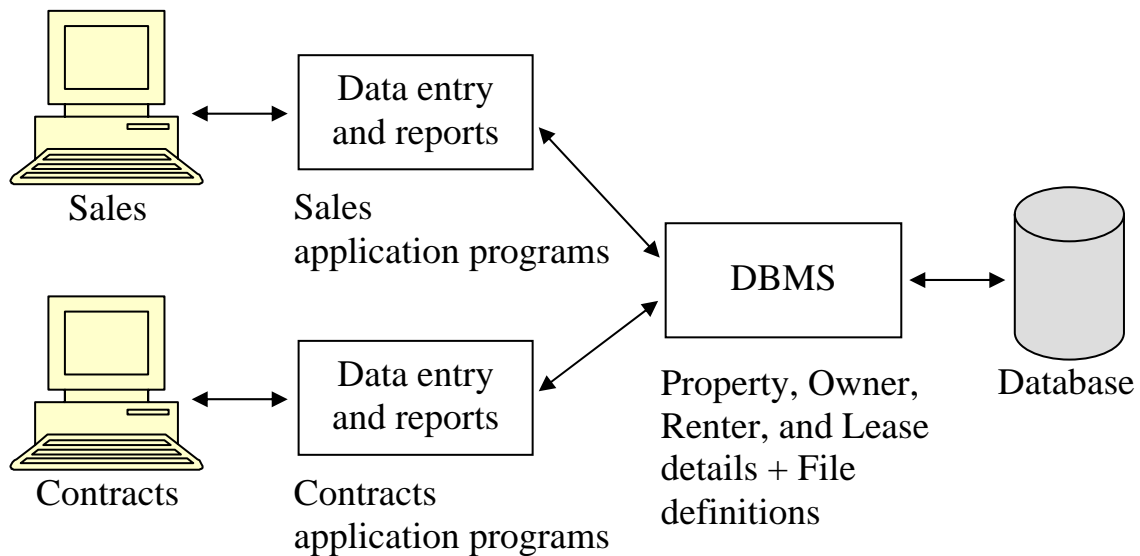
**DBMS** is a software that enables users to define, create, and manipulate the database and provides controlled access to this database.

DBMS គឺជា software ដែលធ្វើអោយអ្នកប្រើប្រាស់មានលទ្ធភាពកំណត់, បង្កើត និងធ្វើការប្រើប្រាស់ database ព្រមទាំងផ្តល់ភាពគ្រប់គ្រងការចូលប្រើប្រាស់ទៅកាន់ database ផងដែរ។

ជាទូទៅ DBMS ផ្តល់នូវភាពងាយស្រួលជាច្រើនដូចជា:

- អនុញ្ញាតិអោយអ្នកប្រើប្រាស់កំណត់ database តាមរយៈ Data Definition Language (DDL) ។ DDL អនុញ្ញាតិអោយអ្នកប្រើប្រាស់កំណត់ Data types, Structures និង Constraints ទៅលើទិន្នន័យរក្សាទុកក្នុង database ។
- អនុញ្ញាតិអោយអ្នកប្រើប្រាស់បញ្ចូល, កែប្រែ, លុប និងទទួល (ស្វែងរក) ទិន្នន័យក្នុង database តាមរយៈ Data Manipulation Language (DML) ។
- ផ្តល់ភាពគ្រប់គ្រងការចូលប្រើប្រាស់ database

- + Security system. ការពារអ្នកប្រើប្រាស់ ដែលគ្មានការអនុញ្ញាតិ (Unauthorized users) ពីការចូលប្រើប្រាស់ database ។
- + Integrity system. ថែរក្សាភាពស៊ីសង្វាក់គ្នានៃទិន្នន័យ (Consistency of stored data) ។
- + Concurrency control system. អនុញ្ញាតិអោយអ្នកប្រើប្រាស់ជាច្រើន ចូលប្រើប្រាស់ទិន្នន័យក្នុង database ក្នុងពេលតែមួយ។
- + Recovery control system. ធ្វើការ restore database ទៅកាន់ ភាពស៊ីសង្វាក់គ្នានៃទិន្នន័យពីមុន (previous consistent state) ប្រសិនបើ មានបញ្ហា។
- + User-accessible catalog. ផ្គុំការពណ៌នាទិន្នន័យក្នុង database ។



