The person who is independent from the DBMS in the organization with more focus on the management side :

A- DA

B- System administrator

C- DBA
D- DDBA

The I/O cost to find the 20 ``excellent`` student (avg >90) from table size 200 using index scan is:

A-

B

40

C

220

D

60

E

200

One of the following can be used to support routine decision making for business

A

OLTP

B

OLAP

C

DB

D

DDBMS

Q= ``UPDATE student SET age= 10`` can be

A

Action

B

Data base query

C

Transaction

D

All correct

Social engineering is considered

A

breach

B

threat

C

Risk

[✓ D]

[✓ vulnerability]

E

Measure

Changing password periodically (e.g. every year) an example of

[✓ A]

[✓ Policy]

B

Standard

C

Procedure

D

All correct

Summing the ``weekly sales of iPhones in Amman city into one month sales=1200 USD`` is an ______ OLAP operation

A

Dice

B

Slice

C

Drill down

[✓ D]

[✓ Roll up]

The internal and external data is stored in the data warehouse __ performing ETL operations

A

Before

B

After

C

Before and after

D

No ETL operation here

One of the following is the best choice to be an index for an employee table

A

Address

B

Gender

C

Name

D

All correct

Product sales data can be_____ sources of the data

A

external

B

Internal

C

External + Internal

D

Not a source of data

DB Consistency are maintained by

A

DB users

B

DBA

C

DBMS

D

None of all

the transaction log is updated for ____transactions

A

Write only

B

read only

C

Read and write

D

Nested

A transaction may consist of the following:

A

All correct

B

UPDATE statements

C

INSERT statements

D

SELECT statement

Q=``Select ID from EMP where age=20`` is an example of_______fragmentation

A

horizontal

B

mixed

C

vertical

D

All correct

E

not fragmentation at all

____handles all submitted requests to the server

A

Scheduler

B

query optimizer

C

User process

D

Listener

E

lock manager

One of the following SQL code is faster when executed

A

Age > 22

B

Age between 10 and 30

C

Age = 17

D

age in (select age from B)

E

Age = min(X)

---

the optimizer will select the best plan for execution during the _____phase

A

Execution

B

Parsing

C

Fetching

D

All phases

---

the optimizer will select the best access plan for execution from ____

A

Data cache

B

SQL cache

C

Server cache

D

Client cache

Designing and implementing databases and applications is a DA ____ Role

A

Managerial

B

Technical

C

Cultural role

D

Not DA role

One of the following concurrency control solutions is more suitable for simple system with few update requests

A

Optimistic

B

Time stamp

C

Locking method

D

All correct

____means when a transaction finished the changes and committed, they cannot be lost in the system failure.

A

Atomicity

B

Isolation

C

Consistency

D

Durability

E

all correct

Protecting data from any type of unauthorized access is more related to:___

A

availability

B

confidentiality

C

integrity

D

security

E

consistency

The data stored in the data warehouse (e.g payments) are ___-oriented

A

Subject

B

object

C

system

D

user

E

analytical

One of the following Lock granularity level achieve lower concurrency

A

DB-level lock

B

page-level lock

C

Row -level lock

D

Table-level lock

E

field -level lock

Q=``SELECT * FROM E5 WHERE EMP_DOB = `01-JAN-1990` `` is____ transparency

A

Local mapping

B

Fragmentation

C

Location

D

None of all

Keep tracking the number of complaints by customers about a product at a time is the responsibility of

A

Data analytics

B

Data visualization

C

Data monitoring

D

Data reporting

____DDBMSs integrate di?erent types of DBMSs over a network that support the same data model.

A

Heterogeneous

B

Fully-heterogeneous

C

Homogeneous

D

All correct

E

Fully-homogeneous

an organization can have Multiple database administrators at the same time based on the:

A

number of DBMS

B

Size of organization

C

number of DBA

D

All correct

One of the following is true about ?The sales of iPhones in January in Amman city was 1200 USD?

A

city is fact

B

city is Dimension

C

city is Data mart

D

City is concept hierarchy

DBMS performance tuning performed on the ___ and SQL performance tuning performed on the ______side

A

Client , Client

B

Client, Server

C

Server ,Client

D

Server, server

_____refers to the process of deciding where to locate fragments

A

Data replication

B

Data fragmentation

C

Data allocation

D

All correct

E

data segmentation

The __ is the software component found in each computer that store or retrieve data located at that location.

A

Data manager (DM)

B

Data processor (DP)

C

Transaction processor (TP)

D

DP + TP

E

DP + TP+ DM

Q=`` UPDATE PRODUCT SET PROD_QTY = PROD_QTY ? 1 WHERE PROD_NUM = ?231785?; INSERT INTO INVOICE (CUS_NUM, INV_DATE, INV_TOTAL) VALUES ?100?, ?15-FEB-2016?, 120.00;`` the Q represent:

A

Remote request

B

Distributed transaction

C

Remote transaction

D

Distributed request

E

all correct

---

any access plan that the optimizer will select consists of :

A

SQL Statements

B

configurations

C

instructions

D

procedures

E

all correct

One of the following transaction sequence of execution represent concurrency in the system at sequence of time (1,2,3,4,5)

A

T1, T1, T2, T2, T1

B

T1, T1, T1, T2, T2

C

T2, T2, T1, T1, T1

D

All correct

The ETL operation performed for external and internal data in order to perform

A

business continuity

B

business decision making

C

Competitive advantage

D

All correct

One of the following is true about decision support data

A

High Query activities

B

Un-summarized data

C

Not complex data

D

Real time data

E

none of all

___ occur if transaction T3 calculated a summary function for data while another transaction (T1) was updating the same data

A

Read uncommitted data

B

Lost Updates

C

Inconsistent retrieval

D

All correct

Using __ is the most important action to improve the Database performance

A

Indexes in the table

B

Best Query reformulation strategies

C

Best Optimizer choice

D

ID Access plan

E

all correct

One of the following is a requirement for using distributed DBMS

A

complex Business environment

B

On demand transactions

C

Low cost smart phones

D

All correct

One of the following DB represent a good performance tuning state

A

the response time of the DB =40

B

the response time of the DB =410

C

the response time of the DB =10

D

All correct

The data in data warehouse is:

A

volatile and read-only database

B

None volatile and read-only database

C

volatile and write-only database

D

None volatile and write -only database

The best measure to handle Denial Of Service Vulnerability is:

A

Implement share access security.

B

Assign user access rights to workstations

C

Enforce complex password policies.

D

Perform periodic system backups.

E

all correct

one of the following is true about data sparsity

A

help determines the need for an index

B

advantage for the table index

C

problem in the table index

D

related to the data allocation

One of the following levels is not applicable level of data and process distribution

A

Single site process , single site data

B

Multiple site process, Multiple site data

C

Multiple site process, single site data

D

single site process , Multiple site data

he optimizer should select one the following plans for execution

A

Plane A with 2 steps and I/O cost 700

B

Plane B with 3 steps and I/O cost 900

C

Plane C with 4 steps and I/O cost 650

D

Plane D with 3 steps and I/O cost 700

One of the following is true about data warehouse and Data mart

A

One data mart consists of different Data warehouses

B

One data warehouse consists of different Data marts

C

different data marts can follow different Data warehouses

D

One data warehouses consists of one Data mart

Query and reporting, Data monitoring, Data visualization.

B

Query and reporting, Data visualization, Data monitoring.

C

Data visualization, Query and reporting, Data monitoring.

D

Data monitoring, Query and reporting, Data visualization.

E

any order

The right order to perform Data analytics is:

ne of the following is true about ``optimized query`` and ``original query``

A

Both have same results but the optimized query more efficiency

B

Both have same results and same efficiency

C

Both don`t have same results and the optimized query more efficiency

D

Both have same results but the optimized query less efficiency

The best solutions for distributed DB design is:

A

Data fragmentation, Data replication, Data cleaning, Data allocation

B

Data fragmentation, Data replication, Data allocation

C

Data fragmentation, Data cleaning , Data allocation

D

Data fragmentation, Data replication, Data cleaning

13 The transacation log is not update for transaction that the date base

Answer : read and write

14 one of the following can be usedto support name routine decision making for business

Answer : OLAP

15 changing password periodically (e.g. every year) an example of

Answer : polices

16 one of the following DB represent a good performance tuning state

Answer : the response time of DB =0

17 select salary from e3 where DOB <18 is transparency

Answer : location

18 the following are solution for distributed DB design except :

Answer : data cleaning , data allocation

19 employee salaries data can be sources of the data

Answer : internal

20 the best measure to handle discover password vulnerability is

Answer : user door locks

21 the person who is independent from the DBMS in the organization with more focus on the management side :

Answer : DA

1**one of the following is true about decision support data

answer:high query activities

2**the I/O cost to find the 90 ?good? Student (grade=78) from table size 290 using index scan is
answer:180

3**an organization can have multiple database administrators at the same times based on the
answer:number of DBA

4**converting the weekly sales of iPhone in Amman city into dally sales is an OLAP operation
answer:drill down

5**the plan that the optemizer will select consists of
answer:DB query

6**the optimizer will select the best acsess plan for execution to be saved in ::: answer: SQL cach

7**the data in data warehouse extracted from its sources before performing .
answer:ETP operations

8**the ETL operation performed for external and internal data in order to perform
answer: business continuity+making important decision+competitive advantage+increase numbers of customers (all correct)

9**one of the following is true about the sales of iPhone in January in NY state was 200 JOD ?
answer: state is dimension

10**one of the following is not suitable choice to be an index for an employee table
answer: all

11**one of the following transactions(T1 and T2) sequence of execution represent concurrency in the system in the system at sequence of time (0,1,2,3,4,5)except:
answer:T2,T2,T1,T1,T1

One of the following is true about decision support data
High Query activities

any access plan that the optimizer will select consists of
SQL Statements

The best measure to handle Denial Of Service Vulnerability is
Perform periodic system backups

the optimizer will select the best access plan for execution from
SQL cache

One of the following Lock granularity level achieve lower concurrency
DB-level Lock

refers to the process of deciding where to locate fragments
Data replication

One of the following is true about data warehouse and Data mart
One data warehouse consists of different Data marts

Changin password periodically (e.g. every year) an example of
Policy

Designing and implementing databases and applications is a DA Role
Technical

One of the following is a requirement for using distributed DBMS
All correct

the optimizer should select one the following plans for execution
Plane C with 4 steps and I/O cost 650

Product sales data can be sources of the data
Internal

1-DB consistency are maintained by ?

