Chapter 2. Concepts & arch of DB sys

1. Introduction

processing & analysis of spatial data – becoming increasingly dependent on the use of DBMS rather than conventional GIS

→ WHY ? 1. cost effective sys security, DB integrity, backup & recovery, data replication
2. close integration w/ mainstream business computing environments

2. DBs & DB sys

2.1 DB terminology

problem space : business functions the DB is designed to address

data model : describing the problem space in a comprehensible way

DB schema : description of the DB including its tables & the relations among them

DB engine (or DB server): a collection of programs that manipulate the data in DB

data dictionary : describes the contents of the DB

DB integrity rules : rules to be enforced to protect the data in the DB

stored procedures : blocks of SQL code for defining, managing & querying data in DB

user interface : DB front-end by which users access & interact w/ DB

middleware : communication SW tools that support data transmission & data processing over networks

DBMS : composed of DB, DB engine, user interface, application programs, middleware

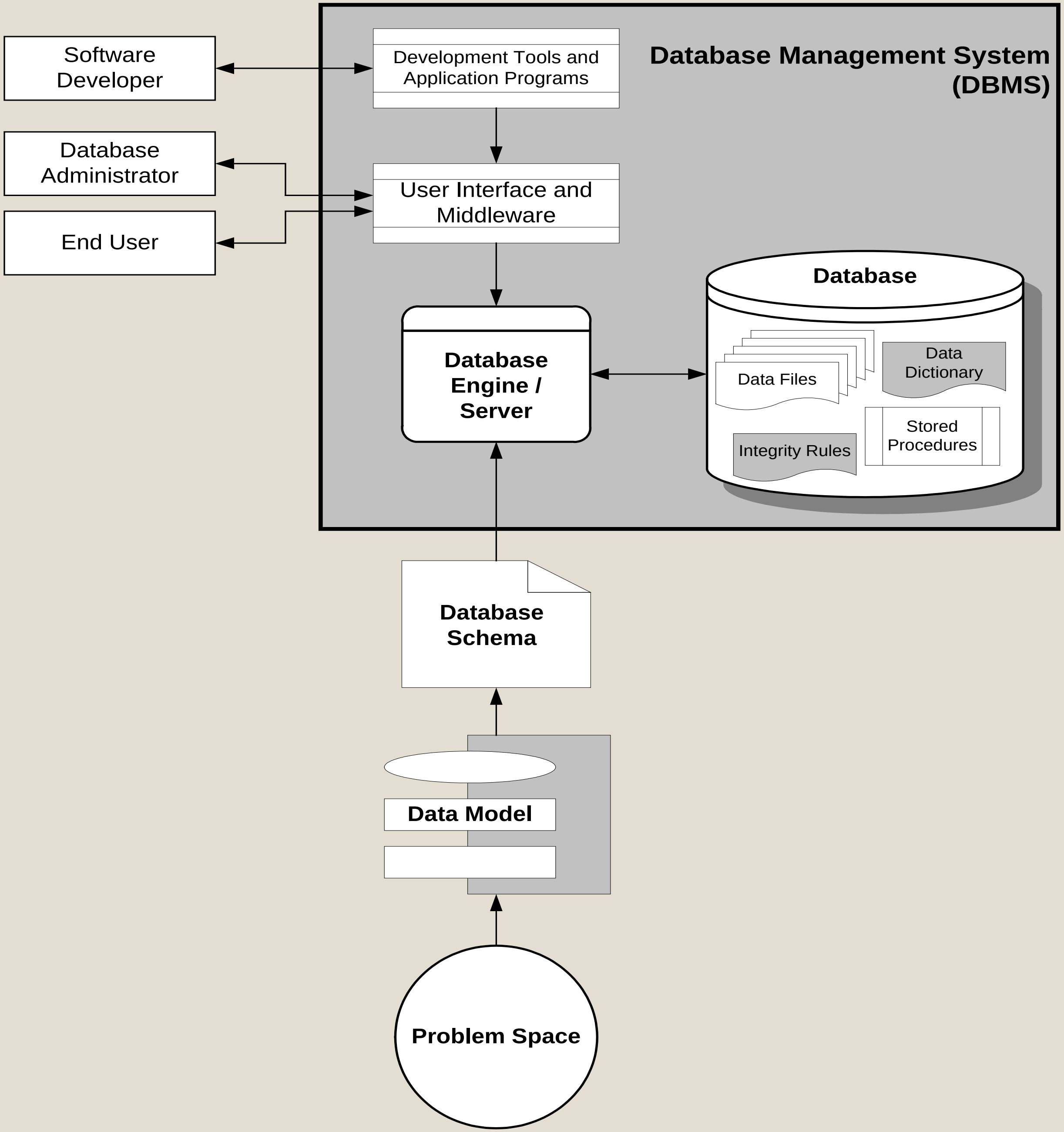


Figure 2-1. Database terminology

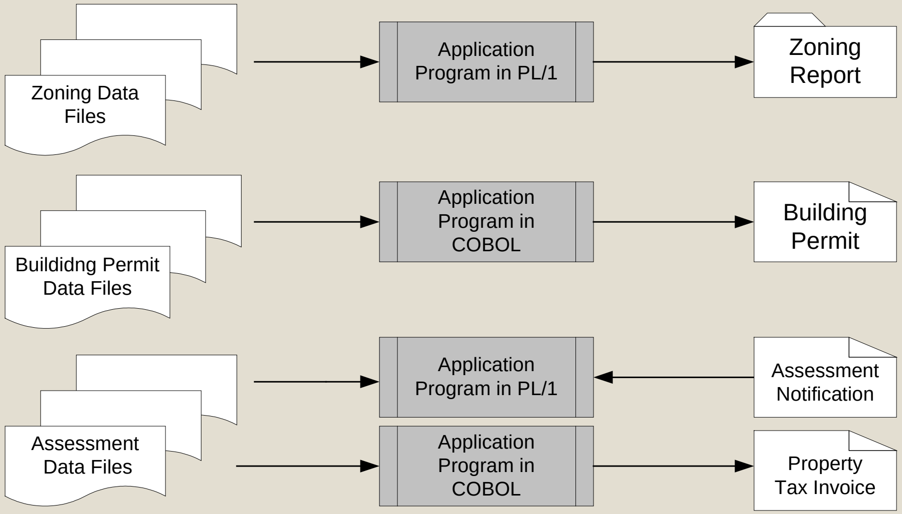
Chapter 2. Concepts & arch of DB sys

2.2 Computer data organization & DB

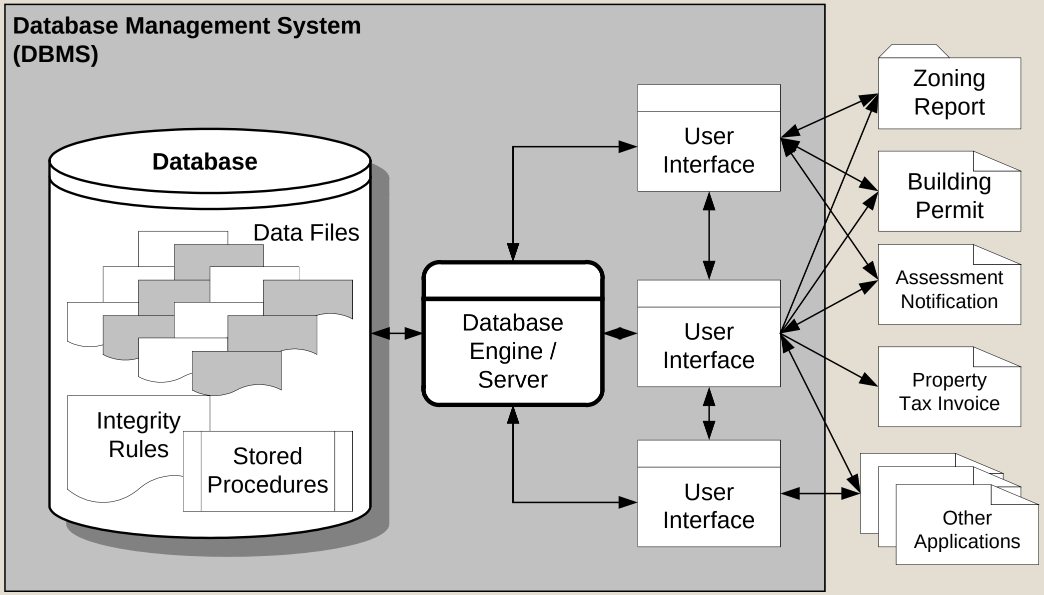
In early days – data were organized as independent data files – file processing systems

development of modern DB

- stores data + characteristics of data + integrity rules
- separate data from applications – independence from business functions
- provide essential tools for info resources or asset management in business, education, government



(a) File-based data processing



(b) Integration of data files and processing procedures in a DBMS

Figure 2-2. File & DB processing

2-3 Classification of DB systems

Table 2-1. Classification of database systems

Classification Criteria	*Categories*
Data Models	o Hierarchical Systems o Network Systems o Relational Systems o Object-oriented Systems o Object-relational Systems
Primary Database Functions	o Data Storage or Inventory Systems o Transaction Systems o Decision Support Systems
Nature of Data	o Spatial Information Systems o Non-spatial Information Systems
Objectives of Information	o Custodial Systems and Data Warehouses o Project-oriented Systems
Hardware Platforms and Systems Configurations	o Distributed Systems o Desktop Systems

3. DB operations

