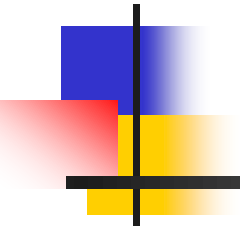
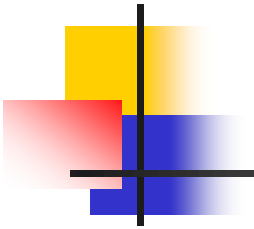
Introduction to Databases



Instructor: C. Pu (Ph.D., Assistant Professor)

puc@marshall.edu

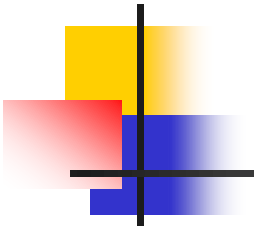


Database Application

- Purchases from the supermarket
- Purchases using your credit card
- Booking a holiday at the travel agents
- Using the local library
- Using the Internet
- Studying at university

File-Based Systems

- Why studying file-based system?
  - Understanding the problems inherent in file-based systems may prevent us from repeating these problems in database systems.
  - If you need to convert a file-based system to a database system, understanding how the file system works will be extremely useful.
- File-based system:
  - A collection of application programs that perform services for the end users (e.g. production of reports).
  - Each program defines and manages ***its own data***.
- File-based system were an early attempt to computerize the manual filing system.
  - Physical files saving information
  - E.g., Bank statement

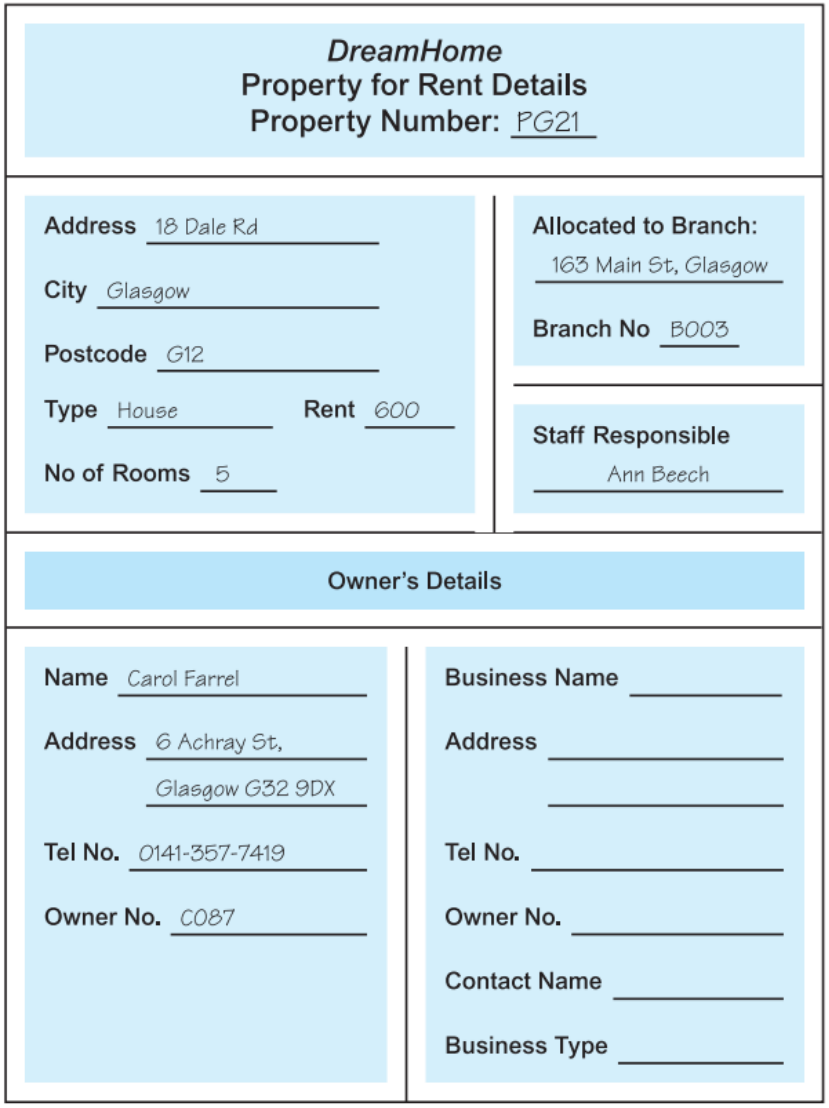


File-Based Systems

- The manual filing system works well when
  - The number of items to be stored is small
  - There are large numbers of items and we have only to store and retrieve them
- The manual filing system breaks down when we have to corss-reference or process the information in the files
- Example:
  - a typical real estate agent's office might have a separate file for each property for sale or rent, each potential buyer and renter, and each member of staff.

File-Based Systems

- Case Study:
  - The Sales Department is responsible for the selling and renting of properties.
    - A client who wishes to offer his or her property as a rental approaches the Sales Department, fill out "Property for Rent" form.