DB users

DBA

DBMS

None of all

2-the transaction log is updated for ________transactions?

Write only

3-the right order to perform data analytics is ?

Query and reporting .data monitoring,data visualization

4-one of the following is true about data sparsity?

Help determines the need for an index

Advantage for the table index

Problem in the table index

Related to the data allocation

5-social engineering is considered ?

Vulnerability

6-one of the following sql code is faster when executed?

Age >22

Age between 10 and 30

Age = 17

Age in (select age from B)

Age =min(x)

7-q= “update student set age =10” can be ?

Action

Data base query

Transaction

All correct

8-a transaction may consist of the following ?

All correct

Update statements

Insert statements

Select statements

9-________handles all submitted requests to the server ?

User process

10-the I/o cost to find the 20 “excellent”student (avg>90)from table size 200 using scan is ?

40

11-the data in data warehouse is?

Volatile and read only database

None volatile and read only database

Volatile and write only database

None volatile and write only database

12-the best solutions for distributed DB design is ?

Data fragmentation , data cleaning , data allocation

1-One of the following can be used to support routine decision making for business:

Answer : OLAP

2-the ETL operation performed for external and internal data in order to perform:

Answer : all correct (business continuity , business decision making and competitive advantage).

3- DBMS performance tuning performed on the _____ and sql performance tuning performed on the _______ side .

Answer : client , client .

4-_______ occure if transaction T3 calculated a summary function for data while another transaction t1 was updating the same data

Answer: inconsistent retrieval

5- _____ means when a transaction finished the changes and committed , they cannot be lost in the system failure .

Answer: durability .

6-The person who is independent from the DBMS in the organization with more focus on the management side :

Answer : DA

7- the optimizer will select the best plan for execution during the _____ phase

answer: parsing

8- one of the following levels is not applicable of data and process distribution .

Answer : single site process , multiple site data .

9- the organization can have multiple database administrators at the same time based on the :

Answer: number of DBA

10- q= '' select * from E5 where emp_dob = ' 01-jan-1990''' is ____ transparency

Answer : location

11-keep tracking the number of complains by customers about a product at a time is the responsibility of :

Answer: data monitoring .

12- one of the following is the best choice to be an index for an employee table :

Answer : all correct (address,gender,name)

13- protection data from any type of unauthorized access is more related to ______ :

Answer : security .

14- one of the following is true about '' optimized query '' and ''original query '':

Answer : both have same results but the optimized query more efficiency .

15- one of the following transaction sequence of execution represent concurrency in the system at sequence of time (1,2,3,4,5) :

Answer:T1,T1,T1,T2,T2

16-the internal and external data is stored in the data warehouse ______ performing ETL, operations : answer : after

17- one of the following is true about ? the sales of iphones in January in amman city was 1200 usd?

Answer : city is dimension

18- the data stored in the data warehouse (e.g payments) are ______ - oriented :

Answer: subject

19- _____ DDBMSs integrate di ? erent types of DBMSs over the network that support the same data model :

Answer: heterogeneous .

20- one of the following concurrency control solutions is more suitable for simple system with few update requests :

Answer: optimistic

21 – using ______ is the most important action to improve the database performance :

Answer: all correct (index in the table , best query reformulation ,best optimizer choice ,id access plan)

22- summing the '' weekly sales of iphones in amman city into one month sales = 1200 usd '' is an _________ OLAP operations

Answer: ROLL UP

33- the _______ is the software component found in each computer that store or retrieve data located at that location

Answer: Data processor (DP)

بسم اللهّٰ الرحمٰن الرحيم

DB.2

** SHROUQ MAHMOUD LAMI **

Ch.10

•••••••••••••••••••••••••••••

Transaction Management and Concurrency Control

Transaction: more than Query

To understand the concept of a transaction, suppose that you sell a product to a customer. Furthermore, suppose that the customer may charge the purchase to his or her account. Given that scenario, your sales transaction consists of at least the following parts:

1• You must write a new customer invoice.

2 • You must reduce the quantity on hand in the product's inventory

3 • You must update the account transactions.

4 • You must update the customer balance.

a transaction is any action that reads from or writes to a database.

A transaction may consist of the following:

1 • A simple SELECT statement to generate a list of table contents.

2-DELETE

3 • A series of related UPDATE statements to change the values of attributes in various tables.

4• A series of INSERT statements to add rows to one or more tables.

5• A combination of SELECT, UPDATE, and INSERT statements.

transaction is a logical unit of work that must be entirely completed or entirely aborted A sequence of database requests that accesses the database. A transaction is a logical unit of work; that is, it must be entirely completed or aborted— no intermediate ending states are accepted. All transactions must have the properties of atomicity, consistency, isolation, and durability.

consistent database state is one in which all data integrity constraints are satisfied A database state in which all data integrity constraints are satisfied

database request The equivalent of a single SQL statement in an application program or a transaction.

The SQL statements that represent this transaction :ملاحظة

تعني أكثر من جملة تحقق معاملة واحدة

10-1b Transaction Properties....الخصائص....

1• Atomicity requires that all operations (SQL requests) of a transaction be completed; if not, the transaction is aborted. If a transaction T1 has four SQL requests, all four requests must be successfully completed; otherwise, the entire transaction is aborted. In other words, a transaction is treated as a single, indivisible, logical unit of work. atomicity The transaction property that requires all parts of a transaction to be treated as a single, indivisible, logical unit of work. All parts of a transaction must be completed or the entire transaction is aborted.

2• Consistency A database condition in which all data integrity constraints are satisfied. To ensure consistency of a database, every transaction must begin with the database in a known consistent state. If not, the transaction will yield an inconsistent database that violates its integrity and business rules.

indicates the permanence of the database's consistent state. A transaction takes a database from one consistent state to another. When a transaction is completed, the database must be in a consistent state. If any of the transaction parts violates an integrity constraint, the entire transaction is aborted.

3• Isolation A database transaction property in which a data item used by one transaction is not available to other transactions until the first one ends.

means that the data used during the execution of a transaction cannot be used by a second transaction until the first one is completed. In other words, if transaction T1 is being executed and is using the data item X, that data item cannot be accessed by any other transaction (T2 … Tn) until T1 ends. This property is particularly useful in multiuser database environments because several users can access and update the database at the same time.

4• Durability ensures that once transaction changes are done and committed, they cannot be undone or lost, even in the event of a system failure.

The transaction property that ensures that once transaction changes are done and committed, they cannot be undone or lost, even in the event of a system failure

5•Serializability ensures that the schedule for the concurrent execution of the transactions yields consistent results.

A property in which the selected order of concurrent transaction operations creates the same final database state that would have been produced if the transactions had been executed in a serial fashion.

10-2 Concurrency Control

concurrency control Coordinating the simultaneous execution of transactions in a multiuser database system is known as concurrency control.

concurrency control A DBMS feature that coordinates the simultaneous execution of transactions in a multiprocessing database system while preserving data integrity

Concurrency control is important because the simultaneous execution of transactions over a shared database can create several data integrity and consistency problems.

lost update A concurrency control problem in which a data update is lost during the concurrent execution of transactions.

Inconsistent retrievals occur when a transaction accesses data before and after one or more other transactions finish working with such data inconsistent retrievals A concurrency control problem that arises when a transaction-calculating summary (aggregate) functions over a set of data while other transactions are updating the data, yielding erroneous results.

(490,491,492,493,494من أهم الجداول الرجوع للصحات)

TABLE 10.2 TWO CONCURRENT TRANSACTIONS TO UPDATE QOH

TABLE 10.3 SERIAL EXECUTION OF TWO TRANSACTIONS

TABLE 10.4 LOST UPDATES

TABLE 10.6 CORRECT EXECUTION OF TWO TRANSACTIONS

TABLE 10.7 AN UNCOMMITTED DATA PROBLEM

TABLE 10.8 RETRIEVAL DURING UPDATE

TABLE 10.9 TRANSACTION RESULTS: DATA ENTRY CORRECTION

TABLE 10.10 INCONSISTENT RETRIEVALS

10-2d The Scheduler

The scheduler is a special DBMS process that establishes the order in which the operations are executed within concurrent transactions.

scheduler The DBMS component that establishes the order in which concurrent transaction operations are executed. The scheduler interleaves the execution of database operations in a specific sequence to ensure serializability.

The scheduler interleaves the execution of database operations to ensure serializability and isolation of transactions.

serializable schedule In transaction management, a schedule of operations in which the interleaved execution of the transactions yields the same result as if they were executed in serial order.

10-3 Concurrency Control with Locking Methods:

Locking methods are one of the most common techniques used in concurrency control because they facilitate the isolation of data items used in concurrently executing transactions

A lock guarantees exclusive use of a data item to a current transaction

lock A device that guarantees unique use of a data item in a particular transaction operation. A transaction requires a lock prior to data access; the lock is released after the operation's execution to enable other transactions to lock the data item for their own use

lock manager A DBMS component that is responsible for assigning and releasing locks. مهم

lock manager OF A DBMS component

All lock information is handled by a lock manager, which is responsible for assigning and policing the locks used by the transactions.

pessimistic locking The use of locks based on the assumption that conflict between transactions is likely

10-3b Lock Types ... D.B لتحسين أداء

binary lock A lock that has only two states: locked (1) and unlocked (0). If a data item is locked by a transaction, no other transaction can use that data item تعريف البوكسات باللون الأخضر

A binary lock has only two states: locked (1) or unlocked (0). مهم شرح الدكتور

If an object such as a database, table, page, or row is locked by a transaction, no other transaction can use that object.