3.1 DB storage & manipulation

DB - large volumes of data   ex. Banking & retailing : terabytes, RS : several petabytes/year

thus, DB system usually stores data in secondary storage devices

(cf. central processing unit (CPU) – primary storage)

even bigger is tertiary storage devices : terabyte storage capacities

transmission of data – by disk blocks : storage locations of 4,000-16,000 bytes of data

not by data files

DB systems control the storage of and access to data by means of the DB engine

: 2 data manipulation SW in DB engine

a. buffer manager : handles the main memory by allocating the data read from secondary storage to a specific page

b. file manager : keeps track of the location of data files & their relationships w/ disk blocks

3.2 DB security & integrity constraints

designed to protect the data in DB from being corrupted, compromised, destroyed

security : all rules & measures that are designed to protect the DB against unauthorized use, modification, destruction of contents

discretionary security : controls the ability of users to access specific data files ex. read-only read-write

mandatory security : classifies users & data into different security levels - access control

integrity : to protect the value of the data by safeguarding their accuracy, correctness, validity

enforced by applying certain rules – called business rules

form an integral part of the DB schema

3 integrity constraints – domain c. : specify the types of data values ex. Numeric, character, Boolean

key & relationship c. : govern the use of entities as primary, secondary, foreign keys

semantic integrity c. : written rules stating what is allowed & not allowed in data structure & data management

Chapter 2. Concepts & arch of DB sys

3.3 DB query

query : question / task a user asks of a DB

handled by a DB tool – called query manager : turn users SQL into a sequence of operations

query manager – perform query optimization : answer the query in the most efficient way

use an index created for the DB

query is made up of one / more operators

ex. relational DB model – 8 operators

: SELECT : lists all of the row / those which match specific criteria

PROJECT : generate a subset of columns from a table, removing duplicate values from the result

JOIN : combine one row of a table w/ rows from another table, use columns relationships

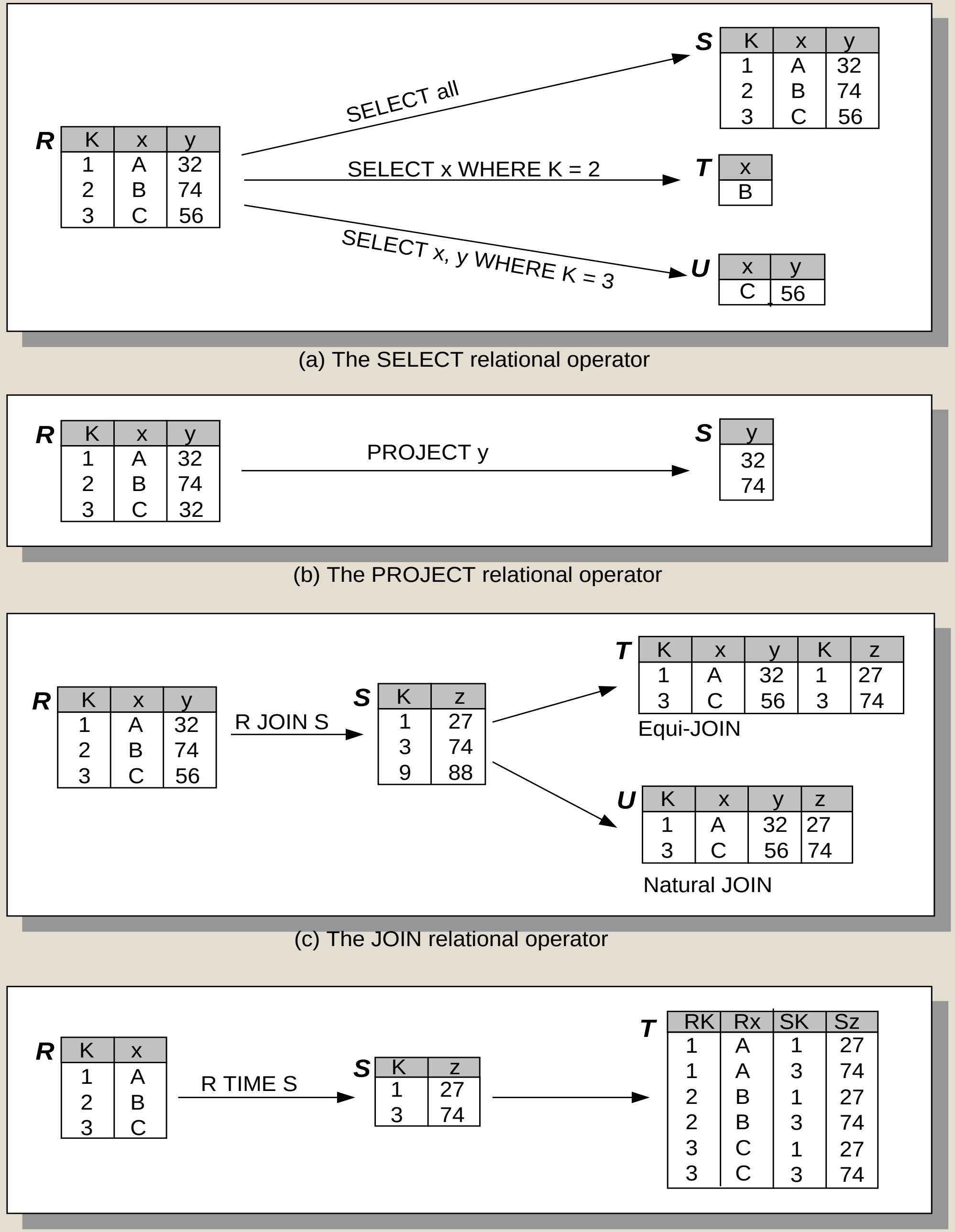
PRODUCT : concatenate every row in one table w/ every row in another table