**Property\_for\_Rent**(Property Number, Street, Area, City, Post Code, Property Type, Number of Rooms, Monthly Rent, Owner Number)

**Owner**(Owner Number, First Name, Last Name, Address, Telephone Number)

**Renter**(Renter Number, First Name, Last Name, Address, Telephone Number, Preferred Type, Maximum Rent)

**Lease**(Lease Number, Property Number, Renter Number, Monthly Rent, Payment Method, Deposit, Paid, Rent Start Date, Rent Finish Date)

(រូប 1.2)

តាមរយៈរូប 1.2 យើងឃើញថាទាំង Sales and Contracts departments ប្រើប្រាស់ application programs ដើម្បីចូលប្រើប្រាស់ database តាមរយៈ DBMS ហើយ application programs ចំពោះ department នីមួយៗដោះស្រាយការបញ្ចូលទិន្នន័យ (data

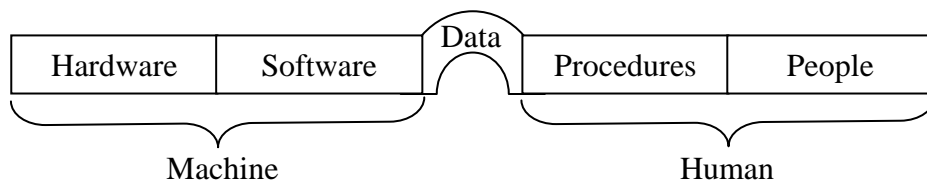
entry), ការថែទាំទិន្នន័យ (data maintenance) និងការបង្កើត reports (generation of reports) ដោយខ្លួនឯង ប៉ុន្តែបើប្រៀបធៀបជាមួយ file-based system យើងឃើញថា physical structure and storage of the data ត្រូវគ្រប់គ្រងដោយ DBMS ។

- Views បានផ្តល់នូវកំរិតនៃសុវត្ថិភាព ដោយមិនបានបង្ហាញទិន្នន័យដែលអ្នកប្រើប្រាស់ មួយចំនួនមិនគួរដឹង។ ឧទាហរណ៍: យើងអាចបង្កើត view ដែលអនុញ្ញាតិអោយ branch manager និង payroll department ឃើញព័ត៌មានលំអិត staff ទាំងអស់ ព្រមទាំង salary details ផងដែរ ប៉ុន្តែចំពោះ staff វិញដែលអាចមើលឃើញតែព័ត៌មានលំអិតរបស់ staff តែមិន អាចឃើញ salary details បានទេ ដូច្នេះយើងត្រូវបង្កើត view មួយផ្សេងទៀតសំរាប់ staff ។

- Views អាចបង្ហាញនូវ consistent, unchanging picture នៃរចនាសម្ព័ន្ធរបស់ database ទោះបីជាមានការកែប្រែ database ក៏ដោយ (ឧទាហរណ៍: បន្ថែម fields, លុប fields, relationships, restructured, files split រឺ renamed)។ ប្រសិនបើលុប រឺបន្ថែម fields ទៅកាន់ database ចំពោះ view ដែលពុំត្រូវការ fields នោះ ដូច្នេះវាគ្មានការប៉ះពាល់ ដល់ view នោះឡើយ។

### 3. 3. Components of the DBMS Environment

DBMS environment ផ្សំឡើងដោយ 5 major components គឺ Hardware, Software, Data, Procedures និង People ដូចរូបខាងក្រោម:



(រូប 1.3)

#### - Hardware

DBMS និង Application ត្រូវការ hardware ជាធាតុចាំបាច់សំរាប់ដំណើរការ ហើយ ការកំរិតកំរូវការរបស់ hardware គឺគិតចាប់ពី a single computer, to a single mainframe, to a network of computers។ ការប្រើប្រាស់ hardware កំរិតណានោះ គឺអាស្រ័យទៅលើកំរូវការរបស់ organization ព្រមទាំង DBMS ដែលប្រើប្រាស់។ គួរកត់សំគាល់ ផងដែរថា មាន DBMS មួយចំនួនដំណើរការលើ hardware រឺ operating system ជាក់លាក់ ណាមួយតែម្តងដូចជា Windows 2000 Server ហើយ DBMS ទាមទារលក្ខខណ្ឌទៅលើ



minimum main memory និង disk space ផងដែរ ប៉ុន្តែការប្រើប្រាស់លក្ខណៈ: minimum ទាំងនេះពុំបានផ្តល់នូវ acceptable performance ទេ។