If an object is unlocked, any transaction can lock the object for its use. Every database operation requires that the affected object be locked. As a rule, a transaction must unlock the object after its termination. Therefore, every transaction requires a lock and unlock operation for each accessed data item. Such operations are automatically managed and scheduled by the DBMS; the user does not lock or unlock data items.

Row-level lock

A less restrictive database lock in which the DBMS allows concurrent transactions to access different rows of the same table, even when the rows are on the same page.

Field-level lock A lock that allows concurrent transactions to access the same row as long as they require the use of different fields (attributes) within that row. This type of lock yields the most flexible multiuser data access but requires a high level of computer overhead.

exclusive lock

An exclusive lock is issued when a transaction requests permission to update a data item and no locks are held on that data item by any other transaction. An exclusive lock does not allow other transactions to access the database.

shared lock

A lock that is issued when a transaction requests permission to read data from a database and no exclusive locks are held on the data by another transaction. A shared lock allows other readonly transactions to access the database.

mutual exclusive rule

A condition in which only one transaction at a time can own an exclusive lock on the same object.

ملاحظة.........................

Update,write(exclusive)

Read(Shared)

Shared/Exclusive An exclusive lock exists when access is reserved specifically for the transaction that locked the object.

shared locks allow several read transactions to read the same data item concurrently(ميزة)

A shared lock exists when concurrent transactions are granted read access on the basis of a common lock. A shared lock produces no conflict as long as all the concurrent transactions are read-only.

A shared lock is issued when a transaction wants to read data from the database and no exclusive lock is held on that data item.

An exclusive lock is issued when a transaction wants to update (write) a data item and no locks are currently held on that data item by any other transaction

Using the shared/exclusive locking concept, a lock can have three states: unlocked, shared (read), and exclusive (write)

ميزة /أيجابية

Advantage of shared locks

the use of shared locks renders data access more efficient,

Disadvantage of shared locks سيئتها

increases the lock manager's overhead

locks prevent serious data inconsistencies عمل المفاتيح بشكل عام / مهم

Although locks prevent serious data inconsistencies, they can lead to two major problems:

1• The resulting transaction schedule might not be serializable.

2• The schedule might create deadlocks

10-3c Two-Phase Locking to Ensure Serializability لضمان الترتيب:

Two-phase locking (2PL) defines how transactions acquire and relinquish locks. Two-phase locking guarantees serializability, but it does not prevent deadlocks. The two phases are:

1. A growing phase, in which a transaction acquires all required locks without unlocking any data. Once all locks have been acquired, the transaction is in its locked point.جمع الطلبات

2. A shrinking phase, in which a transaction releases all locks and cannot obtain a new lock.

The two-phase locking protocol is governed by the following rules:

1 • Two transactions cannot have conflicting locks.

2 • No unlock operation can precede a lock operation in the same transaction.

3 • No data is affected until all locks are obtained—that is, until the transaction is in its locked point.

depicts the two-phase locking protocol:

In this example, the transaction first acquires the two locks it needs.

When it has the two locks, it reaches its locked point. Next, the data is modified to conform to the transaction's requirements. Finally, the transaction is completed as it releases all of the locks it acquired in the first phase

Two-phase locking increases the transaction processing cost and might cause additional undesirable effects, such as deadlocks.

Deadlocks مهم

- 1A deadlock occurs when two transactions wait indefinitely for each other to unlock data.

- 2A condition in which two or more transactions wait indefinitely for the other to release the lock on a previously locked data item. Also called deadly embrace.

The three basic techniques to control deadlocks are: مهم

-1Deadlock prevention.

. Deadlock prevention works because it avoids the conditions that lead to deadlocking

-2Deadlock detection.

(rolled back and restarted)

-3 Deadlock avoidance

10-4 Concurrency Control with Time Stamping Methods: مهم جدا

The time stamping approach to scheduling concurrent transactions assigns a global, unique time stamp to each transaction. The time stamp value produces an explicit order in which transactions are submitted to the DBMS. Time stamps must have two properties: uniqueness and monotonicity

time stamping In transaction management, a technique used in scheduling concurrent transactions that assigns a global unique time stamp to each transaction من البوكس

uniqueness In concurrency control, a property of time stamping that ensures no equal time stamp values can exist. من البوكس

monotonicity A quality that ensures that time stamp values always increase. (The time stamping approach to scheduling concurrent transactions assigns a global, unique time stamp to each transaction. The time stamp value produces an explicit order in which transactions are submitted to the DBMS.) من البوكس

The disadvantage of the time stamping approach is that each value stored in the database requires two additional time stamp fields: one for the last time the field was read and one for the last update. Time stamping thus increases memory needs and the database's processing overhead. Time stamping demands a lot of system resources because many transactions might have to be stopped, rescheduled, and restamped. مهم

10-4a Wait/Die and Wound/Wait Schemes/ فهم الجدول بصفحة503 / Table 10.14 WAIT/DIE AND WOUND/WAIT CONCURRENCY CONTROL SCHEMES مهم جدا

wait/die A concurrency control scheme in which an older transaction must wait for the younger transaction to complete and release the locks before requesting the locks itself. Otherwise, the newer transaction dies and is rescheduled.

wound/wait A concurrency control scheme in which an older transaction can request the lock, preempt the younger transaction, and reschedule it. Otherwise, the newer transaction waits until the older transaction finishes.

optimistic approach In transaction management, a concurrency control technique based on the assumption that most database operations do not conflict. من البوكس

مهم جدا 10-5 Concurrency Control with Optimistic Methods طريقة بسيطة دون قيود على المعاملات

The optimistic approach is based on the assumption that the majority of database operations do not conflict.

The optimistic approach requires neither locking nor time stamping techniques. Instead, a transaction is executed without restrictions until it isملاحظة : هذة الطريقة متفائلة على التساهيل لا تتطلب

locking ولا time stamping

Using an optimistic approach, each transaction moves through two or three phases, referred to as read, validation, and write. 3

1• During the read phase, the transaction reads the database, executes the needed computations, and makes the updates to a private copy of the database values قراءة

2• During the validation phase, the transaction is validated to ensure that the changes made will not affect the integrity and consistency of the database. تثبيت

3• During the write phase, the changes are permanently applied to the database. اضافة

............ملاحظة

The optimistic approach is acceptable for most read or query database systems that require few update transactions.هذة الطريقة مناسبة للمعاملات التي أغلبها قراءة

10-7 Database Recovery Management

Database recovery restores a database from a given state (usually inconsistent) to a previously consistent state.

database recovery The process of restoring a database to a previous consistent state من البوكس

Recovery techniques are based on the atomic transaction property: all portions of the transaction must be treated as a single, logical unit of work in which all operations are applied and completed to produce a consistent database. If a transaction operation cannot be completed for some reason, the transaction must be aborted and any changes to the database must be rolled back (undone).

atomic transaction property A property that requires all parts of a transaction to be treated as a single, logical unit of work in which all operations must be completed (committed) to produce a consistent database. من البوكس

write ahead log protocol In concurrency control, a process that ensures transaction logs are written to permanent storage before any database data is actually updated. Also called a write-ahead protocol. من البوكس

- The write-ahead-log protocol ensures that transaction logs are always written before any database data is actually updated.

This protocol ensures that, in case of a failure, the database can later be recovered to a consistent state using the data in the transaction log.

redundant transaction logs Multiple copies of the transaction log kept by database management systems to ensure that the physical failure of a disk will not impair the DBMS's ability to recover data.

- Redundant transaction logs (several copies of the transaction log) ensure that a physical disk failure will not impair the DBMS's ability to recover data.

This protocol ensures that, in case of a failure, the database can later be recovered to a consistent state using the data in the transaction log

- Database buffers are temporary storage areas in primary memory used to speed up disk operations

buffer Temporary storage area in primary memory used to speed up disk operations.

- Database checkpoints are operations in which the DBMS writes all of its updated buffers in memory (also known as dirty buffers) to disk

When the recovery procedure uses a deferred-write technique (also called a deferred update), the transaction operations do not immediately update the physical database. Instead, only the transaction log is updated.

. The recovery process for all started and committed transactions (before the failure) follows these steps:

1. Identify the last checkpoint in the transaction log. This is the last time transaction data was physically saved to disk.

2. For a transaction that started and was committed before the last checkpoint, nothing needs to be done because the data is already saved.

3. For a transaction that performed a commit operation after the last checkpoint, the DBMS uses the transaction log records to redo the transaction and update the database, using the "after" values in the transaction log

4. For any transaction that had a ROLLBACK operation after the last checkpoint or that was left active (with neither a COMMIT nor a ROLLBACK) before the failure occurred, nothing needs to be done because the database was never updated

checkpoint In transaction management, an operation in which the database management system writes all of its updated buffers to disk.

deferred write technique See deferred update.

deferred update In transaction management, a condition in which transaction operations do not immediately update a physical database. Also called deferred write technique.

write-through technique In concurrency control, a process that ensures a database is immediately updated by operations during the transaction's execution, even before the transaction reaches its commit point. Also called immediate update.

immediate update See write-through technique.

The recovery process follows these steps:

1. Identify the last checkpoint in the transaction log. This is the last time transaction data was physically saved to disk.

2. For a transaction that started and was committed before the last checkpoint, nothing needs to be done because the data is already saved.

3. For a transaction that was committed after the last checkpoint, the DBMS re-does the transaction, using the "after" values of the transaction log. Changes are applied in ascending order, from oldest to newest.

4. For any transaction that had a ROLLBACK operation after the last checkpoint or that was left active (with neither a COMMIT nor a ROLLBACK) before the failure occurred, the DBMS uses the transaction log records to ROLLBACK or undo the operations, using the "before" values in the transaction log. Changes are applied in reverse order, from newest to oldest.

Ch.11

One of the main functions of a database system is to provide timely answers to end users. End users interact with the DBMS through the use of queries to generate information, using the following sequence: 1. The end-user (client-end) application generates a query.