UNION : generate a new table by appending rows from one table w/ those of another table

INTERSECT : generate a new table consisting of all rows appearing in both of two tables

DIFFERENCE : generate a new table consisting of all rows that appear in the 1^{st} table but not in the 2^{nd} of two tables

DIVIDE : create a new table consisting of all values of one column of the binary table that match, in the other column all values in the unary table



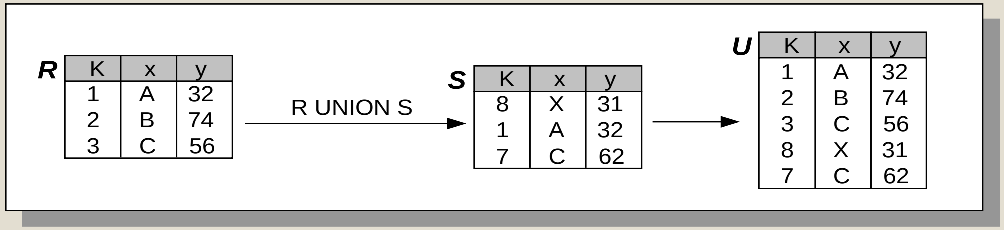
(a) The SELECT relational operator

(b) The PROJECT relational operator

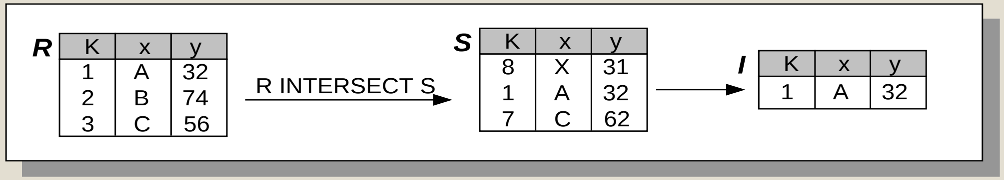
(c) The JOIN relational operator

(d) The PRODUCT relational operator

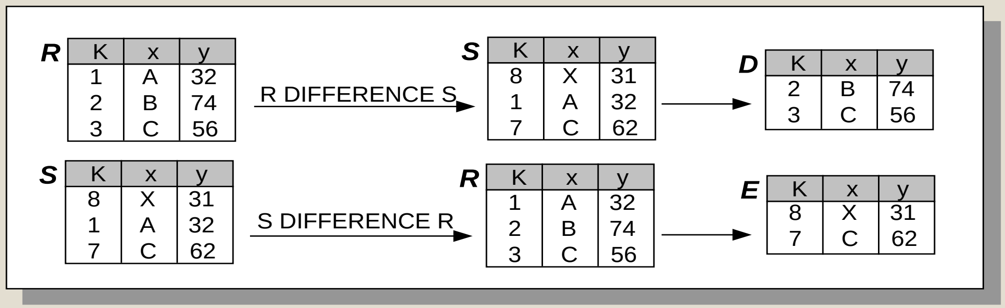
Figure 2-3. Relational operators



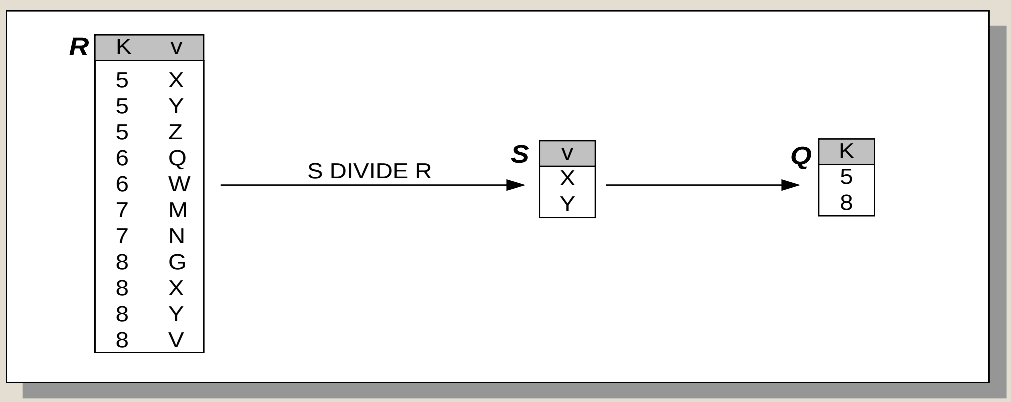
(e) The UNION relational operator



(f) The INTERSECT relational operator



(g) The DIFFERENCE relational operator



(h) The DIVIDE relational operator

Figure 2-3. Relational operators

3.4 DB transactions

transactions : changing values in DB

transaction design principles

: atomicity : transaction can never be completed only partially

consistency preservation : data remain in a consistent state as specified by the DB schema, constraints, integrity rules

isolation : transactions to be independent of each other

durability / permanency : after transaction complete, its results can always be traced

transaction is controlled by the transaction manager of the DB engine

4 transaction control mechanisms

: concurrent control : locks the data items involved in the transaction

logging the transactions : keeps track of all changes made to the DB in a redo log

transaction commitment : prevents any changes to the DB unless the transaction is ready to complete