**- Software**

Software component ផ្ទុកនូវ DBMS software និង application programs រួមគ្នាជាមួយ operating system រួមបញ្ចូលទាំង network software ប្រសិនបើគេប្រើប្រាស់ DBMS នៅលើ network ។ ជាទូទៅ application programs ប្រើនៃសរសេរក្នុង third-generation programming language ដូចជា C, COBOL, Fortran, Ada, Visual Basic រឺប្រើប្រាស់ forth-generation language ដូចជា Structured Query Language (SQL) បង្កប់ (embedded) ក្នុង third-generation programming language ។

**- Data**

Component មានសារៈសំខាន់បំផុតក្នុង DBMS environment គឺ Data ហើយវាគឺជា ស្ថានក្នុងការភ្ជាប់រវាង machine components និង human components ។ Database ផ្ទុកទាំង operational data និង meta-data (data about data) ហើយ structure of the database is called the **schema** ។ Schema ចំពោះ database ខាងក្រោមផ្ទុក tables 4 គឺ Property\_for\_Rent, Owner, Renter និង Lease ។

**Property\_for\_Rent**(Property Number, Address, Area, Post Code, Property Type, Number of Rooms, Monthly Rent, Owner Number)

**Owner**(Owner Number, First Name, Last Name, Address, Telephone Number)

**Renter**(Renter Number, First Name, Last Name, Address, Telephone Number)

**Lease**(Lease Number, Property Number, Renter Number, Payment Method, Deposit, Paid, Rent Start Date, Rent Finish Date)

**- Procedures**

Procedures សំដៅទៅលើ instructions និង rules ដែលគ្រប់គ្រងទៅលើការកសាង និងប្រើប្រាស់ database ។ អ្នកប្រើប្រាស់ប្រព័ន្ធ (The users of the system) និង staff ដែលគ្រប់គ្រង database តែងតែត្រូវការនូវ documented procedures ចំពោះថា តើយើងត្រូវប្រើប្រាស់ រឺដំណើរការប្រព័ន្ធដោយរបៀបណា។ Procedure ផ្ទុកនូវ instructions តើត្រូវ...ដោយរបៀបណា។

- + Log on to the DBMS.
- + Use a particular DBMS facilities or application program.
- + Start and stop the DBMS.
- + Make backup copies of the database.

- + Handle hardware or software failures.
- + Change the structure of a table, reorganize the database across multiple disks, improve performance, or archive data to secondary storage.

**- People**

ចូរមើលចំនុច 4 ខាងក្រោម។

**4. Roles in the Database Environment**

នៅក្នុងផ្នែកខាងលើ យើងបានដឹងអំពី components ទាំង 5 នៃ DBMS environments ហើយយើងបានរៀបរាប់ 4 components រួចហើយ នៅសល់តែ People ប៉ុណ្ណោះ។ យើងអាចកំណត់ People ចេញជាច្រើនប្រភេទដែលបានចូលរួមក្នុង DBMS environment ដូចជា data and database administrators, database designers, application programmers និង end-users ។

**4.1. Data and Database Administrators**

Data Administrator (DA) ទទួលបន្ទុកចំពោះការគ្រប់គ្រងធនធានទិន្នន័យ (the management of the data resource) ក្នុងនោះរួមមាន database planning, development and maintenance of standards, policies and procedures, and conceptual/logical database design ។

Database Administrator (DBA) ទទួលបន្ទុកចំពោះ physical realization of the database ក្នុងនោះរួមមាន physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance for the application and users ។

**4.2. Database Designers**

គេបានបែងចែក designer ជា 2 ប្រភេទចំពោះ large database design projects គឺ logical database designers និង physical database designers ។