2. The query is sent to the DBMS (server end).

3. The DBMS (server end) executes the query.

4. The DBMS sends the resulting data set to the end-user (client-end) application

Database performance tuning refers to a set of activities and procedures designed to reduce the response time of the database system—that is, to ensure that an end-user query is processed by the DBMS in the minimum amount of time. الهدف منها

database performance tuning A set of activities and procedures designed to reduce the response time of a database system—that is, to ensure that an enduser query is processed by the DBMS in the minimum amount of time. من البوكس

Fine-tuning the performance of a system requires a holistic approach. That is, all factors must be checked to ensure that each one operates at its optimum level and has sufficient resources to minimize the occurrence of bottlenecks.

TABLE 11.1
GENERAL GUIDELINES FOR BETTER SYSTEM PERFORMANCE

	SYSTEM RESOURCES	CLIENT	SERVER
Hardware	CPU	The fastest possible Dual-core CPU or higher	The fastest possible Multiple processors (quad-core technology) Cluster of networked computers
	RAM	The maximum possible to avoid OS memory to disk swapping	The maximum possible to avoid OS memory to disk swapping
	Hard disk	Fast SATA/EIDE hard disk with sufficient free hard disk space Solid State Drives (SSD) for faster speed	Multiple high-speed, high-capacity disks Fast disk interface (SAS / SCSI / Firewire / Fibre Channel RAID configuration optimized for throughput Solid State Drives (SSD) for faster speed Separate disks for OS, DBMS, and data spaces
	Network	High-speed connection	High-speed connection
Software	Operating System (OS)	64-bit OS for larger address spaces Fine-tuned for best client application performance	64-bit OS for larger address spaces Fine-tuned for best server application performance
	Network	Fine-tuned for best throughput	Fine-tuned for best throughput
	Application	Optimize SQL in client application	Optimize DBMS server for best performance

In general, database performance-tuning activities can be divided into those on the client side and those on the server side.

- On the client side, the objective is to generate a SQL query that returns the correct answer in the least amount of time, using the minimum amount of resources at the server end. The activities required to achieve that goal are commonly referred to as SQL performance tuning.

- On the server side, the DBMS environment must be properly configured to respond to clients' requests in the fastest way possible, while making optimum use of existing resources. The activities required to achieve that goal are commonly referred to as DBMS performance tuning

DBMS Architecture

- All data in a database is stored in data files. مهم

data file A named physical storage space that stores a database's data. It can reside in a different directory on a hard disk or on one or more hard disks. All data in a database is stored in data files. A typical enterprise database is normally composed of several data files. A data file can contain rows from one or more tables

518 مهمة جدا صفحة 11.1ملاحظة أرجع للكتاب لفهم الرسمة

- Data files are generally grouped in file groups or table spaces. A table space or file group is a logical grouping of several data files that store data with similar characteristics.

data cache A shared, reserved memory area that stores the most recently accessed data blocks in RAM. Also called buffer cache.مهم من البوكس

The data cache, or buffer cache, is a shared, reserved memory area that stores the most recently accessed data blocks in RAM. The data read from the data files is stored in the data cache after the data has been read or before the data is written to the data files. The data cache also caches system catalog data and the contents of the indexes

. • The SQL cache, or procedure cache, is a shared, reserved memory area that stores the most recently executed SQL statements or PL/SQL procedures, including triggers and functions.

- To move data from permanent storage (data files) to RAM (data cache), the DBMS issues I/O requests and waits for the replies. An input/output (I/O) request is a lowlevel data access operation that reads or writes data to and from computer devices, such as memory, hard disks, video, and printers

Working with data in the data cache is many times faster than working with data in the data files because the DBMS does not have to wait for the hard disk to retrieve the data; no hard disk I/O operations are needed to work within the data cache.

• Listener. The listener process listens for clients' requests and handles the processing of the SQL requests to other DBMS processes.

Once a request is received, the listener passes the request to the appropriate user process.

User. The DBMS creates a user process to manage each client session. Therefore, when you log on to the DBMS, you are assigned a user process.

This process handles all requests you submit to the server. There are many user processes—at least one per logged-in client.

• Scheduler. The scheduler process organizes the concurrent execution of SQL requests.

Lock manager. This process manages all locks placed on database objects, including disk pages.

Optimizer. The optimizer process analyzes SQL queries and finds the most efficient way to access the data.

Most of the algorithms proposed for query optimization are based on two principles:

1• The selection of the optimum execution order to achieve the fastest execution time

2• The selection of sites to be accessed to minimize communication costs

Within those two principles, a query optimization algorithm can be evaluated on the basis of its operation mode or the timing of its optimization.

Automatic query optimization means that the DBMS finds the most cost-effective access path without user intervention.

automatic query optimization A method by which a DBMS finds the most efficient access path for the execution of a query من البوكس

manual query optimization An operation mode that requires the end user or programmer to define the access path for the execution of a query. من البوكس

Manual query optimization requires that the optimization be selected and scheduled by the end user or programmer.Automatic query optimization is clearly more desirable from the end user's point of view, but the cost of such convenience is the increased overhead that it imposes on the DBMS.

• Static query optimization takes place at compilation time. In other words, the best optimization strategy is selected when the query is compiled by the DBMS.

static query optimization A query optimization mode in which the access path to a database is predetermined at compilation time من البوكس

• Dynamic query optimization takes place at execution time. Database access strategy is defined when the program is executed

ملاحظة...........

Finally, query optimization techniques can be classified according to the type of information that is used to optimize the query

dynamic query optimization The process of determining the SQL access strategy at run time, using the most up-to-date information about the database. من البوكس

statistically based query optimization algorithm A query optimization technique that uses statistical information about a database. The DBMS then uses these statistics to determine the best access strategy

A statistically based query optimization algorithm uses statistical information about the database.

The statistics provide information about database characteristics such as size, number of records, average access time, number of requests serviced, and number of users with access rights.

• The statistical information is managed by the DBMS and is generated in one of two different modes: dynamic or manual.

In the dynamic statistical generation mode, the DBMS automatically evaluates and updates the statistics after each data access operation.

In the manual statistical generation mode, the statistics must be updated periodically through a user-selected utility such as IBM's RUNSTAT command, which is used by DB2 DBMSs.

• A rule-based query optimization algorithm is based on a set of user-defined rules to determine the best query access strategy. The rules are entered by the end user or database administrator, and they are typically general in nature.

database statistics refers to a number of measurements about database objects, such as number of processors used, processor speed, and temporary space available.

A sample of measurements that the DBMS may gather about various database objects is shown in Table 11.2.

TABLE 11.2

SAMPLE DATABASE STATISTICS MEASUREMENTS

DATABASE OBJECT	SAMPLE MEASUREMENTS
Tables	Number of rows, number of disk blocks used, row length, number of columns in each row, number of distinct values in each column, maximum value in each column, minimum value in each column, and columns that have indexes
Indexes Key ,pointer يتكون من	Number and name of columns in the index key, number of key values in the index, number of distinct key values in the index key, histogram of key values in an index, and number of disk pages used by the index
Environment Resources	Logical and physical disk block size, location and size of data files, and number of extends per data file

ملاحظة أرجع الى الرسمة صفحة523

FIGURE 11.2 QUERY PROCESSING

11-2 Query Processing

What happens at the DBMS server end when the client's SQL statement is received? In simple terms, the DBMS processes a query in three phases: مقسمين على ثلاث مراحل

1. Parsing. The DBMS parses the SQL query and chooses the most efficient access/ execution plan.

2. Execution. The DBMS executes the SQL query using the chosen execution plan.

3. Fetching. The DBMS fetches the data and sends the result set back to the client

FIGURE 11.2 QUERY PROCESSING فهم الرسمة الرجوع الى صفحة 523 مهم

11-2a SQL Parsing Phase

The optimization process includes breaking down—parsing—the query into smaller units and transforming the original SQL query into a slightly different version of the original SQL code, but one that is fully equivalent and more efficient. Fully equivalent means that the optimized query results are always the same as the original query

More efficient means that the optimized query will almost always execute faster than the original query.

query optimizer A DBMS process that analyzes SQL queries and finds the most efficient way to access the data. The query optimizer generates the access or execution plan for the query.

TABLE 11.3

SAMPLE DBMS ACCESS PLAN I/O OPERATIONS مهم

الطرق حفظ وفهم

OPERATION	OPERATION
Table scan (full)	Reads the entire table sequentially, from the first row to the last, one row at a time (slowest)
Table access (row ID)	Reads a table row directly, using the row ID value (fastest)

Index scan (range)	Reads the index first to obtain the row IDs and then accesses the table rows directly (faster than a full table scan)
Index access (unique)	Used when a table has a unique index in a column
Nested loop	Reads and compares a set of values to another set of values, using a nested loop style (slow)
Merge	Merges two data sets (slow)
Sort	Sorts a data set (slow)

access plan A set of instructions generated at application compilation time that is created and managed by a DBMS. The access plan predetermines how an application's query will access the database at run time. مهم جدا

query processing bottleneck In query optimization, a delay introduced in the processing of an I/O operation that causes the overall system to slow down.

11-3 Indexes and Query Optimization

Indexes are crucial in speeding up data access because they facilitate searching, sorting, and using aggregate functions and even join operations. The improvement in data access speed occurs because an index is an ordered set of values that contains the index key and pointers. The pointers are the row IDs for the actual table rows. Conceptually, a data index is similar to a book index. When you use a book index, you look up a word, which is similar to the index key. The word is accompanied by one or more page numbers where the word is used; these numbers are similar to pointers.

An index scan is more efficient than a full table scan because the index data is preordered and the amount of data is usually much smaller. Therefore, when performing searches, it is almost always better for the DBMS to use the index to access a table than to scan all rows in a table sequentially.

FIGURE 11.3 INDEX REPRESENTATION FOR THE CUSTOMER TABLE لفهم الشكل الرجوع الى صفحة 526

مهم جدا جدا /فهم صفحة 527...

If indexes are so important, why not index every column in every table? The simple answer is that it is not practical to do so. Indexing every column in every table overtaxes the DBMS in terms of indexmaintenance processing, especially if the table has many attributes and rows, or requires many inserts, updates, and deletes.

Data sparsity refers to the number of different values a column could have.