rollback : allows the DB to undo an incomplete transaction process

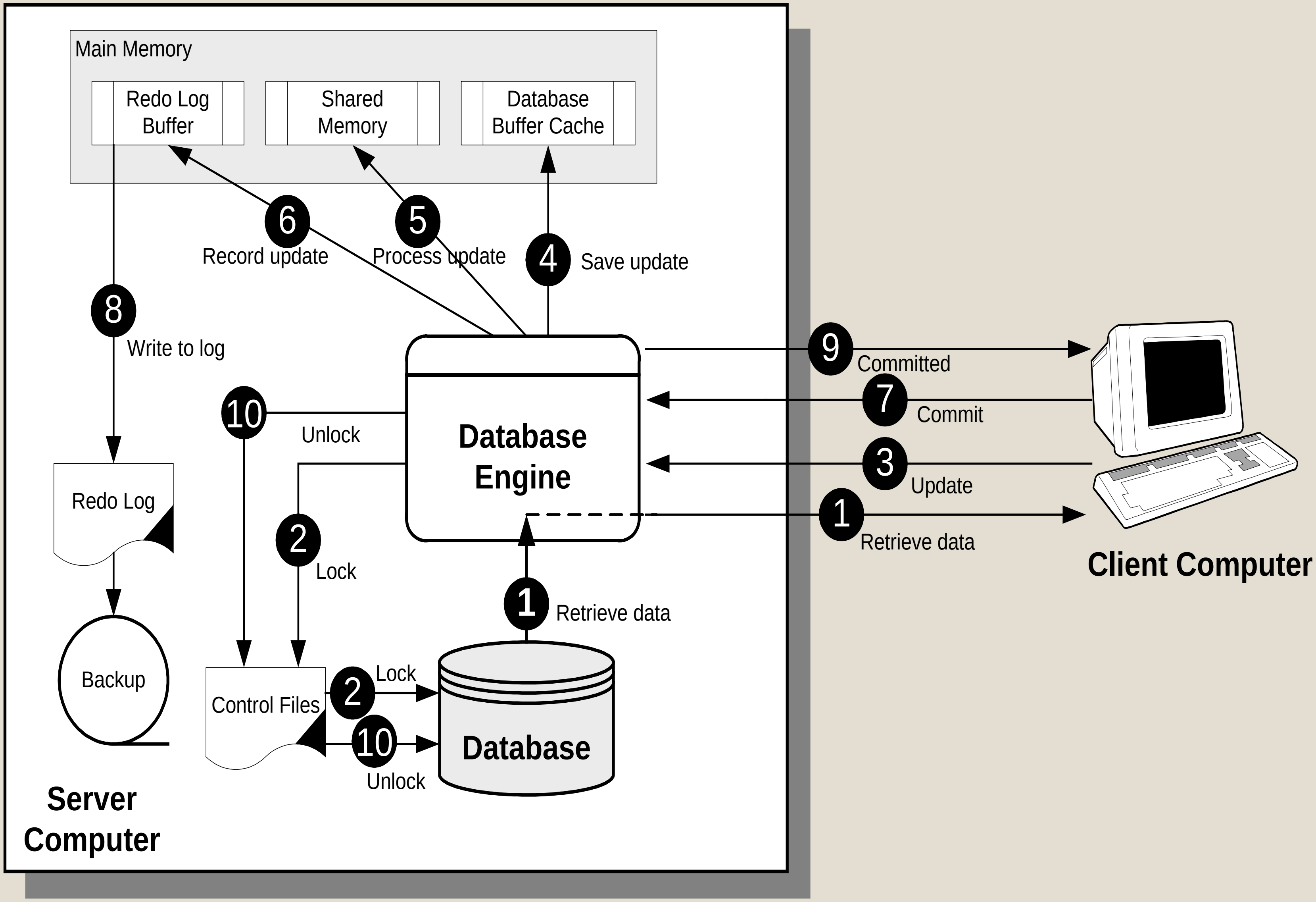


Figure 2-4. Steps in a DB transaction

3.5 DB backup & recovery

to restore the DB in a catastrophic failure such as a disk crash

copy periodically the entire DB & the transaction log on to an external storage medium

3.6 DB replication & synchronization

to support business needs in distributed DB systems : improve system performance, DB availability

process of making a copy of a DB onto one / more additional computers located at different sites

3.7 Structured query language (SQL)

standard language for querying & managing DB

non-procedural computer language – does not have IF, FOR, WHILE, GOTO, CASE

DB sub-language of about 200 words

SQL statements can be used in 5 ways

: interactive processing thru a command-line user interface

embeded in a high-level computer language

using a call level interface(CLI)

using 2 standard Java application programming interface(API) protocols : Java DB connectivity(JDBC)

embeded SQL for Java(SQLJ)

in a stand alone application program modules in the form of stored procedures, function, packages

used in typical DB operations

: DB query, data definition, data manipulation, DB connection & access control, data sharing, data integrity

SQL Function / Example	Explanation
(a) Data retrieval **SELECT** parcel_id, area **FROM** lu_2002 **WHERE** lu_code = 'agr'	*These SQL commands retrieve the identification numbers and areas of parcels whose land use code is 'agr' (agricultural) from data table lu_2002*
(b) Data definition **CREATE TABLE** lu_2002 (parcel_id **VARCHAR2(8) PRIMARY KEY,** lu_code **VARCHAR2(3) NOT NULL,** area **NUM(8,2) NOT NULL,** survey_date **DATE DEFAULT SYSDATE**)	*The SQL command* **CREATE TABLE** *specifies the structure and data types of the data table lu_2002. In the example,* **VARCHARS, NUM** *and* **DATE** *specify the data types of the four columns of the table; the numbers in the brackets specify the number of characters and digits string and numeric data types respectively. The value of* survey_date *will be set to the system date automatically if no date value is supplied duirng data entry. Note how the constraints* **PRIMARY KEY** *and* **NOT NULL** *are specified.*
(c) Data manipulation **INSERT INTO** lu_code **VALUES** ('12322', 'res', 4500.00, '06-may-2002')	*These SQL statements add data into the table lu_2002. Character strings are put inside quotation marks but numerical values are not. Since a date value is given, the default system date as specified in the data structure in the previous example will be not used.*
(d) Database connection and access control **CONNECT** system/passwordxxx **CREATE ROLE** db_maintenance **GRANT INSERT**, **UPDATE ON** lu_2002 **TO** db_maintenance **GRANT** db_maintenance **TO** john.young	*These SQL commands connect to the system, create a database maintenance role, and give this role to a user called john.young*
(e) Data sharing **CREATE TRIGGER** copy_data **AFTER INSERT ON** lu_2002 **FOR EACH ROW** **BEGIN** **INSERT INTO** lu_2002_copy@fes.uwaterloo.ca **VALUES** (:new.parcel_id, :new.lu_code, :new.area, :new.survey_date); **END**; **/**	*This sequence of SQL commands copy newly input data to a data table at a remote site*
(f) Data integrity **CREATE TABLE** lu_2002 (parcel_id **ARCHAR2(8) PRIMARY KEY,** lu_code **VARCHAR2(3) NOT NULL,** area **NUM(8,2) NOT NULL,** survey_date **DATE DEFAULT SYSDATE,** **CONSTRAINT UNIQUE** (parcel_id), **CONSTRAINT** area_chk **CHECK** (area > 0.00)).	*This example shows how data input constraints are imposed to ensure that the parcel_id is a unique number for each land parcel and that the area is not a negative value*

Figure 2-5. Examples of DB operations using SQL

Chapter 2. Concepts & arch of DB sys

4. HW & SW arch

4.1 Centralized & distributed DB arch

1) Centralized DB arch

early generations of DB systems

all processing is done in a single computer, all data are stored in the same secondary memory

2) Distributed DB arch

since 1990s, most DB systems today

include several components

: computers, OS, DB system SW, network cards & SW, communication network protocols(ex. TCP/IP),
DB engine, transaction processor(manager)

allow physical separation of processes & data

allow partitioning of a particular process / data file into smaller units

it has a number of transparency features

: distribution transparency : access any DB w/o knowing where & how the data are stored
performance transparency : behave as if it were a single centralized DB system
transaction transparency : update the DB at different sites
heterogeneity transparency : integration of different DB systems under a common schema