Logical database designer ទទួលបន្ទុកការងារពាក់ព័ន្ធដល់ identifying data (entities និង attributes), relationship between the data និង constraints លើទិន្នន័យ រក្សាទុកក្នុង database ។ Logical database designer ត្រូវតែដឹង និងយល់ពីទិន្នន័យរបស់ organization អោយបានច្បាស់ ព្រមទាំង business rules ។ **Business rules** describe the

main characteristics of the data as viewed by the organization ។ ឧទាហរណ៍ទៅលើ business rules មួយចំនួនដូចខាងក្រោម:

- + បុគ្គលិកមិនអាចលក់ វិជ្ជុល Property លើសពី 10 ក្នុងពេលតែមួយបានទេ។
- + បុគ្គលិកដើរលក់មិនអាចដើរតួនាទីជាអ្នកទិញផង ជាអ្នកលក់ផងបានទេ។

Physical database designer ពណ៌នាតើវាត្រូវរក្សាទុកក្នុងផ្ទៃ disk ដោយរបៀបណា (decides how it is to physical realized) ។ Physical database designer ទទួលបន្ទុក ការងារពាក់ព័ន្ធដល់:

- បំប្លែង logical data model ទៅកាន់សំនុំ tables និង integrity constraints ។ (Mapping the logical data model into a set of tables and integrity constraints.)
- ជ្រើសរើសរចនាសម្ព័ន្ធរក្សាទុកទិន្នន័យជាក់លាក់ និងវិធីចូលប្រើប្រាស់ទិន្នន័យដើម្បី ទទួលបាន good performance ចំពោះសកម្មភាព database ។ (Selecting specific storage structures and access methods for the data to achieve good performance for the database activities.)
- បង្កើតវិធានការការពារសុវត្ថិភាពចំពោះទិន្នន័យ។ (Designing any security measures required on the data.)

**4. 3. Application Programmers**

បន្ទាប់ពីយើងបានបង្កើត database រួចហើយ យើងត្រូវបង្កើត application programs សំរាប់ធ្វើការទាក់ទងទៅកាន់ database សំរាប់ដំណើរការ required functionality ។ Application programmers ជាអ្នកទទួលបន្ទុកបង្កើត application programs ហើយលើស ពីនេះទៀត រាល់ programs តែងតែផ្ទុកនូវ statements សំណើសុំទៅកាន់ DBMS ដើម្បី ដំណើរការប្រតិបត្តិការមួយចំនួនលើ database ដូចជា retrieving, inserting, updating និង deleting data ។ ភាគច្រើនគេធ្វើការបង្កើត application program ក្នុង third-generation programming language រឺ forth-generation language ។

**4. 4. End-Users**

End-users ចែកជា 2 ប្រភេទគឺ:

- Naive users គឺជាអ្នកដែលមិនស្គាល់ពី DBMS ទាល់តែសោះ។ ពួកគេធ្វើការ ប្រើប្រាស់ database តាមរយៈ application programs ដែលគេសរសេរទុកជាមុនសំរាប់ ដំណើរការប្រតិបត្តិការក្នុង database ។

- Sophisticated users គឺជាអ្នកដែលធ្លាប់ស្គាល់ពី structure of database និង facilities ផ្តល់ដោយ DBMS ។ Sophisticated users អាចប្រើប្រាស់ high-level query language ដូចជា SQL ដើម្បីដំណើរការប្រតិបត្តិការដែលគេត្រូវការ លើសពីនេះទៀត Sophisticated user មួយចំនួនប្រហែលជាអាចសរសេរ application programs សំរាប់ប្រើប្រាស់ផ្ទាល់ខ្លួនរបស់គេបានផងដែរ។

### 5. Advantages and Disadvantages of DBMS

DBMS មានគុណសម្បត្តិជាច្រើន ក៏ប៉ុន្តែក្នុងនោះក៏មានគុណវិបត្តិផងដែរ។

#### 5. 1. Advantages

គុណសម្បត្តិរបស់ DBMS មាន:

- Control of data redundancy

ការប្រើប្រាស់ traditional file-based system មានការខាតបង់ space ជាច្រើនដោយសាររក្សាទុកព័ត៌មានដូចគ្នាក្នុង files លើសពី 1 ដូចជារក្សាទុកទិន្នន័យប្រហាក់ប្រហែលគ្នាចំពោះ Property\_for\_Rent និង Renter ក្នុង Sales ផង និង Contract department ផង។ ផ្ទុយទៅវិញ database approach មានគោលបំណងលុបបំបាត់ចោលភាពស្ងួនទិន្នន័យ (eliminate the redundancy) ដោយច្របាច់ files បញ្ចូលគ្នាធ្វើអោយវាមិនបានរក្សាទុកច្បាប់ចំលងទិន្នន័យដូចគ្នាជាច្រើន (several copies of the same data)។ ទោះបីជាយ៉ាងណាក៏ដោយ database approach មិនបានលុបបំបាត់ចោលភាពស្ងួនទិន្នន័យទាំងស្រុងនោះទេ ប៉ុន្តែវាអាចគ្រប់គ្រងលើចំនួនភាពស្ងួនទិន្នន័យក្នុង database បាន ប៉ុន្តែជូនកាលវាមានភាពចាំបាច់ក្នុងការផ្ទុក data items ដើម្បី improve performance ។

- Data consistency

នៅពេលដែលយើងអាចលុបភាពស្ងួនទិន្នន័យ រឺគ្រប់គ្រងភាពស្ងួនទិន្នន័យ (eliminating or controlling redundancy) ពេលនោះយើងបានកាត់បន្ថយ risk of inconsistencies occurring។ ប្រសិនបើយើងកែប្រែទិន្នន័យ នោះយើងកែប្រែទិន្នន័យតែ 1 កន្លែងទេ ប៉ុន្តែផ្ទុយទៅវិញប្រសិនបើ data items រក្សាទុកលើសពី 1 កន្លែង ហើយប្រព័ន្ធមិនបានដឹងពីបញ្ហានេះ នោះប្រព័ន្ធត្រូវតែប្រាកដថារាល់ច្បាប់ចំលង item (all copies of the item) រក្សា consistent ។ គួរអោយសោកស្តាយព្រោះ DBMS ភាគច្រើនមិនបានរក្សា consistency ដោយស្វ័យប្រវត្តិទេ។

- More information from the same amount of data  
នៅពេលយើងច្របាច់ operational data បញ្ចូលគ្នាធ្វើអោយ organization មានលទ្ធភាពក្នុងការបង្កើតព័ត៌មានបន្ថែម (derived additional information) ចេញពីទិន្នន័យដូចគ្នា។

- Sharing data  
ជាទូទៅ people រឺ department ដែលប្រើ files ច្រើនតែជាម្ចាស់កម្មសិទ្ធិ files នោះផ្ទុយទៅវិញ database ដែលជាកម្មសិទ្ធិនៃ organization ទាំងមូលត្រូវតែអាចប្រើប្រាស់ដោយ authorized users។ ប្រសិនបើមាន users ច្រើន ពេលនោះការប្រើប្រាស់ទិន្នន័យក៏កាន់តែច្រើនដែរ ហើយយើងអាចបង្កើត application ថ្មីប្រើប្រាស់ទិន្នន័យក្នុង database មានស្រាប់ និងបន្ថែមទិន្នន័យដែលមិនទាន់បានរក្សាទុក ជាជាងធ្វើការកំណត់តម្រូវការទិន្នន័យទាំងអស់សារជាថ្មី។

- Improved data integrity  
Database integrity សំដៅទៅលើ validity និង consistency នៃទិន្នន័យរក្សាទុក។ Integrity ជាទូទៅបង្ហាញតាមរយៈពាក្យ constraints (**Constraints** ជា consistent rules កំណត់លើ database មិនអោយបំពាន)។ Constraint អាចកំណត់លើ data items ក្នុង single record រឺ relationship រវាង records។ ឧទាហរណ៍: integrity constraint កំណត់ថា employee's salary មិនអាចលើសពី \$4,000 ហើយ DBMS ច្រើនតែធ្វើការ enforce integrity constraints ។