Using the preceding index characteristics, a database designer can determine the best type of index to use. For example, assume that a CUSTOMER table has several thousand rows.

Because the CUS_LNAME column contains many different values that repeat a relatively small number of times compared to the total number of rows in the table, a B-tree index will be used.

11-4 Optimizer Choices

- A rule-based optimizer uses preset rules and points to determine the best approach to execute a query.

rule-based optimizer A query optimization mode based on the rule-based query optimization algorithm.

A cost-based optimizer uses sophisticated algorithms based on statistics about the objects being accessed to determine the best approach to execute a query

cost-based optimizer A query optimization mode that uses an algorithm based on statistics about the objects being accessed, including number of rows, indexes available, index sparsity, and so on.

مهم جدا جدا فهم الجدول الرجوع الى صفحة529

TABLE 11.4

COMPARING ACCESS PLANS AND I/O COSTS

11-5 SQL Performance Tuning

SQL performance tuning is evaluated from the client perspective. Therefore, the goal is to illustrate some common practices used to write efficient SQL code.

11-5a Index Selectivity

Indexes are the most important technique used in SQL performance optimization.

The key is to know when an index is used. As a general rule, indexes are likely to be used:

1• When an indexed column appears by itself in the search criteria of a WHERE or HAVING clause

2• When an indexed column appears by itself in a GROUP BY or ORDER BY clause

3• When a MAX or MIN function is applied to an indexed column

4• When the data sparsity on the indexed column is high

index selectivity A measure of how likely an index is to be used in query processing.

Index selectivity is a measure of the likelihood that an index will be used in query processing. Here are some general guidelines for creating and using indexes:

1• Create indexes for each single attribute used in a WHERE, HAVING, ORDER BY, or GROUP BY clause.

2• Do not use indexes in small tables or tables with low sparsity.

3• Declare primary and foreign keys so the optimizer can use the indexes in join operations.

4• Declare indexes in join columns other than PK or FK. If you perform join operations on columns other than the primary and foreign keys, you might be better off declaring indexes in those columns.

TABLE 11.6 الرجوع الى صفحة 533 CONDITIONAL CRITERIA

Most of the query optimization techniques mentioned below are designed to make the optimizer’s work easier.

The following common practices are used to write efficient conditional expressions in SQL code.) مهم/ فهم أيضا الرجوع للصفحات رقم 533/534 (

1• Use simple columns or literals as operands in a conditional expression—avoid the use of conditional expressions with functions whenever possible

2• Numeric field comparisons are faster than character, date, and NULL comparisons.

3• Equality comparisons are generally faster than inequality comparisons. If there are no exact matches, the condition is evaluated as false. However, if you use an inequality symbol (>, >=, <=), the DBMS must perform additional processing to complete the request, because there will almost always be more “greater than” or “less than” values in the index than “equal” values.

4• Whenever possible, transform conditional expressions to use literals. For example, if your condition is P_PRICE − 10 = 7, change it to read P_PRICE = 17. Also, if you have a composite condition such as: P_QOH < P_MIN AND P_MIN = P_REORDER AND P_QOH = 10 change it to read: P_QOH = 10 AND P_MIN = P_REORDER AND P_MIN > 10

5• When using multiple conditional expressions, write the equality conditions first.

6• If you use multiple AND conditions, write the condition most likely to be false first.

7• When using multiple OR conditions, put the condition most likely to be true first.

8• Whenever possible, try to avoid the use of the NOT logical operator

Ch.12

…………………………………

12-1 The Evolution of Distributed Database Management Systems :

distributed database management system (DDBMS) A DBMS that supports a database distributed across several different sites; a DDBMS governs the storage and processing of logically related data over interconnected computer systems in which both data and processing functions are distributed among several sites.

A distributed database management system (DDBMS) governs the storage and processing of logically related data over interconnected computer systems in which both data and processing are distributed among several sites.

To understand how and why the DDBMS is different from the DBMS, it is useful to briefly examine the changes in the business environment that set the stage for the development of the DDBMS.

النقاط التي جعلوا الحاجة الى distributed database management system/ مهم

The last two decades gave birth to a series of crucial social and technological changes that affected the nature of the systems and the data they use:

1• Business operations became global; with this change, competition expanded from the shop on the next corner to the web store in cyberspace.

2• Customer demands and market needs favored an on-demand transaction style, mostly based on web-based services.

3• Rapid social and technological changes fueled by low-cost, smart mobile devices increased the demand for complex and fast networks to interconnect them

4• Data realms are converging in the digital world more frequently. As a result, applications must manage multiple types of data, such as voice, video, music, and images.

الشكل مطلوب صفحة554

FIGURE 12.1 CENTRALIZED DATABASE MANAGEMENT SYSTEM

The distributed database is especially desirable because centralized database management is subject to problems such as:

1• Performance degradation because of a growing number of remote locations over greater distances. الأداء قل

2• High costs associated with maintaining and operating large central (mainframe) database systems and physical infrastructure. تكلفة عالية

3• Reliability problems created by dependence on a central site (single point of failure syndrome) and the need for data replication. الجهاز اذا تعطل يتعطل الشغل كاملا

4• Scalability problems associated with the physical limits imposed by a single location, such as physical space, temperature conditioning, and power consumption. عدم قدرة التعامل مع عدد كبير من المعاملات /الناس

5• Organizational rigidity imposed by the database, which means it might not support the flexibility and agility required by modern global organizations.

ملاحظة ....

CENTRALIZED :

غير كافي بسبب المشاكل السابقة / أيضا اذا عكسنا تلك المشاكل بتصير ميزات

Distributed Database

12-3 Distributed Processing and Distributed Databases

In distributed processing, a database's logical processing is shared among two or more physically independent sites that are connected through a network. For example, the data input/output (I/O), data selection, and data validation might be performed on one computer, and a report based on that data might be created on another computer.

عبارة عن توزيع وتكون على محورين(processing, database)

الإحتمالات كالآتي :

1) Distributed الإثنان ليس

2) البروسيس هو Distributed والداتا بيس لأ

3) الداتا Distributed و process لأ

4) كلاهما Distributed وهو أفضل إحتمال

distributed processing Sharing the logical processing of a database over two or more sites connected by a network.

distributed database A logically related database that is stored in two or more physically independent sites.

TABLE 12.1

DISTRIBUTED DBMS ADVANTAGES AND DISADVANTAGES

ADVANTAGES	DISADVANTAGES
Data is located near the site of greatest demand. The data in a distributed database system is dispersed to match business requirements.	Complexity of management and control. Applications must recognize data location, and they must be able to stitch together data from various sites. Database administrators must have the ability to coordinate database activities to prevent database degradation due to data anomalies.
Faster data access. End users often work with only the nearest stored subset of the data.	Technological difficulty. Data integrity, transaction management, concurrency control, security, backup, recovery, and query optimization must all be addressed and resolved.
Faster data processing. A distributed database system spreads out the system's workload by processing data at several sites.	Security. The probability of security lapses increases when data is located at multiple sites. The responsibility of data management will be shared by different people at several sites.
Growth facilitation. New sites can be added to the network without affecting the operations of other sites	Lack of standards. There are no standard communication protocols at the database level. For example, different database vendors employ different and often incompatible techniques to manage the distribution of data and processing in a DDBMS environment.
Improved communications. Because local sites are smaller and located closer to customers, local sites foster better communication among departments	Increased storage and infrastructure requirements. Multiple copies of data are required at different sites, thus requiring additional storage space.

and between customers and company staff	
Reduced operating costs. It is more cost-effective to add nodes to a network than to update a mainframe system. Development work is done more cheaply and quickly on low-cost PCs than on mainframes.	Increased training cost. Training costs are generally higher in a distributed model than they would be in a centralized model, sometimes even to the extent of offsetting operational and hardware savings.
User-friendly interface. PCs and workstations are usually equipped with an easy-to-use graphical user interface (GUI). The GUI simplifies training and use for end users.	Costs. Distributed databases require duplicated infrastructure to operate, such as physical location, environment, personnel, software, and licensing.
Less danger of a single-point failure. When one of the computers fails, the workload is picked up by other workstations. Data is also distributed at multiple sites.	
Processor independence. The end user can access any available copy of the data, and an end user's request is processed by any processor at the data location.	

. In a distributed database system, a database is composed of several parts known as database fragments. يمكن تقسيمها لعدة أجزاء

The database fragments are located at different sites and can be replicated among various sites.

امكانية نسخها

database fragment

A subset of a distributed database. Although the fragments may be stored at different sites within a computer network, the set of all fragments is treated as a single database. See also horizontal fragmentation and vertical fragmentation.

مهم

FIGURE 12.2 DISTRIBUTED PROCESSING ENVIRONMENT

FIGURE 12.3 DISTRIBUTED DATABASE ENVIRONMENT

FIGURE 12.4 A FULLY DISTRIBUTED DATABASE MANAGEMENT SYSTEM

FULLY أغلب الأنظمة الكبيرة تستخدم

الرجوع صفحة / الرسمات وفهم لتوضيح558 وصفحة 560

12-5 DDBMS Components

The DDBMS must include at least the following components:

1• Computer workstations or remote devices (sites or nodes) that form the network system. The distributed database system must be independent of the computer system hardware.

2• Network hardware and software components that reside in each workstation or device. The network components allow all sites to interact and exchange data. Because the components—computers, operating systems, network hardware, and so on—are likely to be supplied by different vendors, it is best to ensure that distributed database functions can be run on multiple platforms.

3• Communications media that carry the data from one node to another. The DDBMS must be communications media-independent; that is, it must be able to support several types of communications media.

4• The transaction processor (TP) is the software component found in each computer or device that requests data. The transaction processor receives and processes the application's remote and local data requests. The TP is also known as the application processor (AP) or the transaction manager (TM).