DreamHome
Property for Rent Details
Property Number: PG21

Address 18 Dale Rd
City Glasgow
Postcode G12
Type House Rent 600
No of Rooms 5

Allocated to Branch:
163 Main St, Glasgow
Branch No B003

Staff Responsible
Ann Beech

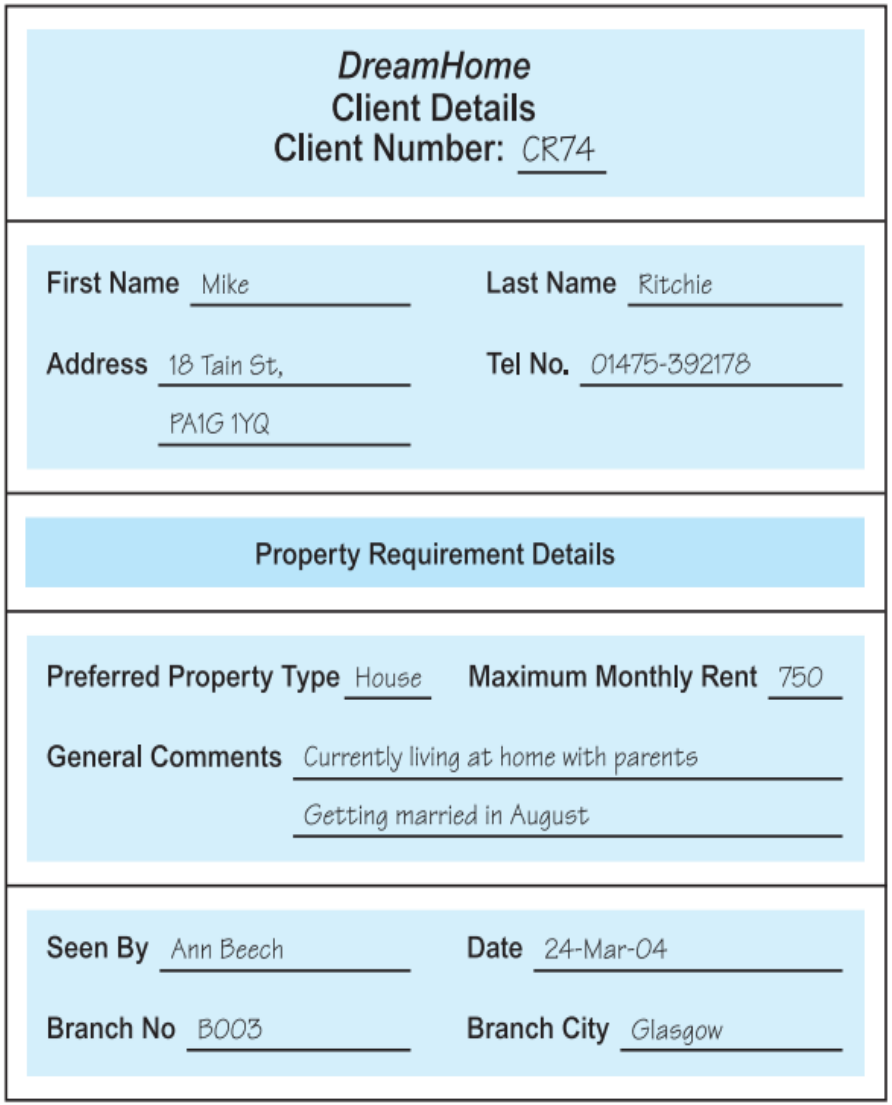
Owner's Details

Name Carol Farrel
Address 6 Achray St,
Glasgow G32 9DX
Tel No. 0141-357-7419
Owner No. C087

Business Name
Address
Tel No.
Owner No.
Contact Name
Business Type

File-Based Systems

- Case Study:
  - The Sales Department is responsible for the selling and renting of properties.
    - The Sales Department handles inquiries from clients, and inquiry form is filled out.



DreamHome
Client Details
Client Number: CR74

First Name Mike
Last Name Ritchie
Address 18 Tain St, PA1G 1YQ
Tel No. 01475-392178

Property Requirement Details

Preferred Property Type House
Maximum Monthly Rent 750
General Comments Currently living at home with parents Getting married in August

Seen By Ann Beech
Date 24-Mar-04
Branch No B003
Branch City Glasgow

File-Based Systems

- The Sales Department creates an information systems to handle the renting of property.
- The system consists of three files containing property, owner, and client details.

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40
PG36	2 Manor Rd						
PG21	18 Dale Rd						
PG16	5 Novar Dr						

PrivateOwner

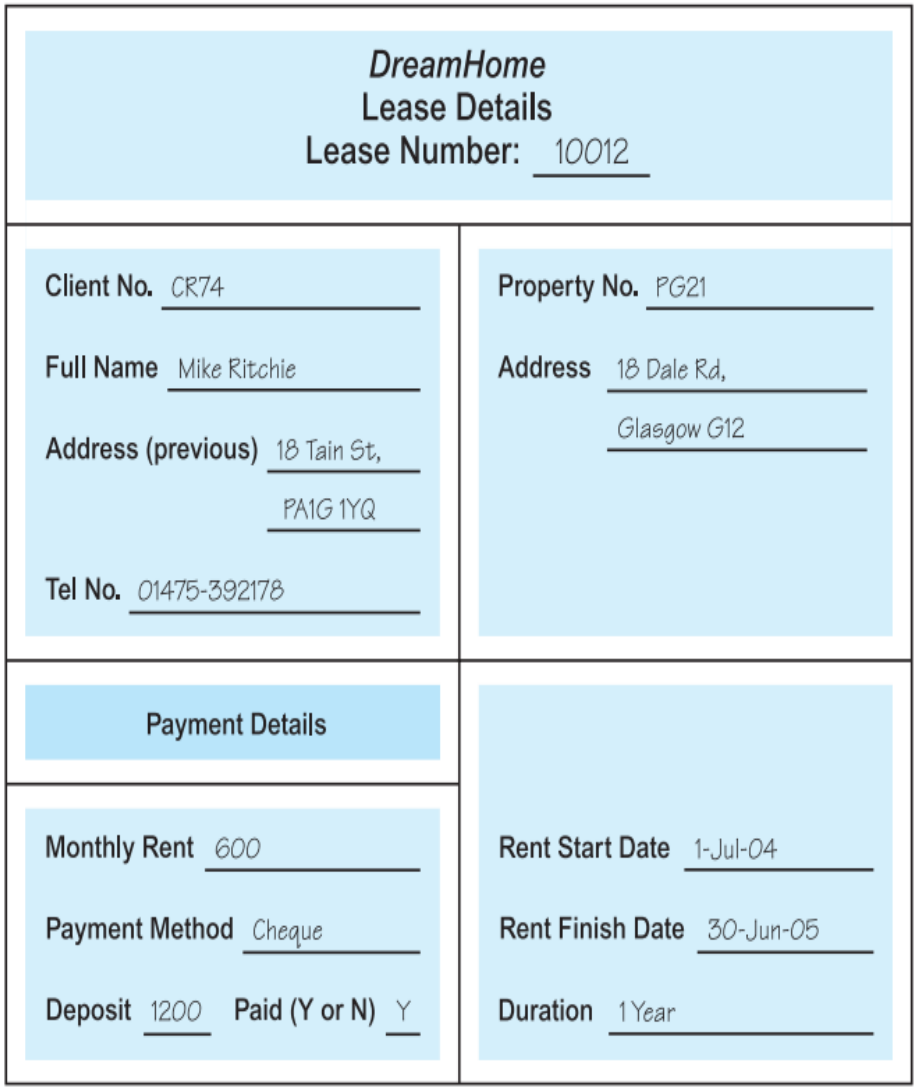
ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol			
CO40	Tina			
CO93	Tony			