- Improved security  
Database security សំដៅលើការរារាំងចូលប្រើប្រាស់ទិន្នន័យក្នុង database ពី unauthorized users។ ប្រសិនបើមិនបានចាត់វិធានការការពារអោយបានសមរម្យនោះទេ ការច្របាច់ទិន្នន័យចូលគ្នាមានលក្ខណៈសុគតស្មាញជាង file-based system ហើយជាទូទៅ DBMS ធ្វើការកំណត់ database security ក្នុងទម្រង់ជា user names និង passwords សំរាប់កំណត់ authorized user។ ជាងនេះទៅទៀត យើងអាចកំណត់សិទ្ធិបន្ថែមទៀតទៅលើ authorized user ក្នុងការប្រើប្រាស់ទិន្នន័យដូចជា retrieval, insert, update, delete។ ឧទាហរណ៍: DBA អាចចូលប្រើប្រាស់រាល់ទិន្នន័យទាំងអស់ក្នុង database ប៉ុន្តែ branch manager អាចចូលប្រើប្រាស់ទិន្នន័យដែលពាក់ព័ន្ធនឹងការងាររបស់គាត់តែប៉ុណ្ណោះ។

- Economy of scale  
ការបញ្ចូលគ្នារាល់ operational data ទាំងអស់នៃ organization ទៅកាន់ database ហើយបង្កើត applications ធ្វើការជាមួយប្រភពទិន្នន័យនោះ អាចជួយកាត់បន្ថយតំលៃក្នុងការ

ចំណាយ។ លុយធ្លាប់ចំណាយទៅលើ department នីមួយៗចំពោះការអភិវឌ្ឍន៍ និងថែរក្សា file-based system អាចបញ្ចូលគ្នា ប្រហែលជាទទួលបាន lower total cost នាំអោយមាន economy of scale។ Combined budget អាចប្រើប្រាស់សំរាប់ទិញ system configuration ដែលសមរម្យទៅនឹងតំរូវការ organization ហើយប្រហែលជាផ្ទុកនូវ one large, powerful computer រឺ a network of smaller computers ។

- Balance of conflicting requirements

តំរូវការរបស់ user រឺ department អាចមាន conflict ជាមួយតំរូវការរបស់ user ផ្សេងទៀត។ នៅពេលដែល database ស្ថិតក្រោមការគ្រប់គ្រងរបស់ DBA, DBA អាចសំរេចចិត្តធ្វើការកសាង និងប្រើប្រាស់ database ដែលផ្តល់នូវការប្រើប្រាស់ប្រភពទិន្នន័យល្អបំផុតចំពោះ organization ទាំងមូល។

- Improved data accessibility and responsiveness

DBMS ភាគច្រើនផ្តល់នូវ query language រឺ report ដែលអនុញ្ញាតិអោយ users ធ្វើការសាកសួរ ad hoc question (សំនួរដែលកើតឡើងភ្លាមៗ) និងទទួលបានព័ត៌មានទាំងនោះស្ទើរតែភ្លាមៗលើ terminal របស់ពួកគេដោយមិនចាំបាច់ទាមទារអោយ programmers សរសេរ software ទាញព័ត៌មានទាំងនោះពី database ទេ។ ឧទាហរណ៍: branch manager អាចបង្ហាញរាល់ flats ទាំងអស់ដែលមាន monthly rent លើសពី \$400 ដោយឃ្លាបញ្ជា SQL លើ terminal ដូចខាងក្រោម:

```
SELECT *
FROM Property_for_Rent
WHERE type = 'Flat' AND rent > 400
```

- Increased productivity

DBMS បានផ្តល់នូវ standard functions មួយចំនួនដែល programmers ជាធម្មតាតែងតែសរសេរនៅក្នុង file-based application។ ការផ្តល់ functions ទាំងនេះនៃ DBMS អនុញ្ញាតិអោយ programmers ជញ្ជីងគិតបន្ថែមទៅលើ specific functions ណាដែល users ត្រូវការដោយគ្មានការអង្កេត low-level implementation details។ DBMSs ជាច្រើនបានផ្តល់ផងដែរនូវ fourth-generation environment ដោយផ្ទុកនូវ tools សំរាប់ជួយសំរួលការអភិវឌ្ឍន៍ database applications ហើយលទ្ធផលនេះធ្វើអោយកើន programmer productivity និងកាត់បន្ថយពេលវេលាកសាង (ជាមួយតំលៃចំណាយផងដែរ) ។