5• The data processor (DP) is the software component residing on each computer or device that stores and retrieves data located at the site. The DP is also known as the data manager (DM). A data processor may even be a centralized DBMS.

transaction processor (TP) In a DDBMS, the software component on each computer that requests data. The TP is responsible for the execution and coordination of all database requests issued by a local application that accesses data on any DP. Also called transaction manager (TM) or application processor (AP).

application processor (AP) See transaction processor (TP).

transaction manager (TM) See transaction processor (TP).

data processor (DP) The resident software component that stores and retrieves data through a DDBMS. The DP is responsible for managing the local data in the computer and coordinating access to that data. Also known as data manager (DM).

data manager (DM) See data processor (DP).

FIGURE 12.5 DISTRIBUTED DATABASE SYSTEM COMPONENTS

صفحة561

12-6 Levels of Data and Process Distribution

1 Single-Site Processing, Single-Site Data (SPSD)

2 Multiple-Site Processing, Single-Site Data (MPSD)

3 Multiple-Site Processing, Multiple-Site Data (MPMD)

single-site processing, single-site data (SPSD) A scenario in which all processing is done on a single host computer and all data is stored on the host computer’s local disk.

multiple-site processing, singlesite data (MPSD) A scenario in which multiple processes run on different computers sharing a single data repository multiple-site processing, multiple site data (MPMD) A scenario describing a fully distributed database management system with support for multiple data processors and transaction processors at multiple sites

TABLE 12.2

DATABASE SYSTEMS: LEVELS OF DATA AND PROCESS DISTRIBUTION

ADVANTAGES	SINGLE-SITE DATA	MULTIPLE-SITE DATA
Single-site process	Host DBMS	Not applicable (Requires multiple processes)
Multiple-site process	File server Client/server DBMS (LAN DBMS)	Fully distributed Client/server DDBMS

12-6c Multiple-Site Processing, Multiple-Site Data

The multiple-site processing, multiple-site data (MPMD) scenario describes a fully distributed DBMS with support for multiple data processors and transaction processors at multiple sites.

Depending on the level of support for various types of databases, DDBMSs are classified as either homogeneous or heterogeneous. Homogeneous DDBMSs integrate multiple instances of the same DBMS over a network—for example, multiple instances of Oracle 11g running on different platforms. In contrast, heterogeneous DDBMSs integrate different types of DBMSs over a network, but all support the same data model. For example, Table 12.3 lists several relational database systems that could be integrated within a DDBMS. A fully heterogeneous DDBMS will support different DBMSs, each one supporting a different data model, running under different computer systems.

12-7 Distributed Database Transparency Features

A distributed database system should provide some desirable transparency features that make all the system's complexities hidden to the end user.

In other words, the end user should have the sense of working with a centralized DBMS. For this reason, the minimum desirable DDBMS transparency features are:

- Distribution transparency allows a distributed database to be treated as a single logical database. If a DDBMS exhibits distribution transparency, the user does not need to know:

1 – The data is partitioned—meaning the table's rows and columns are split vertically or horizontally and stored among multiple sites. تقسيم

2 – The data is geographically dispersed among multiple sites. توزيع

3– The data is replicated among multiple sites. نسخ

fully heterogeneous distributed database system (fully heterogeneous DDBMS)

A system that integrates different types of database management systems (hierarchical, network, and relational) over a network. It supports different database management systems that may even support different data models running under different computer systems. See also heterogeneous DDBMS and homogeneous DDBMS.

distribution transparency A DDBMS feature that allows a distributed database to look like a single logical database to an end user.

- Transaction transparency allows a transaction to update data at more than one network site.

Transaction transparency ensures that the transaction will be either entirely completed or aborted, thus maintaining database integrity.

- Failure transparency ensures that the system will continue to operate in the event of a node or network failure

- Performance transparency allows the system to perform as if it were a centralized DBMS.

- Heterogeneity transparency allows the integration of several different local DBMSs (relational, network, and hierarchical) under a common, or global, schema.

transaction transparency A DDBMS property that ensures database transactions will maintain the distributed database's integrity and consistency, and that a transaction will be completed only when all database sites involved complete their part of the transaction.

failure transparency A feature that allows continuous operation of a DDBMS, even if a network node fails.

performance transparency A DDBMS feature that allows a system to perform as though it were a centralized DBMS.

heterogeneity transparency A feature that allows a system to integrate several centralized DBMSs into one logical DDBMS.

fragmentation transparency A DDBMS feature that allows a system to treat a distributed database as a single database even though it is divided into two or more fragments.

location transparency A property of a DDBMS in which database access requires the user to know only the name of the database fragments. (Fragment locations need not be known.

12-8 Distribution Transparency

Distribution transparency allows a physically dispersed database to be managed as though it were a centralized database.

The level of transparency supported by the DDBMS varies from system to system. Three levels of distribution transparency are recognized:

1• Fragmentation transparency is the highest level of distribution transparency. The end user or programmer does not need to know that a database is partitioned. Therefore, neither fragment names nor fragment locations are specified prior to data access.

2• Location transparency exists when the end user or programmer must specify the database fragment names but does not need to specify where those fragments are located.

3• Local mapping transparency exists when the end user or programmer must specify both the fragment names and their locations.

ملخص النقاط السابقة بالجدول التالي مهم / فهم وحفظ

TABLE 12.4

SUMMARY OF TRANSPARENCY FEATURES			
IF THE SQL STATEMENT REQUIRES:			
FRAGMENT NAME?	LOCATION NAME?	THEN THE DBMS SUPPORTS	LEVEL OF DISTRIBUTON TRANSPARENCY
Yes	Yes	Local mapping transparency	Low
Yes	No	Location transparency	Medium
No	No	Fragmentation transparency	High

الرسمة مطلوبة للفهم / صفحة 566 FIGURE 12.8 FRAGMENT LOCATIONS

local mapping transparency A property of a DDBMS in which database access requires the user to know both the name and location of the fragments.

unique fragment In a DDBMS, a condition in which each row is unique, regardless of which fragment it is located in.

12-9 Transaction Transparency

Transaction transparency is a DDBMS property that ensures database transactions will maintain the distributed database's integrity and consistency.

Transaction transparency ensures that the transaction will be completed only when all database sites involved in the transaction complete their part of the transaction.

Transaction Transparency

-1Distributed Requests

-2Distributed Transactions

-3A REMOTE REQUEST لفهمها الرجوع للرسمة صفحة 568

A remote request lets a single SQL statement access the data that are to be processed by a single remote database processor. مهم , the SQL statement (or request) can reference data at only one remote site.

remote request A DDBMS feature that allows a single SQL statement to access data in a single remote DP

a remote transaction, composed of several requests, accesses data at a single remote site. مهم

remote transaction A DDBMS feature that allows a transaction (formed by several requests) to access data in a single remote DP.

FIGURE 12.10 A REMOTE TRANSACTION

FIGURE 12.11 A DISTRIBUTED TRANSACTION

FIGURE 12.12 A DISTRIBUTED REQUEST

لفهم الاشكال والكود570 / 569الرجوع صفحة

distributed transaction A database transaction that accesses data in several remote data processors (DPs) in a distributed database.

distributed request A database request that allows a single SQL statement to access data in several remote data processors (DPs) in a distributed database.

12-10 Performance and Failure Transparency

One of the most important functions of a database is its ability to make data available. Web-based distributed data systems demand high availability, which means not only that data is accessible but that requests are processed in a timely manner.

Performance transparency allows a DDBMS to perform as if it were a centralized database. In other words, no performance degradation should be incurred due to data distribution.

The objective of query optimization is to minimize the total cost associated with the execution of a request. The costs associated with a request are a function of the following:

1• Access time (I/O) cost involved in accessing the data from multiple remote sites

2• Communication cost associated with data transmission among nodes in distributed database systems

3• CPU time cost associated with the processing overhead of managing distributed transactions

12-11 Distributed Database Design

Whether the database is centralized or distributed, the design principles and concepts described in Chapters 3, 4, and 6 are still applicable. However, the design of a distributed database introduces three new issues:

1• How to partition the database into fragments

2• Which fragments to replicate

3• Where to locate those fragments and replicas

Distributed Database Design

-1Data Fragmentation

-2Data Replication

-3Data Allocation

Data fragmentation allows you to break a single object into two or more segments, or fragments.

The object might be a user's database, a system database, or a table. Each fragment can be stored at any site over a computer network. Information about data fragmentation is stored in the distributed data catalog (DDC), from which it is accessed by the TP to process user requests.

data fragmentation A characteristic of a DDBMS that allows a single object to be broken into two or more segments or fragments. The object might be a user's database, a system database, or a table. Each fragment can be stored at any site on a computer network.

• Horizontal fragmentation refers to the division of a relation into subsets (fragments) of tuples (rows). Each fragment is stored at a different node, and each fragment has unique rows. However, the unique rows all have the same attributes (columns). In short, each fragment represents the equivalent of a SELECT statement, with the WHERE clause on a single attribute.

horizontal fragmentation The distributed database design process that breaks a table into subsets of unique rows.

• Vertical fragmentation refers to the division of a relation into attribute (column) subsets. Each subset (fragment) is stored at a different node, and each fragment has unique columns—with the exception of the key column, which is common to all fragments. This is the equivalent of the PROJECT statement in SQL.

vertical fragmentation In distributed database design, the process that breaks a table into a subset of columns from the original table. Fragments must share a common primary key.

- Mixed fragmentation refers to a combination of horizontal and vertical strategies. In other words, a table may be divided into several horizontal subsets (rows), each one having a subset of the attributes (columns).

mixed fragmentation A combination of horizontal and vertical strategies for data fragmentation, in which a table may be divided into several rows and each row has a subset of the attributes (columns).

576 /577صفحة

TABLE 12.5

HORIZONTAL FRAGMENTATION OF THE CUSTOMER TABLE BY STATE

الرجوع للسؤال فلوريدا /فهم ومهم جدا

12-11b Data Replication

Data replication refers to the storage of data copies at multiple sites served by a computer network. Fragment copies can be stored at several sites to serve specific information requirements. Because the existence of fragment copies can enhance data availability and response time, data copies can help to reduce communication and total query costs.

data replication The storage of duplicated database fragments at multiple sites on a DDBMS. Duplication of the fragments is transparent to the end user. Data replication provides fault tolerance and performance enhancements.