Client

clientNo	fName	lName	address	telNo	prefType	maxRent
CR76	John	Kay	56 High St, London SW1 4EH	0207-774-5632	Flat	425
CR56	Aline	Stewart	64 Fern Dr, Glasgow G42 0BL	0141-848-1825	Flat	350
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178	House	750
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720	Flat	600

File-Based Systems

- Case Study:
  - The Contracts Department is responsible for handling the lease agreements associated with properties for rent.
    - Whenever a client agrees to rent a property, a form with the client and property details is filled.



DreamHome
Lease Details
Lease Number: 10012

Client No. CR74
Full Name Mike Ritchie
Address (previous) 18 Tain St, PA1G 1YQ
Tel No. 01475-392178

Property No. PG21
Address 18 Dale Rd, Glasgow G12

Payment Details

Monthly Rent 600
Payment Method Cheque
Deposit 1200 Paid (Y or N) Y

Rent Start Date 1-Jul-04
Rent Finish Date 30-Jun-05
Duration 1 Year

File-Based Systems

- The Contract Department creates an information systems to handle lease agreements.
- The system consists of three files storing lease, properpty, and client details.

Lease

leaseNo	propertyNo	clientNo	rent	payment Method	deposit	paid	rentStart	rentFinish	duration
10024	PA14	CR62	650	Visa	1300	Y	1-Jun-05	31-May-05	12
10075	PL94	CR76	400	Cash	800	N	1-Aug-05	31-Jan-05	6
10012	PG21	CR74	600	Cheque	1200	Y	1-Jul-05	30-Jun-05	12

PropertyForRent

propertyNo	street	city	postcode	rent
PA14	16 Holhead	Aberdeen	AB7 [illegible]	[illegible]
PL94	6 Argyll St	London	NW2	
PG21	18 Dale Rd	Glasgow	G12	

Client

clientNo	fName	lName	address	telNo
CR76	John	Kay	56 High St, London SW1 4EH	0171-774-5632
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720