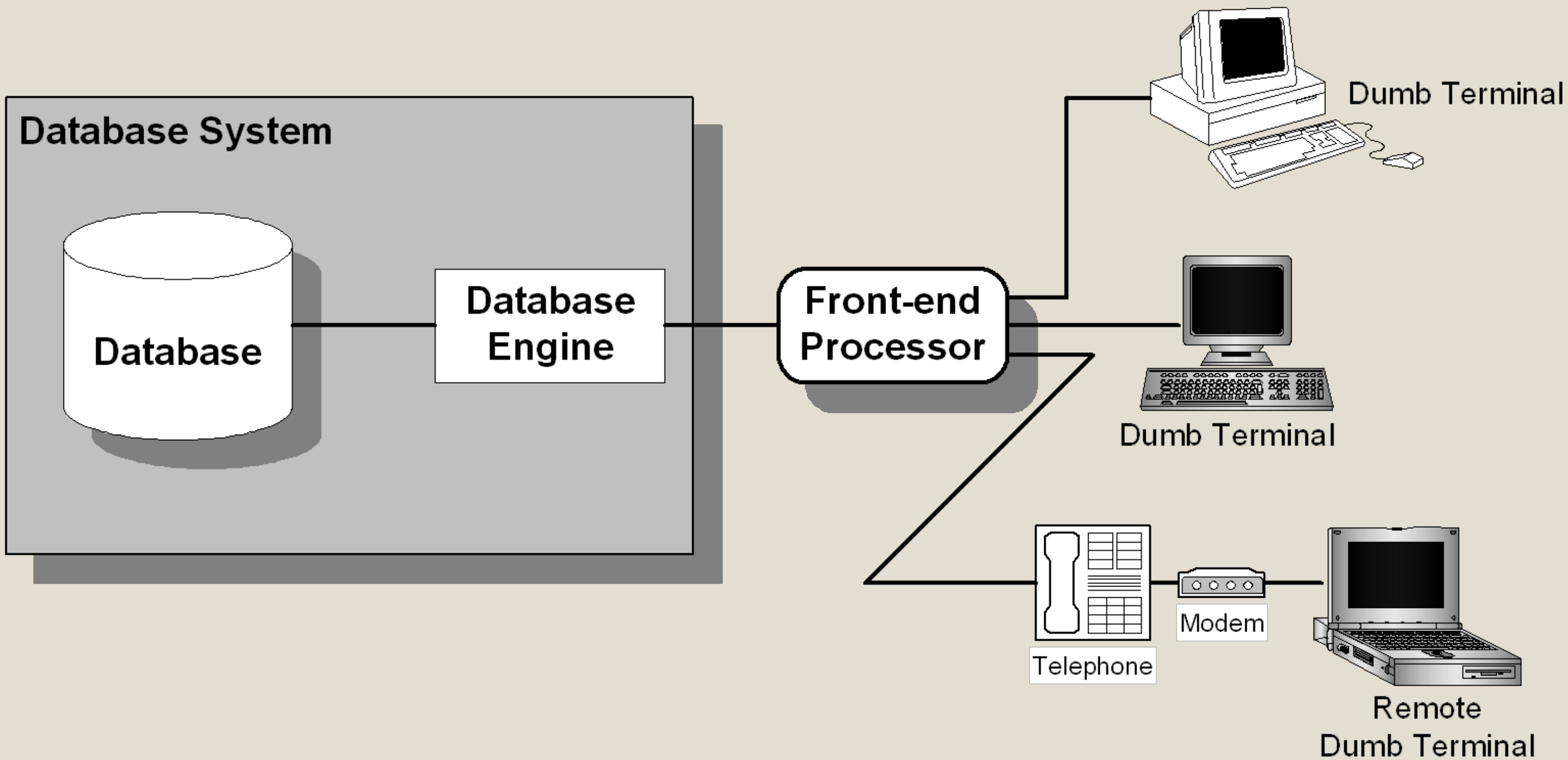


Figure 2-6. Configurations of a typical centralized DB system

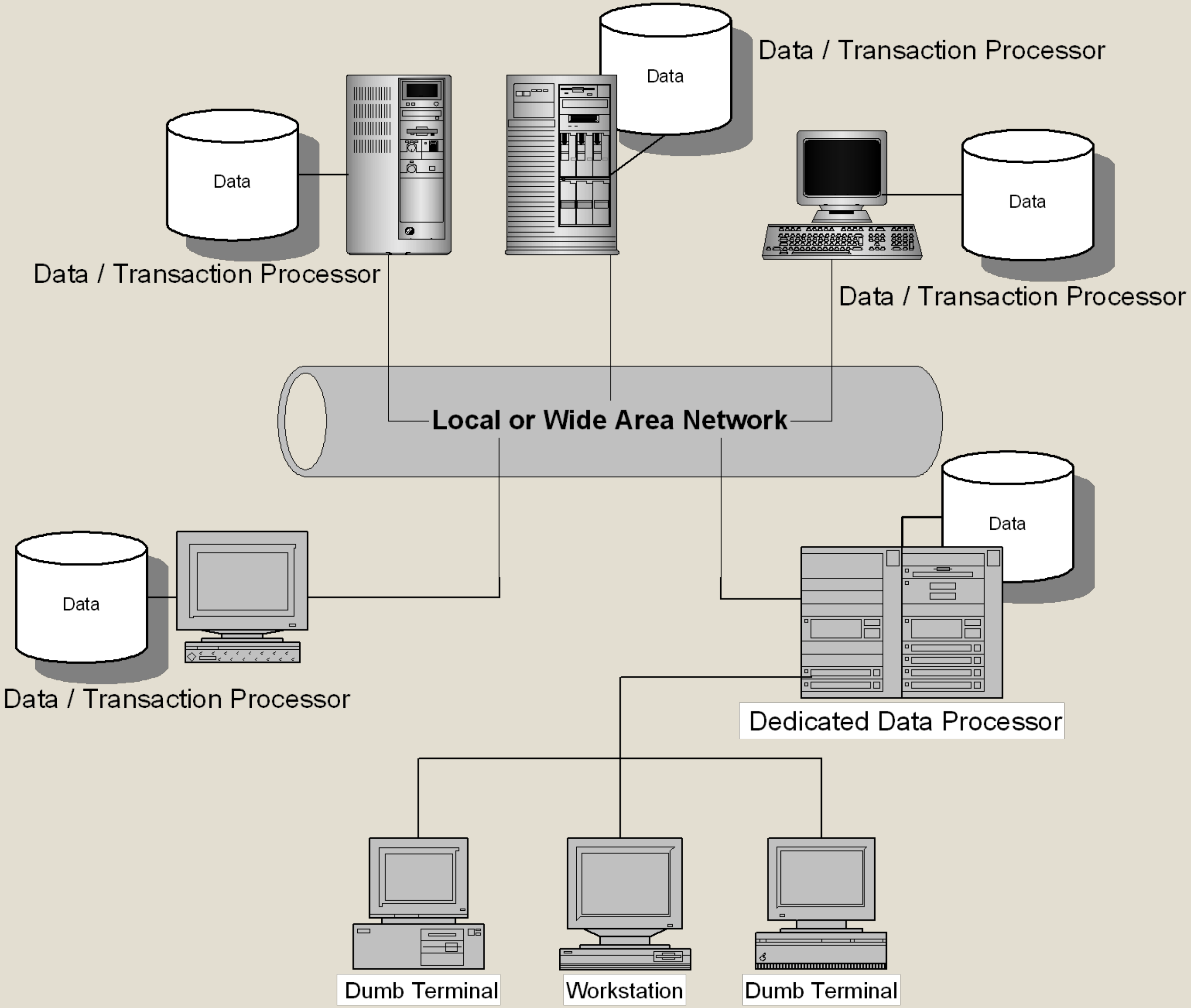


Figure 2-7. Configuration of a typical distributed DB system

4.2 Client/server computing

client – requests services, server – provides services

can be thin / fat client / server

two-tier client / server arch : one server one client

three-tier client / server arch : extension of the two-tier

client : used for interaction w/ the DB

application server : application programs are stored & executed

DB server : used for storage & retrieve of data