Replicated data is subject to the mutual consistency rule, which requires that all copies of data fragments be identical. Therefore, to maintain data consistency among the replicas, the DDBMS must ensure that a database update is performed at all sites where replicas exist.

mutual consistency rule A data replication rule that requires all copies of data fragments to be identical.

There are basically two styles of replication:

1• Push replication

2• Pull replication.

• Push replication. After a data update, the originating DP node sends the changes to the replica nodes to ensure that data is immediately updated. This type of replication focuses on maintaining data consistency. However, it decreases data availability due to the latency involved in ensuring data consistency at all nodes.

• Pull replication. After a data update, the originating DP node sends “messages” to the replica nodes to notify them of the update. The replica nodes decide when to apply the updates to their local fragment. In this type of replication, data updates propagate more slowly to the replicas. The focus is on maintaining data availability. However, this style of replication allows for temporary data inconsistencies. Although replication has some benefits, such as improved data availability, better load distribution, improved data failure tolerance, and reduced query costs, it also imposes additional DDBMS processing overhead because each data copy must be maintained by the system.فوائد / ايجابيات

To illustrate the replica overhead imposed on a DDBMS, consider the processes that the DDBMS must perform to use the database:

1• If the database is fragmented, the DDBMS must decompose a query into subqueries to access the appropriate fragments.

2• If the database is replicated, the DDBMS must decide which copy to access. A READ operation selects the nearest copy to satisfy the transaction. A WRITE operation requires that all copies be selected and updated to satisfy the mutual consistency rule.

3• The TP sends a data request to each selected DP for execution.

4• The DP receives and executes each request and sends the data back to the TP.

5• The TP assembles the DP responses.

The problem becomes more complex when you consider additional factors such as network topology and communication throughputs. Three replication scenarios exist: a database can be fully replicated, partially replicated, or unreplicated.

- A fully replicated database stores multiple copies of each database fragment at multiple sites. In this case, all database fragments are replicated. A fully replicated database can be impractical due to the amount of overhead it imposes on the system.
- A partially replicated database stores multiple copies of some database fragments at multiple sites. Most DDBMSs are able to handle the partially replicated database well.
- An unreplicated database stores each database fragment at a single site. Therefore, there are no duplicate database fragments.

fully replicated database In a DDBMS, the distributed database that stores multiple copies of each database fragment at multiple sites.

partially replicated database A distributed database in which copies of only some database fragments are stored at multiple sites.

unreplicated database A distributed database in which each database fragment is stored at a single site.

Several factors influence the decision to use data replication:

1• Database size.

2• Usage frequency

3• Costs.

12-11c Data Allocation

Data allocation describes the process of deciding where to locate data.

Data allocation strategies are as follows:

1• With centralized data allocation, the entire database is stored at one site.

2• With partitioned data allocation, the database is divided into two or more disjointed parts (fragments) and stored at two or more sites.

3• With replicated data allocation, copies of one or more database fragments are stored at several sites.

data allocation In a distributed DBMS, the process of deciding where to locate data fragments.

centralized data allocation A data allocation strategy in which the entire database is stored at one site. Also known as a centralized database.

partitioned data allocation A data allocation strategy of dividing a database into two or more fragments that are stored at two or more sites.

replicated data allocation A data allocation strategy in which copies of one or more database fragments are stored at several sites.

Ch.13

Business Intelligence and Data Warehouses الاستفادة منها

business intelligence provides

business decision support framework

business intelligence architecture

operational data and decision support data

13-1The Need for Data Analysis

1-Organizations tend to grow and prosper as they gain a better understanding of their environment. Most managers need to track daily transactions to evaluate how the business is performing. By tapping into the operational database, management can develop an understanding of how the company is performing and evaluate whether the current strategies meet organizational goals.

2- In addition, analyzing the company data can provide insightful information about short-term tactical evaluations and strategic questions, such as: Are our sales promotions working?

What market percentage are we controlling? Are we attracting new customers? Tactical and strategic decisions are also shaped by constant pressure from external and internal forces, including globalization, the cultural and legal environment, and technology.

3- Organizations are always looking for a competitive advantage through product development, market positioning, sales promotions, and customer service

13-2Business Intelligence

Business intelligence (BI)1 is a term that describes a comprehensive, cohesive, and integrated set of tools and processes used to capture, collect, integrate, store, and analyze data with the purpose of generating and presenting information to support business decision making. This intelligence is based on learning and understanding the facts about the business environment.

BI is a framework that allows a business to transform data into information, information into knowledge, and knowledge into wisdom.

BI has the potential to positively affect a company's culture by creating continuous business performance improvement through active decision support at all levels in an organization.

business intelligence (BI) A comprehensive, cohesive, and integrated set of tools and processes used to capture, collect, integrate, store, and analyze data with the purpose of generating and presenting information to support business decision making

BI is not a product by itself, but a framework of concepts, practices, tools, and technologies that help a business better understand its core capabilities, provide snapshots of the company situation, and identify key opportunities to create competitive advantage. In general, BI provides a framework for:

- Collecting and storing operational data
- Aggregating the operational data into decision support data
- Analyzing decision support data to generate information
- Presenting such information to the end user to support business decisions

FIGURE 13.1 BUSINESS INTELLIGENCE FRAMEWORK الرجوع لفهم الشكل صفحة592

TABLE 13.2

BASIC BI ARCHITECTURAL COMPONENTS

COMPONENT	DESCRIPTION
ETL tools	Data extraction, transformation, and loading (ETL) tools collect, filter, integrate, and aggregate internal and external data to be saved into a data store optimized for decision support. Internal data is generated by the company during its day-to-day operations, such as product sales history, invoicing, and payments. The external data sources provide data that cannot be found within the company but is relevant to the business, such as stock prices, market indicators, marketing information (such as demographics), and competitors' data. Such data is generally located in external databases provided by industry groups or companies that market the data.
Data store	The data store is optimized for decision support and is generally represented by a data warehouse or a data mart. The data is stored in structures that are optimized for data analysis and query speed.
Query and reporting	This component performs data selection and retrieval, and it is used by the data analyst to create queries that access the database and create the required reports. Depending on the implementation, the query and reporting tool accesses the operational database, or more commonly, the data store
Data visualization	This component presents data to the end user in a variety of meaningful and innovative ways. This tool helps the end user select the most appropriate presentation format, such as summary reports, maps, pie or bar graphs, mixed graphs, and static or interactive dashboards.
Data monitoring and alerting	This component allows real-time monitoring of business activities. The BI system will present the concise information in a single integrated view for the data analyst. This integrated view could include specific metrics about the system performance or activities, such as number of orders placed in the last four hours, number of customer complaints by product by month, and total revenue by region. Alerts can be placed on a given metric; once the value of a metric goes below or above a certain baseline, the system will perform a given action, such as emailing shop floor managers, presenting visual alerts, or starting an application.
Data analytics	This component performs data analysis and data-mining tasks using the data in the data store. This tool advises the user as to which data analysis tool to select and how to build a reliable business data model. Business models are generated by special algorithms that

	identify and enhance the understanding of business situations and problems. Data analysis can be either explanatory or predictive. Explanatory analysis uses the existing data in the data store to discover relationships and their types, and predictive analysis creates statistical models of the data that allow predictions of future values and events. Chapter 14, Big Data Analytics and NoSQL, covers these topics in more detail.

extraction, transformation, and loading (ETL) In a data warehousing environment, the integrated processes of getting data from original sources into the data warehouse. ETL includes retrieving data from original data sources (extraction), manipulating the data into an appropriate form (transformation), and storing the data in the data warehouse (loading).

TABLE 13.5

CONTRASTING OPERATIONAL AND DECISION SUPPORT DATA CHARACTERISTICS

CHARACTERISTIC	OPERATIONAL DATA	DECISION SUPPORT DATA
Data currency	Current operations Real-time data	Historic data Snapshot of company data Time component (week/month/year)
Granularity	Atomic-detailed data	Summarized data
Summarization level	Low; some aggregate yields	High; many aggregation levels
Data model	Highly normalized Mostly relational DBMSs	Non-normalized Complex structures Some relational, but mostly multidimensional DBMSs
Transaction type	Mostly updates	Mostly query
Transaction volumes	High-update volumes	Periodic loads and summary calculations
Transaction speed	Updates are critical	Retrievals are critical
Query activity	Low to medium	High
Query scope	Narrow range	Broad range
Query complexity	Simple to medium	Very complex
Data volumes	Hundreds of gigabytes	Terabytes to petabytes

13-3 The Data Warehouse

data warehouse An integrated, subjectoriented, time-variant, nonvolatile collection of data that provides support for decision making من البوكس

Bill Inmon, the acknowledged "father" of the data warehouse, defines the term as "an integrated, subject-oriented, time-variant, nonvolatile collection of data that provides support for decision making."2 (Italics were added for emphasis.)

To understand that definition, take a more detailed look at its components.

1• Integrated. The data warehouse is a centralized, consolidated database that integrates data derived from the entire organization and from multiple sources with diverse formats

2• Subject-oriented. Data warehouse data is arranged and optimized to provide answers to questions from diverse functional areas within a company. Data warehouse data is organized and summarized by topic, such as sales, marketing, finance, distribution, and transportation. For each topic, the data warehouse contains specific subjects of interest—products, customers, departments, regions, promotions, and so on. This form of data organization is quite different from the more functional or process-oriented organization of typical transaction systems.

3• Time-variant. In contrast to operational data, which focuses on current transactions, warehouse data represents the flow of data through time. The data warehouse can even contain projected data generated through statistical and other models. It is also time-variant in the sense that when data is periodically uploaded to the data warehouse, all time-dependent aggregations are recomputed.

4• Nonvolatile. Once data enters the data warehouse, it is never removed.

In summary, the data warehouse is a read-only database optimized for data analysis and query processing.

Table 13.8 summarizes the differences between data warehouses and operational databases.