File-Based Systems

Sales Department

DreamHome
Client Details
Client Number: CR74

First Name Mike Last Name Ritchie

Address 18 Tain St, PA1G 1YQ Tel No. 01475-392178

Property Requirement Details

Preferred Property Type House Maximum Monthly Rent 750

General Comments Currently living at home with parents Getting married in August

Seen By Ann Beech Date 24-Mar-04

Branch No B003 Branch City Glasgow

Contracts Department

DreamHome
Lease Details
Lease Number: 10012

Client No. CR74

Full Name Mike Ritchie

Address (previous) 18 Tain St, PA1G 1YQ

Tel No. 01475-392178

Property No. PG21

Address 18 Dale Rd, Glasgow G12

Payment Details

Monthly Rent 600

Payment Method Cheque

Deposit 1200 Paid (Y or N) Y

Rent Start Date 1-Jul-04

Rent Finish Date 30-Jun-05

Duration 1 Year

File-Based Systems

File-based processing in Sales and Contracts Departments

Sales Files

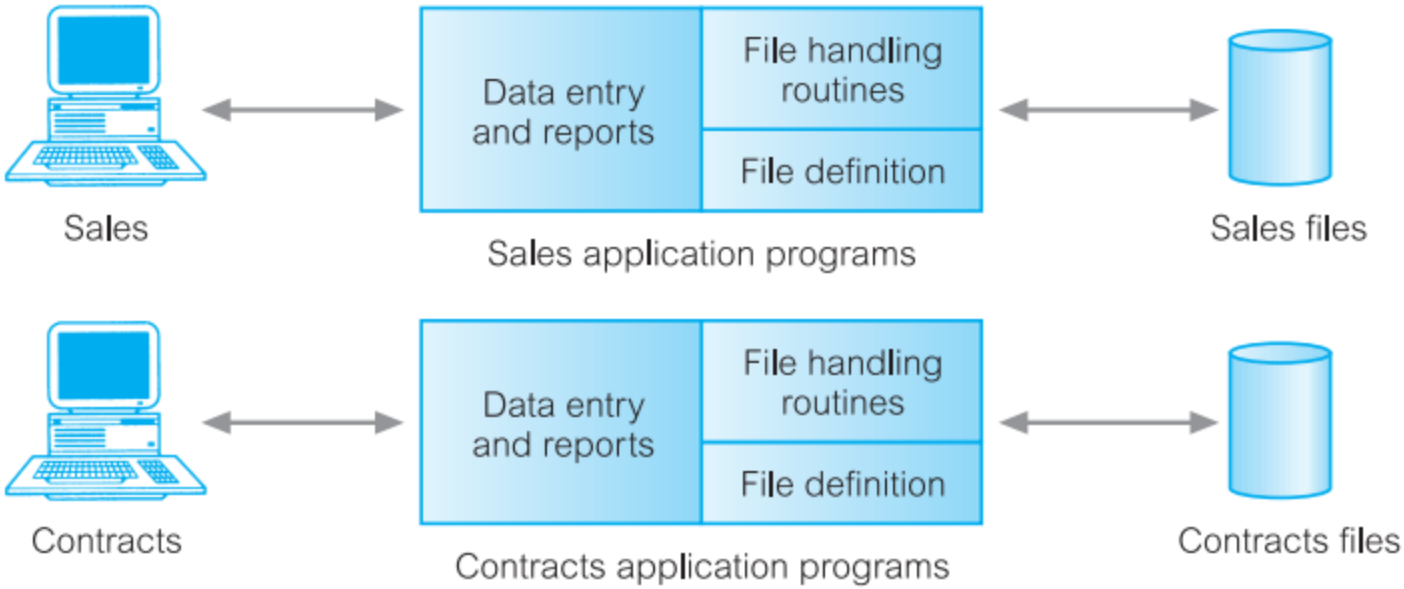
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

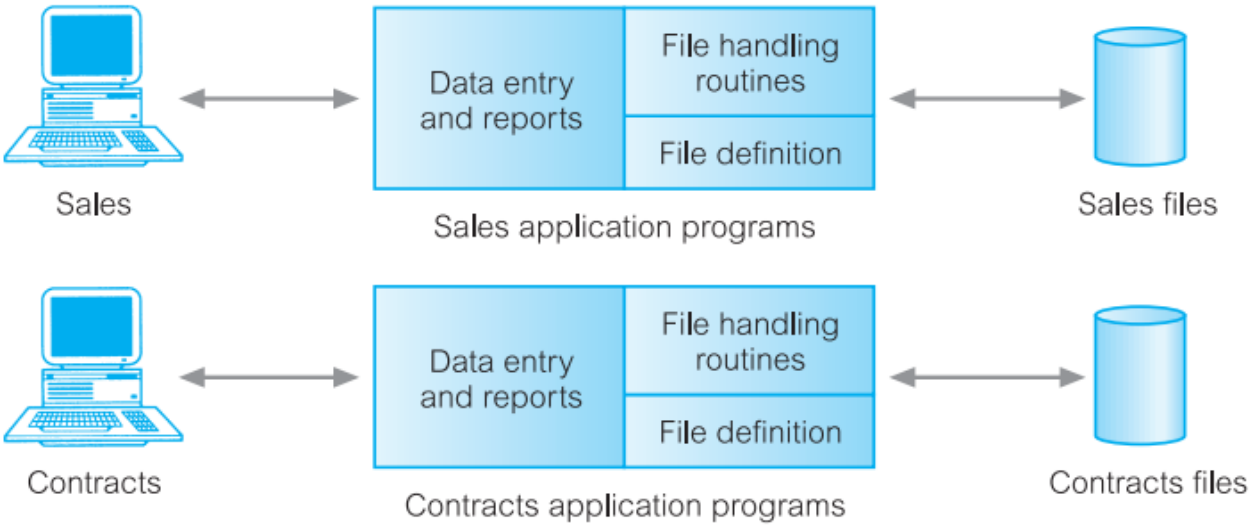
File-Based Systems

File-based processing in Sales and Contracts Departments



Limitations of File-Based Approach

- **Separation and isolation of data**
  - Each program maintains its own set of data.
  - Users of one program may be unaware of potentially useful data held by other programs.



Limitations of File-Based Approach

- **Duplication of data**
  - Same data is held by different programs.
  - Wasted space and potentially different values and/or different formats for the same item.
  - Data integrity problem.

Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

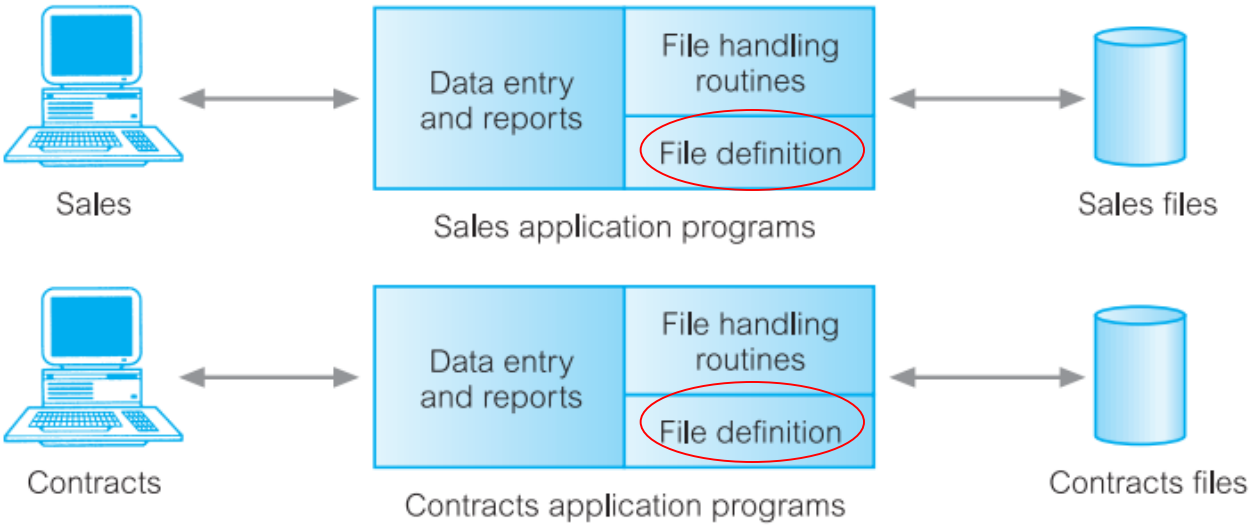
Contracts

Contracts application programs

Contracts files

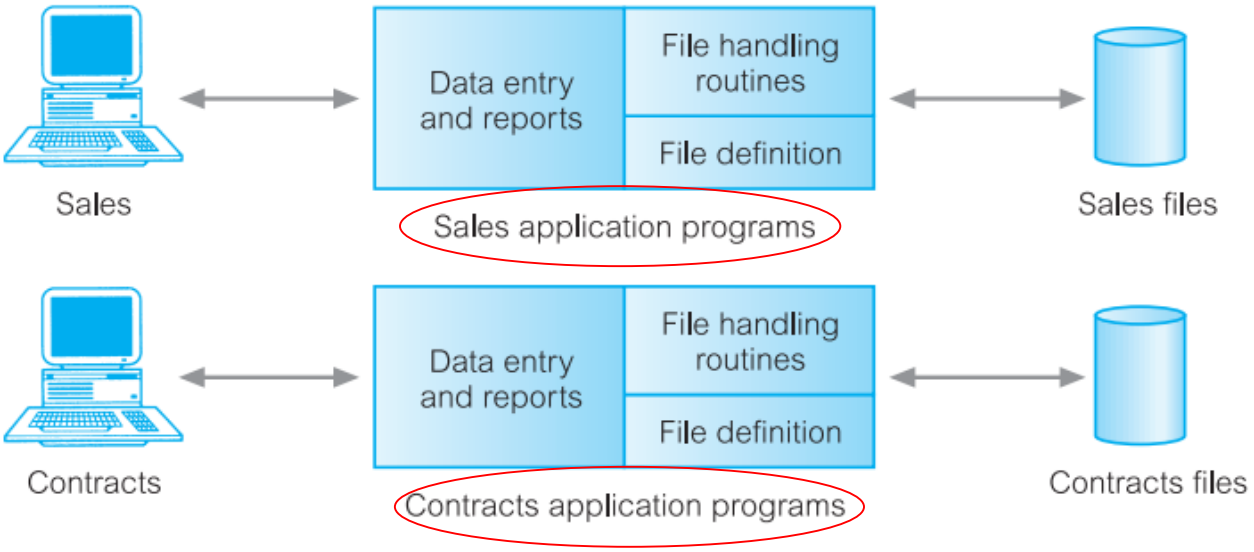
Limitations of File-Based Approach

- **Data dependence**
  - File structure is defined in the program code.
  - Changes to an existing structure are difficult to make.



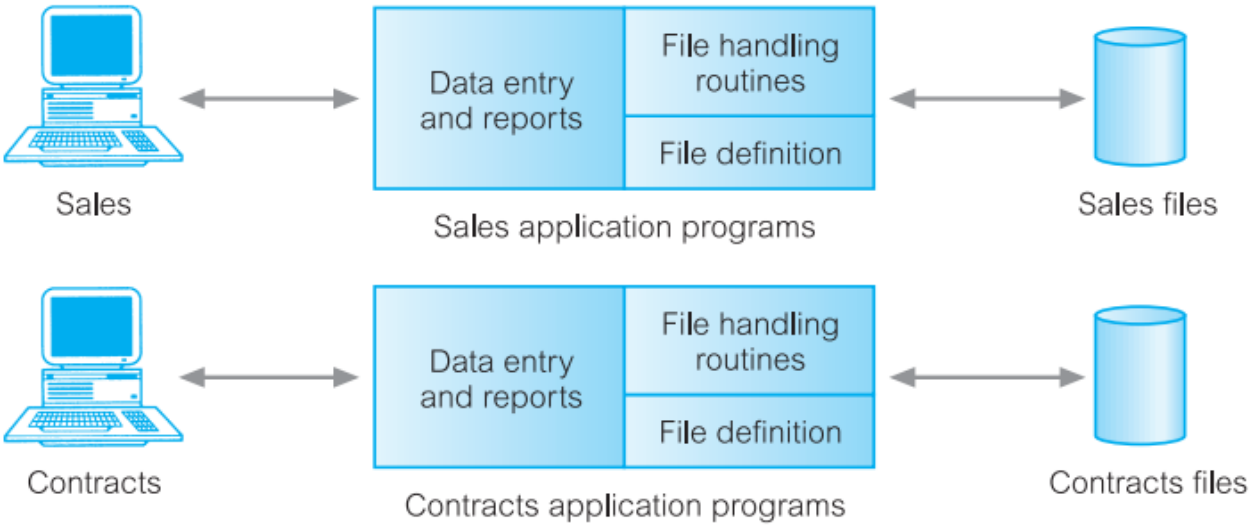
Limitations of File-Based Approach

- **Incompatible file formats**
  - Programs are written in different languages, and so cannot easily access each other's files.



Limitations of File-Based Approach

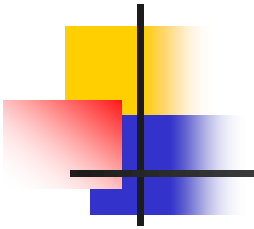
- **Fixed Queries/Proliferation of application programs**
  - Programs are written to satisfy particular functions.
  - Any new requirement needs a new program.