- Improved maintenance through data independence  
 នៅក្នុង file-based system ការពណ៌នាទិន្នន័យ (description of data) និង logic សំរាប់ប្រើប្រាស់ (accessing) ទិន្នន័យត្រូវបានច្របាច់បញ្ចូលទៅជាមួយ application program មួយៗ ធ្វើអោយ programs ពឹងពាក់លើទិន្នន័យ។ ប្រសិនបើមានការកែប្រែទៅលើរចនាសម្ព័ន្ធ ទិន្នន័យ (structure of the data) ឧទាហរណ៍: កែប្រែ address ដែលកំនត់ 40 តួអក្សរទៅជា 41 តួអក្សរ វិការកែប្រែទៅលើវិធីដែលទិន្នន័យរក្សាទុកលើ disk (a change to the way the data is stored on disk) តម្រូវអោយមានការកែប្រែសមរម្យទៅលើ programs ដែលប៉ះពាល់។ ផ្ទុយទៅវិញចំពោះ DBMS ដែលផ្ទុក data descriptions ដាច់ដោយឡែកពី applications ធ្វើអោយការកែប្រែ applications មិនប៉ះពាល់ដល់ data descriptions ទេ នេះគឺជាលក្ខណៈ **data independence** ។

- Increased concurrency  
 នៅក្នុង file-based system ប្រសិនបើអនុញ្ញាតិអោយ users 2 រឺច្រើនប្រើប្រាស់ file តែមួយក្នុងពេលដំណាលគ្នានោះ វាអាចមានការប៉ះទង្គិចគ្នាទៅវិញទៅមកបណ្តាលអោយមានការ បាត់បង់ព័ត៌មាន (loss of information) រឺជាងនេះទៅទៀត loss of integrity ។ DBMS ជាច្រើនអាចគ្រប់គ្រងការចូលប្រើប្រាស់ database ក្នុងពេលតែមួយបាន ហើយអាចការពារបញ្ហា នេះមិនអោយកើតឡើង។

- Improved backup and recovery services  
 File-based systems ជាច្រើនបានដាក់ការទទួលខុសត្រូវទៅលើ users ដើម្បីចាត់ វិធានការការពារទិន្នន័យពីបញ្ហាកើតមានឡើងនៅលើ computer system រឺ application program ដែលទាមទារអោយធ្វើការ backup ទិន្នន័យរៀងរាល់យប់។ នៅពេលមានព្រឹត្តិការណ៍ បញ្ហាមួយកើតឡើងនៅថ្ងៃបន្ទាប់ នោះយើងត្រូវតែ restore backup ហើយរាល់ការងារដែលបាន បញ្ចូលចាប់តាំងពីពេល backup ត្រូវបាត់បង់ដោយទាមទារអោយបញ្ចូលសារជាថ្មី។ ផ្ទុយទៅវិញ modern DBMSs ផ្តល់ facilities ដើម្បីកាត់បន្ថយចំនួនប្រតិបត្តិការដែលបានបាត់បង់។

**5. 2. Disadvantages**

គុណវិបត្តិរបស់ DBMS មាន:

- Complexity  
 ការផ្តល់ទុកជាមុននូវ functionality នៃ good DBMS ធ្វើអោយ DBMS មាន លក្ខណៈស្មុគស្មាញសំបាប់ (extremely complex piece of software)។ Database designers and developers, data and database administrators, and end-users ត្រូវ

តែយល់ functionality អោយបានច្បាស់ដើម្បីទាញផលប្រយោជន៍ពីវា។ កំហុសមិនបានយល់ ប្រព័ន្ធអាចបណ្តាលទៅដល់ការសំរេចចិត្តកសាងមិនល្អ (bad design decision) ដែលអាចបង្ក អោយមានផលវិបាកដល់ organization ។

- Size

ភាពស្មុគស្មាញ និងភាពទូលំទូលាយ (complexity and breadth) នៃ functionality ធ្វើអោយ DBMS ជា software មានទំហំធំធេង ដែលត្រូវការ disk space ជាច្រើន megabytes និងទាមទារ memory សមរម្យដើម្បីដំណើរការអោយកាន់តែប្រសើរ។