web-based DB systems are three-tier

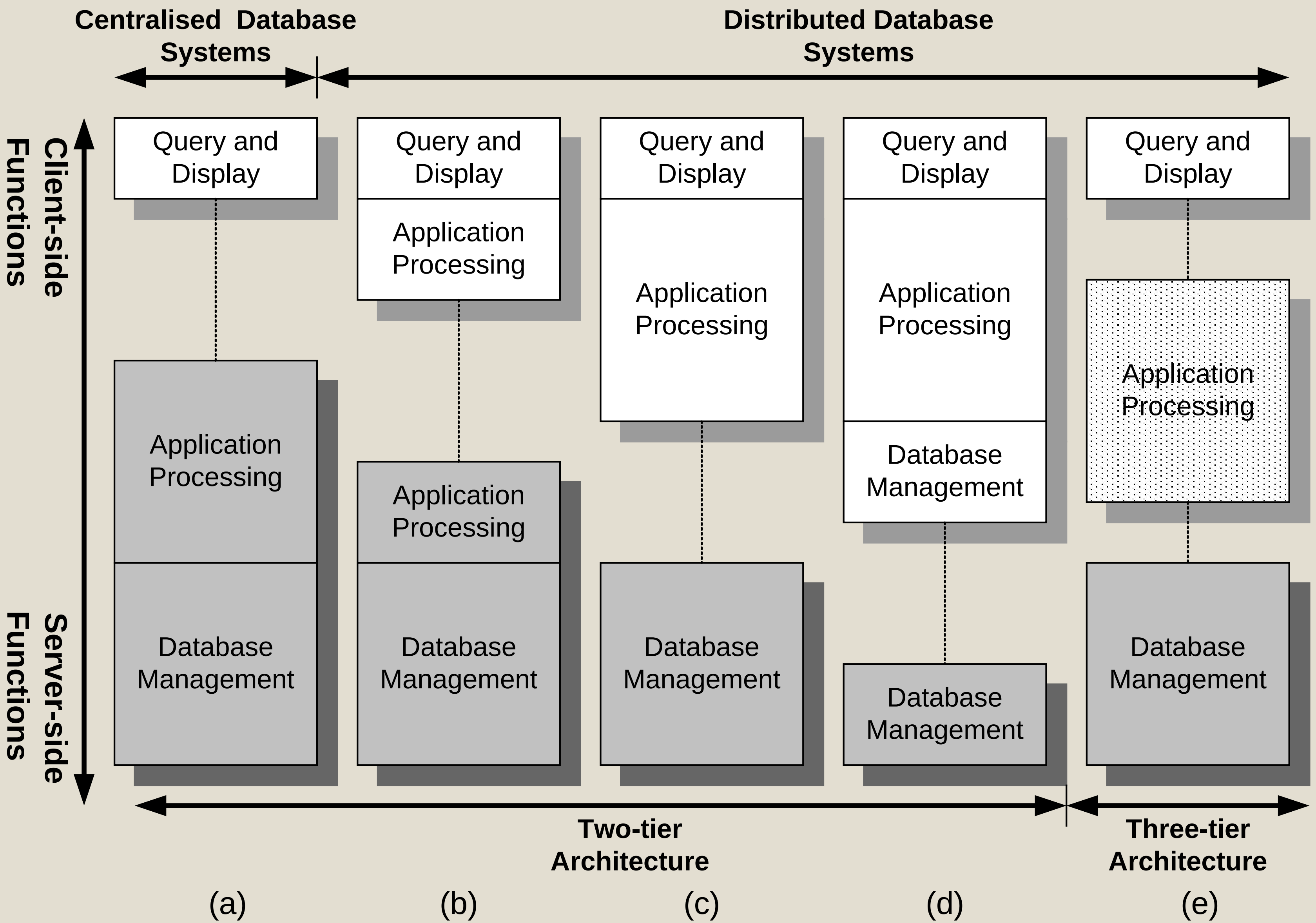


Figure 2-8. Client-server computing for DB systems

4.3 DB SW

DB SW comprises different modules of application programs

- DB engine : storing, retrieving, manipulating, conversion, transaction logging, memory management
- SQL (or program written in SQL extensions) : managing data storage & retrieval routines
- network middleware : DB connectivity middleware ex. CORBA, JDBC, ODBC, ADO
  - * middleware tool : made up of API on client / communication SW on client & server