Database Approach

- Limitations of the file-based approach can be attributed to two factors:
  - Definition of data was embedded in application programs, rather than being stored separately and independently.
  - No control over access and manipulation of data beyond that imposed by application programs.

- New approach:
  - Database and Database Management System (DBMS).



Database

- Database:
  - (Definition) A shared collection of logically related data and a description of this data, designed to meet the information needs of an organization.
  - The database is a single, possibly large repository of data that can be used simultaneously by many users.
  - All data times are integrated with a minimum amount of duplication.
  - The database is not longer owned by one department but is a shared corporate resource.
  - The database holds not only the organization's operational data but also a description of this data.
  - *Self-describing collection of integrated records*

Database

- System catalog:
  - System catalog (metadata) provides description of data to enable program–data independence.

- Logically related data comprises ***entities***, ***attributes***, and ***relationships*** of an organization's information.
  - Entity: a distinct object in the organization that is to be represented in the database
  - Attribute: a property that describes some aspect of the object
  - Relationship: an association between entities

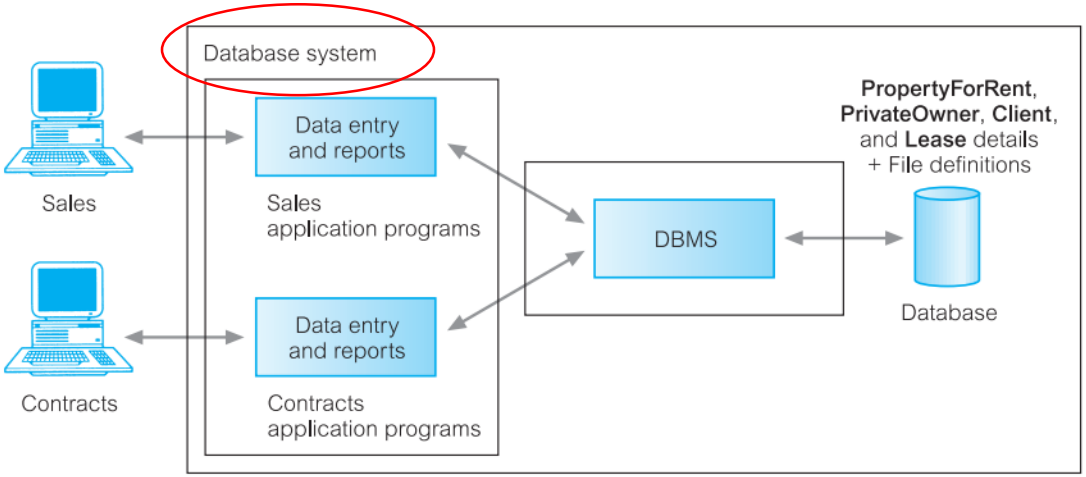
Database Management System (DBMS)

- DBMS:
  - A software system that enables users to define, create, maintain, and control access to the database.
  - A DBMS is the *software* that interacts with the user's application programs and the database.
  - It allows users to define the database.
  - IT allows users to insert, update, delete, and retrieve data from the database.
  - It provides controlled access to the database.

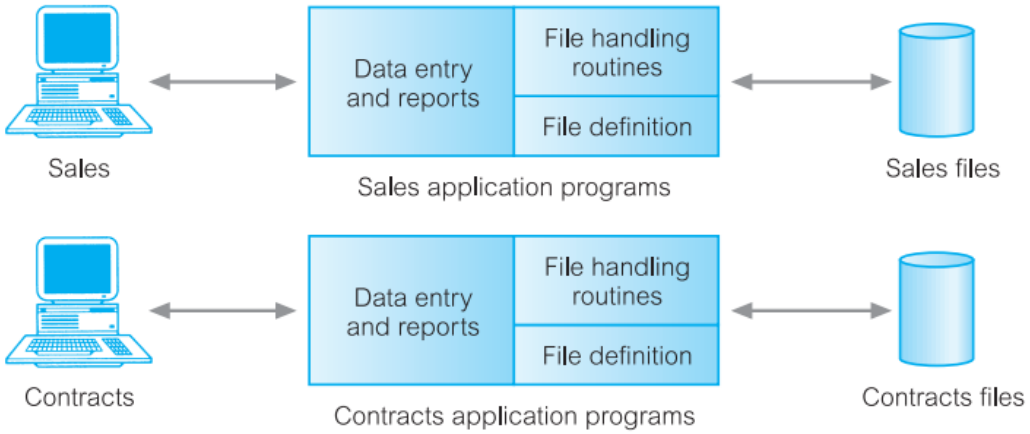
- Database application program:
  - A computer program that interacts with database by issuing an appropriate request (SQL statement) to the DBMS.