- Cost of DBMSs

Cost of DBMSs ប្រែប្រួលគួរអោយកត់សំគាល់ អាស្រ័យទៅនឹង environment និង functionality ដែលវាបានផ្តល់អោយ។ ឧទាហរណ៍: a single-user DBMS សំរាប់ a personal computer ប្រហែលជាតំលៃ \$100 ទោះបីជាយ៉ាងណាក៏ដោយ a large mainframe multi-user DBMS សំរាប់ផ្តល់សេវាដល់អ្នកប្រើប្រាស់ជាច្រើនមានតំលៃថ្លៃណាស់ ប្រហែល \$100,000 ទៅ \$500,000 លើសពីនេះទៀត វាត្រូវការថ្លៃថែទាំប្រចាំឆ្នាំ (annual maintenance cost) ទៀតផង។

- Additional hardware costs

តម្រូវការ disk space សំរាប់ DBMS និង database អាចធ្វើអោយមានការទិញបន្ថែម storage space ថែមទៀត។ លើសពីនេះទៀត ដើម្បីសំរេចបាន performance ត្រូវការវាចាំបាច់ ទិញ a large machine សំរាប់ដំណើរការ DBMS នាំអោយមានការចំណាយបន្ថែមទៀត។

- Cost of conversion

ក្នុងករណីមួយចំនួន ចំណាយទៅលើ DBMS និង extra hardware ប្រហែលជាមិនសូវ មានសារៈសំខាន់ប៉ុន្មានទេបើប្រៀបធៀបទៅនឹងចំណាយលើការបំប្លែង existing application អោយដំណើរការលើ new DBMS និង hardware ។ ការចំណាយនេះរួមបញ្ចូលទាំងចំណាយ ក្នុងការបណ្តុះបណ្តាលបុគ្គលិក (cost of training staff) អោយចេះប្រើប្រាស់ប្រព័ន្ធច្នី និងប្រហែលជាចំណាយលើបុគ្គលិកជំនាញ (specialist staff) ដើម្បីជួយទៅលើការបំប្លែង និងដំណើរការប្រព័ន្ធនោះ។ ការចំណាយនេះជាហេតុផលមួយសំខាន់បំផុតដែល organizations មួយចំនួននៅតែប្រើប្រាស់ប្រព័ន្ធចាស់របស់គេដែល ដោយមិនប្តូរទៅ modern database technology ។ ពាក្យ **legacy system** ជួនកាលប្រើប្រាស់សំរាប់សំគាល់ older and inferior system ។



- Performance

File-based system ត្រូវគេសរសេរសំរាប់ specific applications ណាមួយតែម្តង ដូចជា invoicing ដូចនេះ performance ជាទូទៅតែងតែល្អ។ ទោះបីជាយ៉ាងណាក៏ដោយ DBMS ដែលសរសេរក្នុងលក្ខណៈទូលំទូលាយដើម្បីបំពេញបំណង applications ជាច្រើន ដូចនេះ applications មួយចំនួនប្រហែលដំណើរការមិនបានលឿនទេ។

- Higher impact of a failure

ការគ្រប់គ្រងធនធានពីកន្លែងតែមួយ (centralization of resources) បង្កើនលទ្ធភាព ដែលអាចធ្វើអោយប្រព័ន្ធជាមិនមានបញ្ហា។ ដោយសារ users និង applications ទាំងមូល ពឹងពាក់ទាំងស្រុងទៅលើ DBMS ដូចនេះប្រសិនបើ DBMS មានបញ្ហាទៅលើផ្នែកណាមួយ វាអាចបណ្តាលអោយមានការប៉ះពាល់ដល់ប្រព័ន្ធទាំងមូល។

**Review Questions**

1. ចូរពន្យល់អោយបានក្បោះក្បាយនូវពាក្យខាងក្រោម
  - (a) data
  - (b) database
  - (c) database management system
  - (d) data independence
  - (e) security
  - (f) integrity
  - (g) views
2. ចូរពិពណ៌នាដំណើរការលើ file-based system ព្រមទាំងគុណវិបត្តិរបស់វាផងដែរ?
3. ចូរពិពណ៌នាសមាសភាពទាំង 5 នៃ DBMS environment?
4. ចូរពិពណ៌នាតួនាទីបុគ្គលខាងក្រោមក្នុង database environment
  - (a) data administrator
  - (b) database administrator
  - (c) logical database designer
  - (d) physical database designer
  - (e) application programmer
  - (f) end-users
5. ចូរពិពណ៌នាអំពីគុណសម្បត្តិ និងគុណវិបត្តិនៃ database management system?