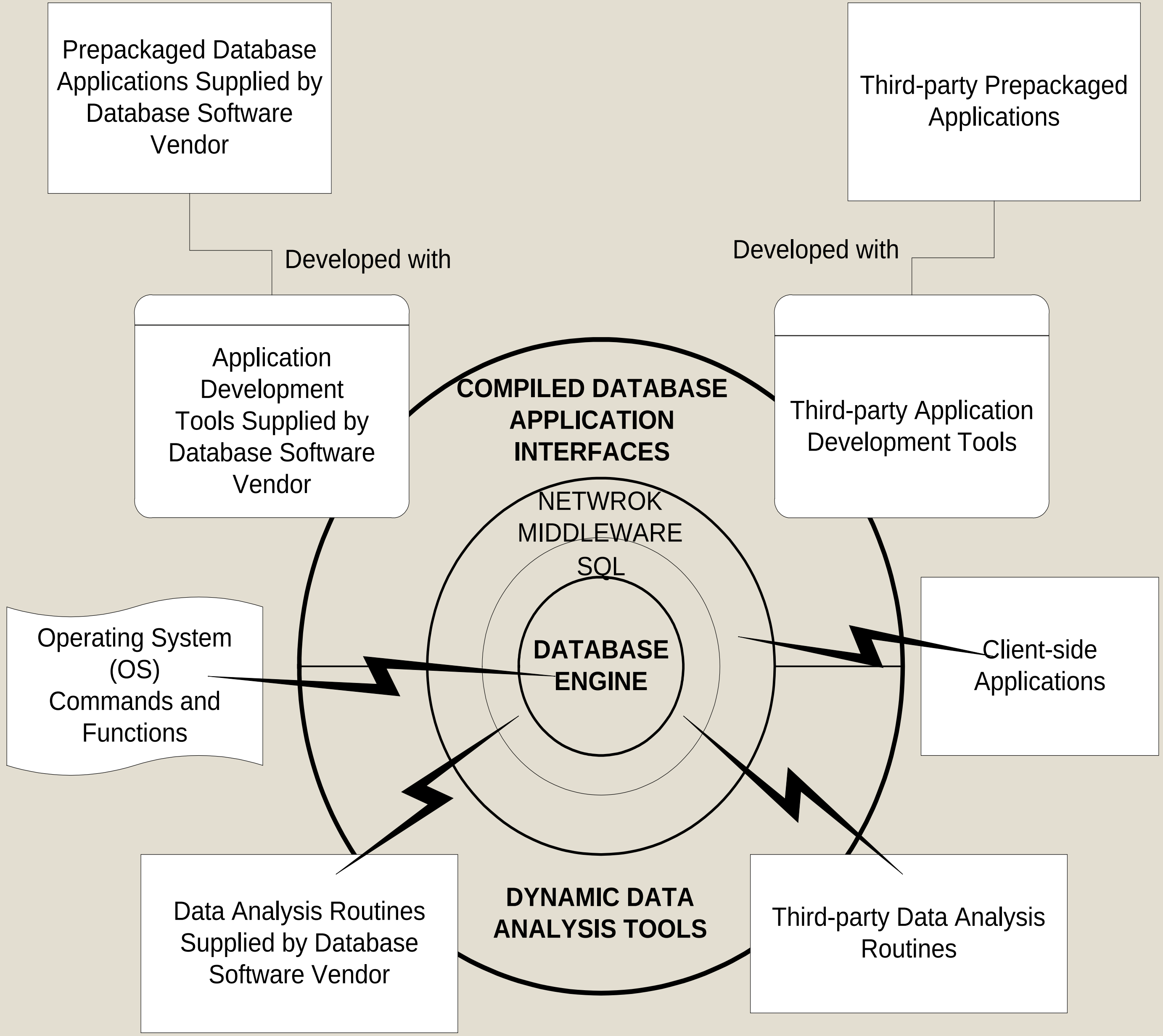


Figure 2-9. SW layers of a DB system

4.4 Web-based DB arch

WWW has significant impact on the development of DB system

Web browser interface enables a user to access data anywhere in the world

: *universal access arch* is introduced based on Internet standard

web-based DB – accessible to internal & external users

three-tier client/server configuration

components : a. client computer – user submits a request by HTML-formatted page thru HTTP