Database Management System (DBMS)

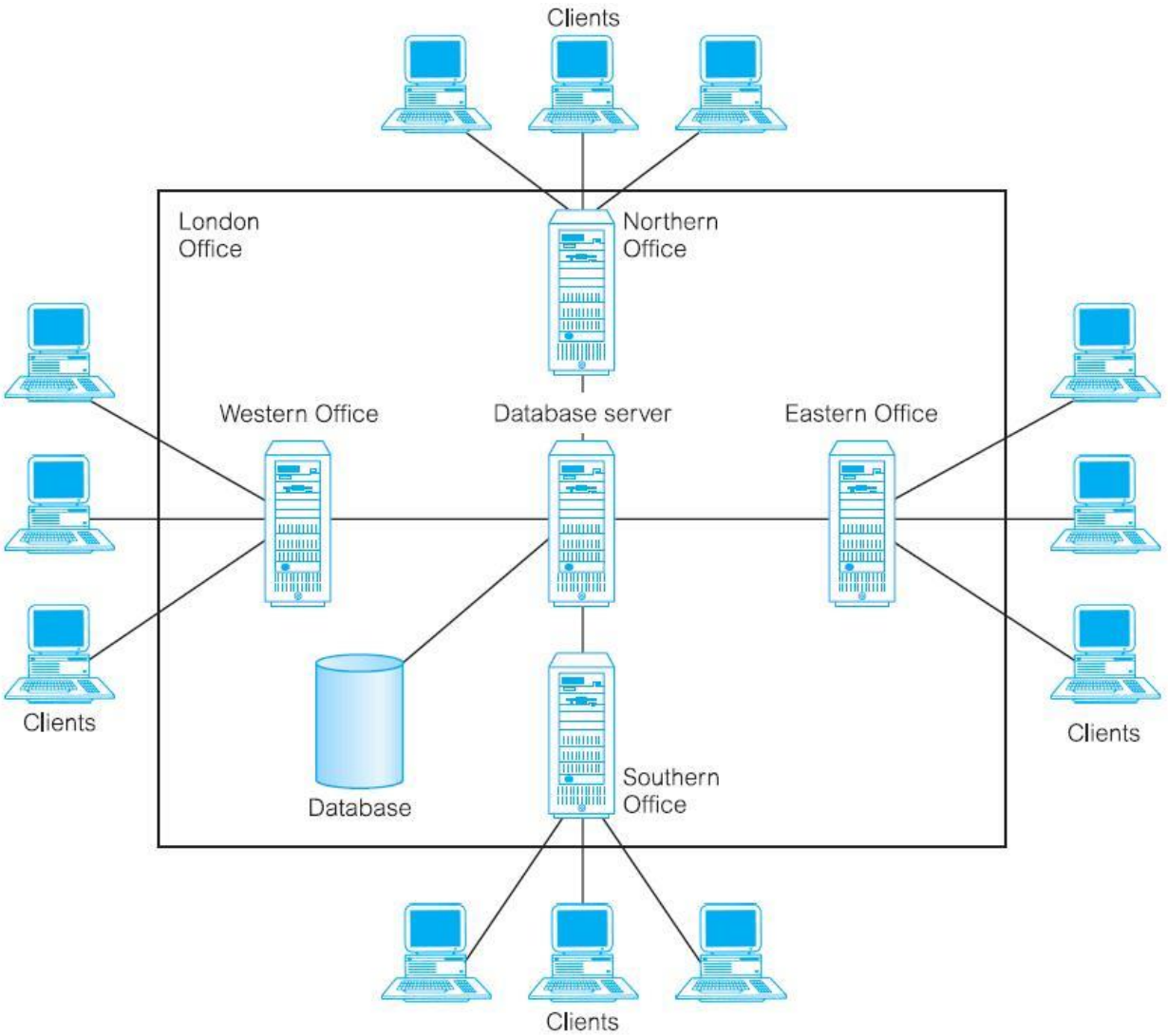
Database Management System (DBMS)



File-based System

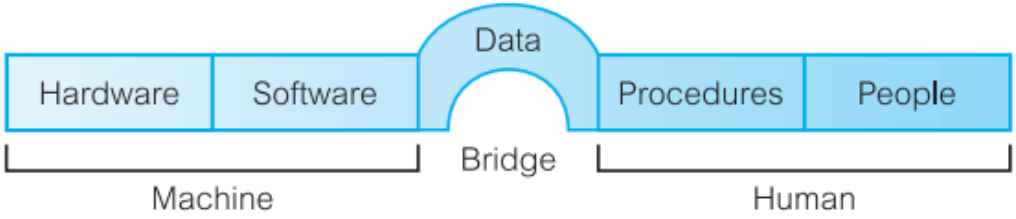


Database Management System (DBMS)



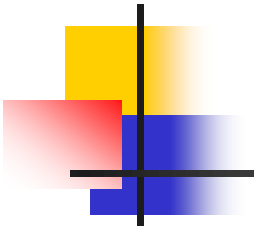
Components of DBMS Environment

- Hardware
  - The hardware can range from a PC to a network of computers.
- Software
  - DBMS, operating system, network software (if necessary) and also the application programs.
- Data
  - Used by the organization and a description of this data called the schema.
- Procedures
  - Instructions and rules that should be applied to the design and use of the database and DBMS.
- People



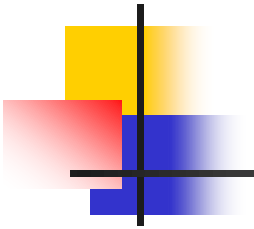
Roles in the Database Environment

- Data Administrator (DA)
  - DA is responsible for the management of the data resource, including database planning, development and maintenance of standards, policies and procedures, and conceptual/logical database design.
- Database Administrator (DBA)
  - DBA is responsible for the physical realization of the database, including physical database design and implementation, security and integrity control.
- **Database Designers**
  - Designers are concerned with identifying the data, the relationship between the data, and the constraints on the data that is to be stored in the database.
- **Application Programmers**
  - Once the database has been implemented, the application programs that provide the required functionality for the end-user must be implemented.
- End Users



Advantages of DBMSs

- Control of data redundancy
- Data consistency
- Sharing of data
- Improved data integrity
- Improved security
- Increased concurrency
- Improved backup and recovery services
- And more…



Disadvantages of DBMSs

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- And some…

An Example

- A UNIVERSITY database for maintaining information concerning students, courses, and grades in a university environment.