Table 13.8

CHARACTERISTICS OF DATA WAREHOUSE DATA AND OPERATIONAL DATABASE DATA

CHARACTERISTIC	OPERATIONAL DATABASE DATA	DATA WAREHOUSE DATA
Integrated	Similar data can have different representations or meanings. For example, Social Security numbers may be stored as ###-##-#### or as #########, and a given condition may be labeled as T/F or 0/1 or Y/N. A sales value may be shown in thousands or in millions.	Provide a unified view of all data elements with a common definition and representation for all business units.
Subject-oriented	Data is stored with a functional, or process, orientation. For example, data may be stored for invoices, payments, and credit amounts.	Data is stored with a subject orientation that facilitates multiple views of the data and decision making. For example, sales may be recorded by product, division, manager, or region.
Time-variant	Data is recorded as current transactions. For example, the sales data may be the sale of a product on a given date, such as $342.78 on 12-MAY-2016.	Data is recorded with a historical perspective in mind. Therefore, a time dimension is added to facilitate data analysis and various time comparisons.
Nonvolatile	Data updates are frequent and common. For example, an inventory amount changes with each sale. Therefore, the data environment is fluid.	Data cannot be changed. Data is added only periodically from historical systems. Once the data is properly stored, no changes are allowed. Therefore, the data environment is relatively static.

FIGURE 13.5 THE ETL PROCESS لفهم الشكل الرجوع لصفحة 609

Ch.16

••

16-5a The DBA's Managerial Role : الدور الاداري

As a manager, the DBA must concentrate on the control and planning of database administration. Therefore, the DBA is responsible for the following: مسؤولياتة

1• Coordinating, monitoring, and allocating database administration resources: people and data

2 • Defining goals and formulating strategic plans for database administration

3 • Gathering user requirements. The DBA must work with end users to help gather the data required to identify and describe their present and future information needs. The DBA's communication skills are important in working closely with people who have varying computer backgrounds and communication styles

4• Building end-user confidence. Finding adequate solutions to end users' problems increases their trust and confidence in the DBA. The DBA also should educate end users about the services provided and how they enhance data stewardship and data security.

5•Resolving conflicts and problems. Finding solutions to end users' problems in one department might trigger conflicts with other departments. End users are typically concerned with their own data needs rather than those of others, and they might not consider how their data might affect other departments within the organization. When conflicts arise, the DBA must have the authority and responsibility to resolve them.

6• Finding solutions to information needs. The ability and authority to resolve data conflicts enables the DBA to develop solutions that will properly fit within the data management framework and address end users' information needs.

7• Ensuring quality and integrity of data and applications. Once the right solution has been found, it must be properly implemented and used. The DBA must work with application programmers and end users to teach them the database standards and procedures required for data quality, access, and manipulation.

8•Managing the training and support of DBMS users One of the most time-consuming DBA activities is teaching end users how to use the database. The DBA must ensure that all users understand the basic functions of the DBMS software. The DBA coordinates and monitors all DBMS training activities.

ROLES: Policies, Procedures, and Standards

policy General statement of direction that is used to manage company operations through the communication and support of the organization's objectives.

standard A detailed and specific set of instructions that describes the minimum requirements for a given activity. Standards are used to evaluate the quality of the output

The DBA must define, document, and communicate the following before they can be enforced:

1• Policies. are general statements of direction or action that communicate and support DBA goal

2• Standards describe the minimum requirements of a given DBA activity; they are more detailed and specific than policies.

3• Procedures are written instructions that describe a series of steps to be followed during the performance of a given activity. Procedures must be developed within existing working conditions, and they must support and enhance the work environment.

To illustrate the distinctions among policies, standards, and procedures, look at the following examples: مثال عليهم

Policies

• All users must have passwords.

• Passwords must be changed every six months.

Standards

• A password must have a minimum of 5 characters.

• A password must have a maximum of 12 characters.

• Social Security numbers, names, and birth dates cannot be used as passwords.

Procedures

To create a password

735مجموعة من الخطوات الرجوع الى صفحة

Securityو integrity/Areaysخاص Procedures أمثلة على

Standards and procedures defined by the DBA apply to all end users who want to benefit from the database. Standards and procedures must complement each other and must constitute an extension of data administration policies. Procedures must facilitate the work of end users and the DBA. The DBA must define, communicate, and enforce procedures that cover areas such as:

- End-user database requirements gathering. What documentation is required? What forms must be used?

- Database design and modeling. What database design methodology will be used (normalization or object-oriented)? What tools will be used (CASE tools, data dictionaries, UML or ER diagrams)?

- Documentation and naming conventions. What documentation must be used in the definition of all data elements, sets, and programs that access the database?

- Design, coding, and testing of database application programs. The DBA must define the standards for application program coding, documentation, and testing. The DBA standards and procedures are given to the application programmers, and the DBA must enforce those standards.

- Database software selection. The selected DBMS

- Database security and integrity. بصير تعديل لكل داتا بيس مرة واحدة في كل الاماكن The DBA must define policies that govern security and integrity. Database security is especially crucial. DB

- Database backup and recovery

procedure Series of steps to be followed during the performance of an activity or process.

database security officer (DSO) The person responsible for the security, integrity, backup, and recovery of the database.

IBM’s DB2 allow different types of backups: backup أنواع /مهم

1- full backup
2- incremental backup
3- concurrent backup

A full backup, also known as a database dump,
An incremental backup produces a backup of all data since the last backup date
A concurrent backup takes place while the user is working on the database.

full backup (database dump) A complete copy of an entire database saved and periodically updated in a separate memory location. A full backup ensures a full recovery of all data after a physical disaster or database integrity failure.

incremental backup A process that only backs up data that has changed in the database since the last incremental or full backup.

concurrent backup A backup that takes place while one or more users are working on a database.

16-5b The DBA's Technical Role:

The technical aspects of the DBA's job are rooted in the following areas of operation:

•Evaluating, selecting, and installing the DBMS and related utilities

• Designing and implementing databases and applications

• Testing and evaluating databases and applications

• Operating the DBMS, utilities, and applications

• Training and supporting users

• Maintaining the DBMS, utilities, and applications

Evaluating, Selecting, and Installing the DBMS and Utilities

16-6 Security:

-1Information system security refers to activities and measures that ensure the confidentiality, integrity, and availability of an information system and its main asset, data

2• Confidentiality deals with ensuring that data is protected against unauthorized access, and if the data is accessed by an authorized user, that it is used only for an authorized purpose.

3•Integrity, within the data security framework, is concerned with keeping data consistent and free of errors or anomalies.

integrity In a data security framework, refers to keeping data consistent and free of errors or anomalies. See also data integrity

16-6b Security Vulnerabilities:

A security vulnerability is a weakness in a system component that could be exploited to allow unauthorized access or cause service disruptions. Such vulnerabilities could fall under one of the following categories:

1• Technical. An example would be a flaw in the operating system or web browser.

2• Managerial. For example, an organization might not educate users about critical security issues.

3• Cultural. Users might hide passwords under their keyboards or forget to shred confidential reports.

4• Procedural. Company procedures might not require complex passwords or the checking of user IDs.

When a security vulnerability is left unchecked, it could become a security threat.

A security threat is an imminent security violation.

A security breach occurs when a security threat is exploited to endanger the integrity, confidentiality, or availability of the system

مهم جدا حفظهم بترتيب التسلسل

- 1availability In the context of data security, it refers to the accessibility of data whenever required by authorized users and for authorized purposes.

-2 security policy A collection of standards, policies, and procedures created to guarantee the security of a system and ensure auditing and compliance.

- 3security vulnerability A weakness in a system component that could be exploited to allow unauthorized access or cause service disruptions.

- 4security threat An imminent security violation that could occur due to unchecked security vulnerabilities.

- 5security breach An event in which a security threat is exploited to endanger the integrity, confidentiality, or availability of the system.

TABLE 16.4		
SAMPLE SECURITY VULNERABILITIES AND RELATED PROTECTIVE MEASURES		
SYSTEM COMPONENT	SECURITY VULNERABILITY	SECURITY MEASURES
People	• The user sets a blank password. • The password is short or includes a birth date. • The user leaves the office door open all the time. • The user leaves payroll information on the screen for long periods of time.	• Enforce complex password policies. • Use multilevel authentication. • Use security screens and screen savers. • Educate users about sensitive data. • Install security cameras. • Use automatic door locks.
Workstation and servers	• The user copies data to a flash drive. • The workstation is used by multiple users. • A power failure crashes the computer. • Unauthorized personnel can use the computer. • Sensitive data is stored on a laptop computer. • Data is lost due to a stolen hard disk or laptop. • A natural disaster occurs.	• Use group policies to restrict the use of flash drives. • Assign user access rights to workstations. • Install uninterrupted power supplies (UPSs). • Add security locks to computers. • Implement a kill switch for stolen laptops. • Create and test data backup and recovery plans. • Protect the system against natural disasters—use co-location strategies.

Operating system (OS)	• Buffer overflow attacks • Virus attacks • Root kits and worm attacks • Denial-of-service attacks • Trojan horses • Spyware applications • Password crackers	• Apply OS security patches and updates. • Apply application server patches. • Install antivirus and antispyware software. • Enforce audit trails on the computers. • Perform periodic system backups. • Install only authorized applications. • Use group policies to prevent unauthorized installations.
Applications	• Application bugs—buffer overflow • SQL injection, session hijacking, etc. • Application vulnerabilities—cross-site scripting, nonvalidated inputs • Email attacks—spamming, phishing, etc. • Social engineering emails	• Test application programs extensively. • Build safeguards into code. • Do extensive vulnerability testing in applications. • Install spam filters and antivirus software for email systems. • Use secure coding techniques (see www. owasp.org). • Educate users about social engineering attacks.
Network	• IP spoofing • Packet sniffers • Hacker attacks • Clear passwords on network	• Install firewalls. • Use virtual private networks (VPNs). • Use intrusion detection systems (IDSs). • Use network access control (NAC). • Use network activity monitoring.
Data	• Data shares are open to all users. • Data can be accessed remotely. • Data can be deleted from a shared resource	• Implement file system security. • Implement share access security. • Use access permission. • Encrypt data at the file system or database level.

** Shrouq Lami **

DB.2