b. web server – equipped w/ program called a *server-side extension*

= *web-to-DB middleware* : understand, validate, process DB queries

using CGI / API protocol

functionality of web browser can be enhanced by adding *client-side extensions*

: plug-ins, Java, JavaScript, ActiveX, VBScript

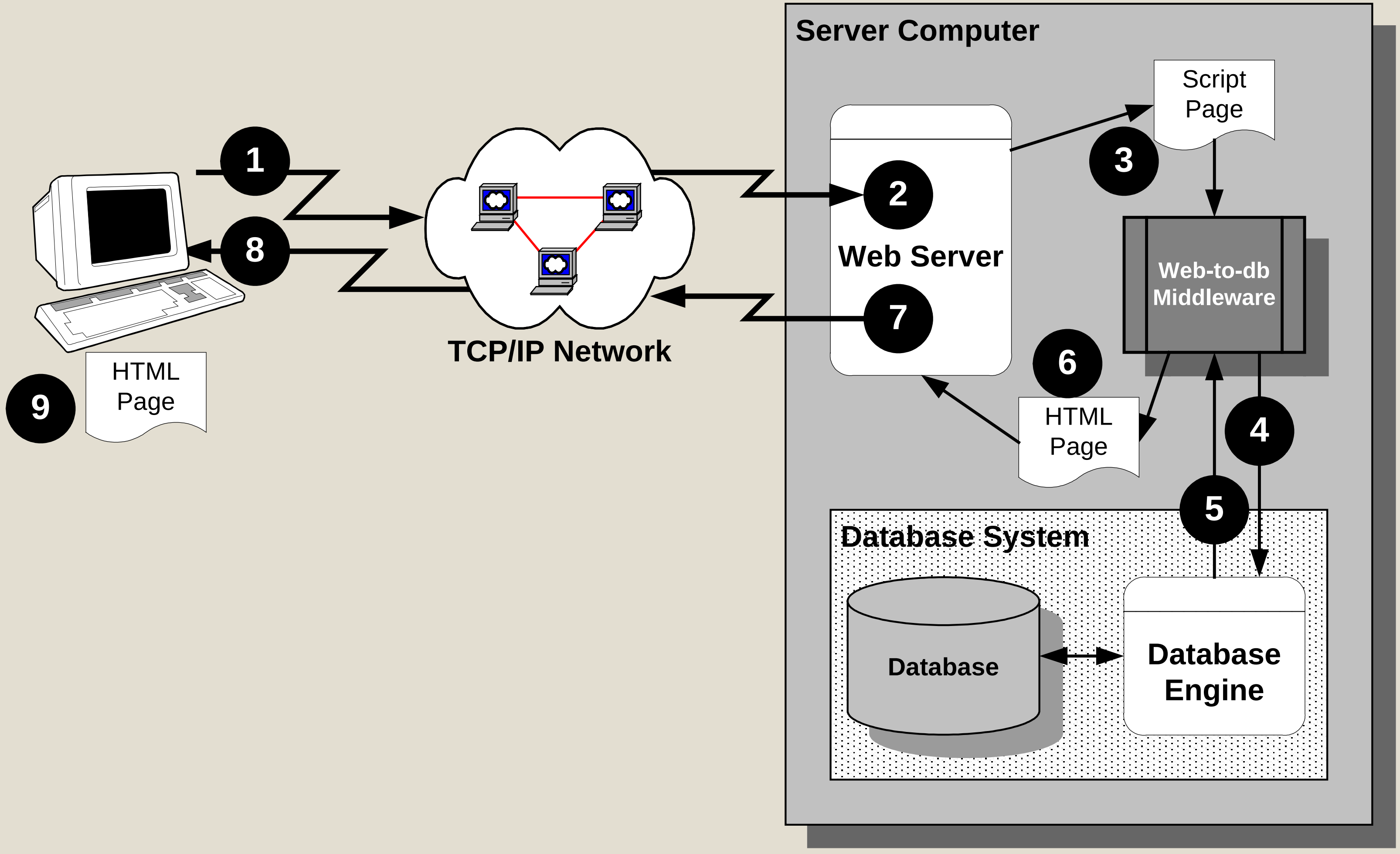


Figure 2-10. Arch of a typical web-based DB

5. Data structure

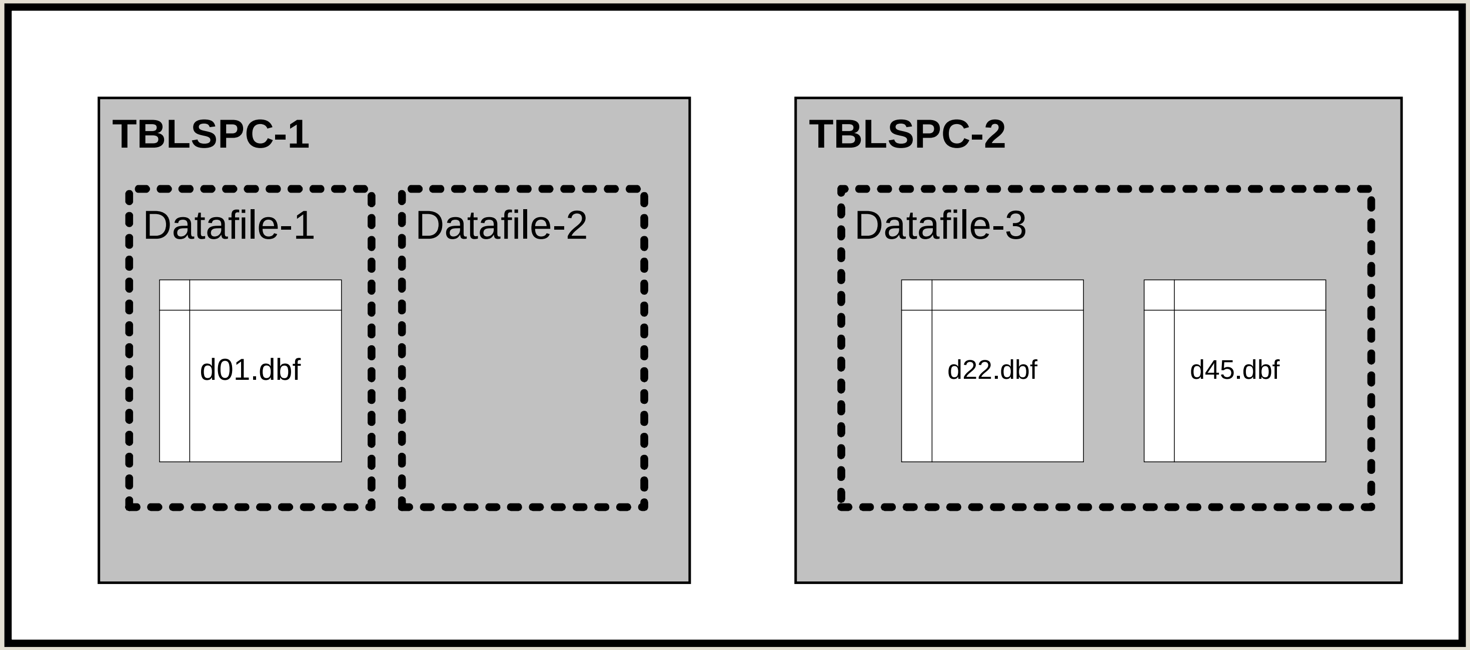
5.1 Logical data structure

how data are organized in a DB for optimal performance & ease of administration

ex. Oracle Optimal Flexible Arch

: logical structure

table space – data file - dbf



TBLSPC-1 consists of two datafiles Datafile-1 and Datafile-2, and the table d01.dbf is allocated to Datafile-1

TBLSPC-2 consists of one datafile Datafile-3, which houses two tables d22.dbf and d45.dbf

Figure 2-11. Relationship between tablespace, data files & tables in Oracle's optimal flexible arch(OFA) logical DB structure

Chapter 2. Concepts & arch of DB sys

5.2 Physical data structure

actual organization & placement of data files in the DB

dependent on the model of its DB system

ex. relational – values of attributes are stored in a table in a certain data type

* data types
  - character / string data type
  - numeric data type
  - date data type
  - others including BLOB & user-defined abstract data type(ADT)

5.3 DB indexing

an index = an element of data structure, used to speed up access to a specific part of the DB

many indexing methods have been proposed

ex. B-tree – the most commonly used form

use case) users issue a command to index → column for a row identification + index table are created

index table : root block – branch block – leaf block structure

since all leaf are at the same depth – all retrieval require the same amount of I/O

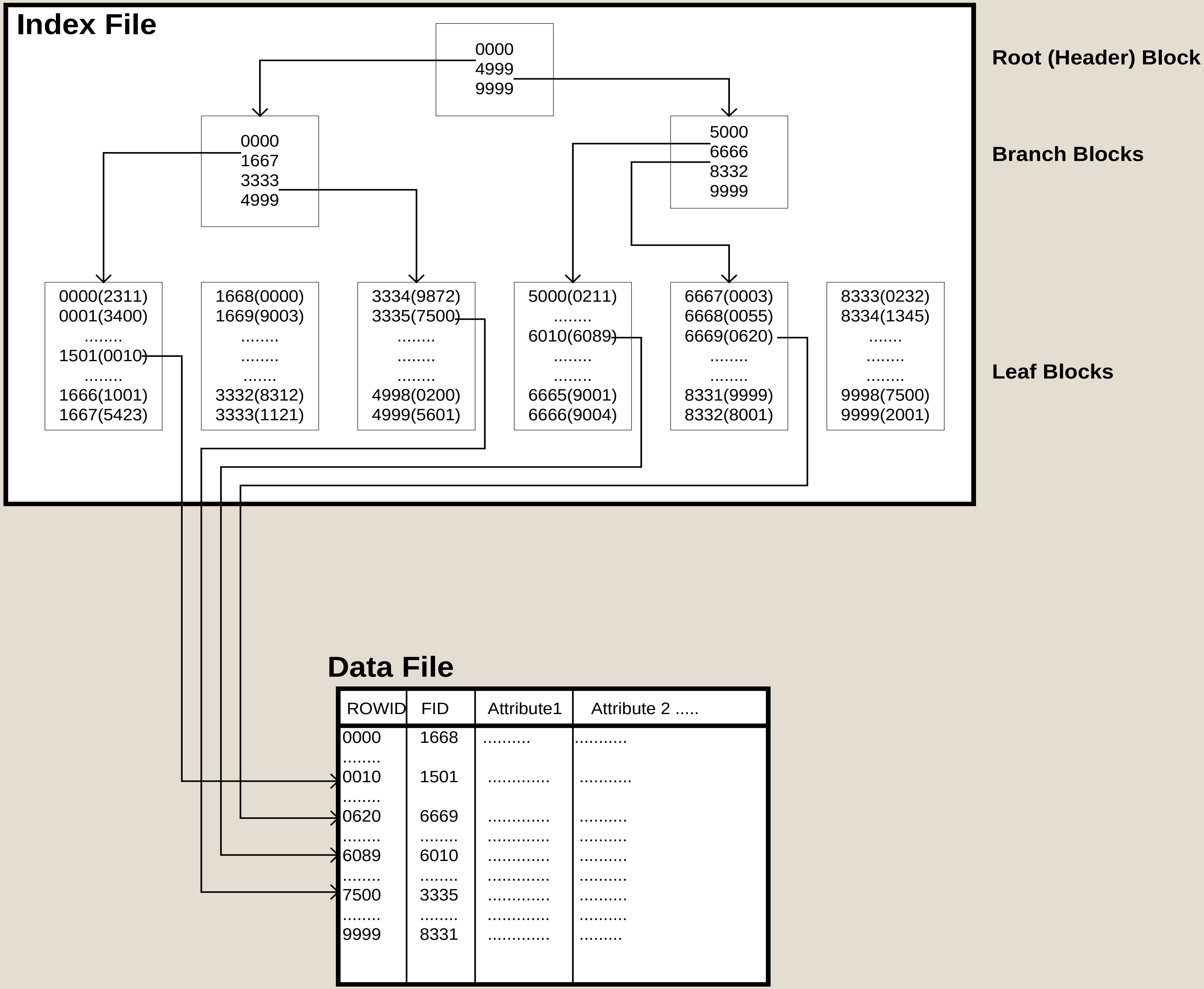


Figure 2-12. A B-tree index