- The database is organized as five files, each of which stores data records of the same type:
  - The STUDENT file stores data on each student
  - The COURSE file stores data on each course
  - The SECTION file stores data on each section of a course
  - The GRADE_REPORT file stores the grades that students receive in the various sections they have completed
  - The PREREQUISITE file stores the prerequisites of each course

An Example

- The STUDENT file stores data on each student

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

- Each STUDENT record includes data to represent the student's Name, Student_number, Class (such as freshman or '1', sophomore or '2', and so forth), and Major (such as mathematics or 'MATH' and computer science or 'CS');

An Example

- The COURSE file stores data on each course

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

- Each COURSE record includes data to represent the Course_name, Course_number, Credit_hours, and Department (the department that offers the course);

An Example

- The SECTION file stores data on each section of a course

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

An Example

- The GRADE_REPORT file stores the grades that students receive in the various sections they have completed

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

An Example

- The PREREQUISITE file stores the prerequisites of each course

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

An Example

- To construct the UNIVERSITY database, we store data to represent each student, course, section, grade report, and prerequisite as a record in the appropriate file.
- Notice that records in the various files may be related.
  - The record for Smith in the STUDENT file is related to two records in the GRADE_REPORT file that specify Smith's grades in two sections.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

An Example

- To construct the UNIVERSITY database, we store data to represent each student, course, section, grade report, and prerequisite as a record in the appropriate file.
- Notice that records in the various files may be related.
  - Each record in the PREREQUISITE file relates two course records: one representing the course and the other representing the prerequisite.

PREREQUISITE

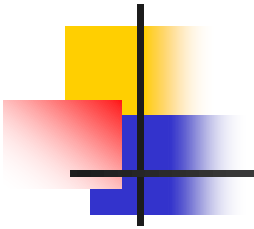
Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

An Example

- Database manipulation involves querying and updating.
- Examples of queries are as follows:
  - Retrieve the transcript—a list of all courses and grades—of 'Smith'
  - List the names of students who took the section of the 'Database' course offered in fall 2008 and their grades in that section
  - List the prerequisites of the 'Database' course
- Examples of updates include the following:
  - Change the class of 'Smith' to sophomore
  - Create a new section for the 'Database' course for this semester
  - Enter a grade of 'A' for 'Smith' in the 'Database' section of last semester



An Example

- Design of a new application for an existing database or design of a brand new database starts off with a phase called *requirements specification and analysis*.
  - These requirements are documented in detail and transformed into a conceptual design that can be represented and manipulated using some computerized tools so that it can be easily maintained, modified, and transformed into a database implementation.
  - The design is then translated to a logical design that can be expressed in a data model implemented in a commercial DBMS.
  - The final stage is physical design, during which further specifications are provided for storing and accessing the database.
  - The database design is implemented, populated with actual data, and continuously maintained to reflect the state of the miniworld.

Characteristics of the Database Approach

- In the database approach, a single repository maintains data that is defined once and then accessed by various users.

- The main characteristics of the database approach versus the file-processing approach are the following:
  - Self-describing nature of a database system
  - Insulation between programs and data, and data abstraction
  - Support of multiple views of the data
  - Sharing of data and multiuser transaction processing

Self-Describing Nature of a Database System

- A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints.
- This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data.
- The information stored in the catalog is called meta-data, and it describes the structure of the primary database.

Self-Describing Nature of a Database System

- The catalog is used by the DBMS software and also by database users who need information about the database structure.

- A general-purpose DBMS software package is not written for a specific database application.
  - Therefore, it must refer to the catalog to know the structure of the files in a specific database, such as the type and format of data it will access.

Self-Describing Nature of a Database System

- Example:

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Insulation between Programs and Data, and Data Abstraction

- In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the structure of a file may require changing all programs that access that file.

- By contrast, DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from the access programs. We call this property program-data independence.

Insulation between Programs and Data, and Data Abstraction

- Example: a file access program may be written in such a way that it can access only STUDENT records of the structure shown in Figure.

Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

  - If we want to add another piece of data to each STUDENT record, say the Birth_date, such a program will no longer work and must be changed.

Insulation between Programs and Data, and Data Abstraction

- The characteristic that allows program-data independence and program-operation independence is called data abstraction.
- A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented.
- Informally, a data model is a type of data abstraction that is used to provide this conceptual representation.
- The data model uses logical concepts, such as objects, their properties, and their interrelationships, that may be easier for most users to understand than computer storage concepts.

Support of Multiple Views of the Data

- A database typically has many users, each of whom may require a different perspective or view of the database.
- A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.
- A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining multiple views.

Support of Multiple Views of the Data

- Example:

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Support of Multiple Views of the Data

- Example:

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Support of Multiple Views of the Data

- Example: One user of the database of Figure may be interested only in accessing and printing the transcript of each student; the view for this user is shown.

TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

Support of Multiple Views of the Data

- Example: A second user, who is interested only in checking that students have taken all the prerequisites of each course for which they register, may require the view shown in Figure

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

Sharing of Data and Multiuser Transaction Processing

- A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time.
- This is essential if data for multiple applications is to be integrated and maintained in a single database.
- The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.
- Example:
  - When several reservation agents try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one agent at a time for assignment to a passenger.
- A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly and efficiently.

Sharing of Data and Multiuser Transaction Processing

- The concept of a transaction has become central to many database applications.
- A transaction is an executing program or process that includes one or more database accesses, such as reading or updating of database records.
  - Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions.
- The DBMS must enforce several transaction properties:
  - The isolation property ensures that each transaction appears to execute in isolation from other transactions, even though hundreds of transactions may be executing concurrently.
  - The atomicity property ensures that either all the database operations in a transaction are executed or none are.