اخر صفحة بالمُلخص مُهمة

17 / 18 / 19 / 20 / 21

• A **track** is divided into smaller **blocks** or sectors
• The **block size** B is **fixed** for each system.
• Typical **block sizes range from B=512 bytes to B=4096 bytes**.
• **Blocking factor (bfr)** refers to the number of records per block.

A physical disk block (hardware) address consists of:

1) a cylinder number (imaginary collection of tracks of same radius from all recorded surfaces)
2) the track number or surface number (within the cylinder)
3) and block number (within track).

Records Fixed and variable length records
• Records contain fields which have values of a particular type
• **A file descriptor** (or file header) includes information that describes the file, such as the field names and their data types
• **A file** can have fixed-length records or variable-length records.
• **File records** can be unspanned or spanned
○ **Unspanned:** no record can span two blocks
○ **Spanned:** a record can be stored in more than one block

Unordered Files called a heap or a pile file. (A linear search) b/2
○ New records are inserted at the end of the file.
○ Record insertion is quite efficient.

Ordered Files (sequential file)
○ Insertion is expensive, Reading efficient.
○ A binary search (Log2 b)
• **Hashing Techniques Types:** Internal Hashing (inside), External Hashing (Outside)
• **Main disadvantages of static external hashing:**
Fixed number of buckets M is a problem if the number of records in the file grows or shrinks.
Ordered access on the hash key is quite inefficient (requires sorting the records).

methods for collision resolution:
1. **Open addressing:** checks the subsequent positions in order until an empty position is found.
2. **Chaining:** extending the array with a number of overflow positions (pointer field is added)
3. **Multiple hashing:** applies a second hash function if the first results in a collision
• **Dynamic and Extendible Hashing Techniques:** dynamic hashing, extendible hashing, linear hashing
• **RAID** (Redundant Arrays of Inexpensive Disks) use for performance and reliability

RAID Levels:
○ Raid level 0 (uses striping) no redundant, best write performance at the risk of data loss.
○ Raid level 1 uses mirrored disks.
○ Raid level 2 uses memory-style redundancy by using Hamming codes. includes both error detection and correction.
○ Raid Level 5 with an extra drive for parity.
• **Design decisions for Raid include:** Level of Raid, Number of Disks, Choice of parity scheme, grouping of disks for block-level striping.

هذا الفيديو شامل لـ عملي شابتر 17 + شابتر 18
https://www.youtube.com/watch?v=7BmmOxbc-Fk&list=PLO6V6DeYbPNElVPIhQ9iHYzOwBINtekzN&index=34

• **Indexes as Access Paths:**
single-level index:
○ Primary Index: data file is ordered on a key field, primary index is a nondense (sparse) index.
○ Clustering Index: ordered on a non-key field, non-dense.
○ Secondary Index: non-ordering field, dense index
• One form of an index is a file of entries <field value, pointer to record>
• the index entry has the key field value for the first record in the block, called the **block anchor**
• Indexes characterized:
Dense index has an index entry for every search key value
sparse (or nondense) index, on the other hand, has index entries for only some of the search values

BY : TOKA AKRAD
WWW.SEU1.ORG

Multi-Level Indexes :

Because a single-level index is an ordered file, we can create a primary index to the index itself.
In this case, the original index file is called the first-level index and the index to the index is called the second-level index
We can repeat the process, creating a third, fourth, ..., top level until all entries of the top level fit in one disk block
A multi-level index can be created for any type of first-level index (primary, secondary, clustering) as long as the first-level index consists of more than one disk block
Such a multi-level index is a form of search tree
However, insertion and deletion of new index entries is a severe problem because every level of the index is an ordered file.

Difference between B-tree and B+-tree In a B-tree:

pointers to data records exist at all levels of the tree
In a B+-tree, all pointers to data records exists at the leaf-level nodes
A B+-tree can have less levels (or higher capacity of search values) than the corresponding B-tree

فيديوهات مُساعدة لفهم سؤال الـ B+Tree شابتر 18
https://www.youtube.com/watch?v=UcvU7hO0CQo
https://www.youtube.com/channel/UCbowfWkY0jVt2eM_r9oOUVA
https://www.youtube.com/watch?v=YIROy2XFjQk

- Query optimization: process of choosing a suitable لضفا execution strategy for processing a query.
- Query block: The basic unit that can be translated into the algebraic operators and optimized.
- Query tree: A tree data structure that corresponds to a relational algebra expression.
- Query graph: A graph data structure that corresponds to a relational calculus expression.

Process for heuristics optimization:

1. The parser of a high-level query generates an initial internal representation;
2. Apply heuristics rules to optimize the internal representation.
3. A query execution plan is generated to execute groups of operations based on the access paths available on the files involved in the query.

Cost Components for Query Execution:

1. Access cost to secondary storage
2. Storage cost
3. Computation cost
4. Memory usage cost
5. Communication cost

- Materialized evaluation: the result of an operation is stored as a temporary relation.
- Pipelined evaluation: as the result of an operator is produced, it is forwarded to the next operator in sequence.

- **Factors that Influence Physical Database Design:**

1. **Analyzing the database queries and transactions.**
2. **Analyzing the expected frequency of invocation of queries and transactions (80-20 rule)**
3. **Analyzing the time constraints of queries and transactions**
4. **Analyzing the expected frequencies of update operations.**
5. **Analyzing the uniqueness constraints on attributes**

The goal of normalization is to separate the logically related attributes into tables to minimize redundancy
The goal of denormalization is to improve the performance of frequently occurring queries and transactions.

Tuning: process of continuing to revise/adjust the physical database design by monitoring resource utilization as well as internal DBMS processing to reveal bottlenecks such as contention for the same data or devices

Tuning goal:

1. To make application run faster
2. To lower the response time of queries/transactions
3. To improve the overall throughput الانتاجية of transactions

Statistics internally collected in DBMSs:

- Size of individual tables
- Number of distinct values in a column
- The number of times a particular query or transaction is submitted/executed in an interval of time

• The times required for different phases of query and transaction processing

Statistics obtained from monitoring:
1. Storage statistics
2. I/O and device performance statistics
3. Query/transaction processing statistics
4. Locking/logging related statistics Index statistic

Reasons to tuning indexes
Certain queries may take too long to run for lack of an index;
Certain indexes may not get utilized at all
Certain indexes may be causing excessive overhead because the index is on an attribute that undergoes frequent changes

Options to tuning indexes
Drop or/and build new indexes
Change a non-clustered index to a clustered index (and vice versa)
Rebuilding the index

Tuning the Database Design: Dynamically changed processing requirements need to be addressed by making changes to the conceptual schema if necessary and to reflect those changes into the logical schema and physical design.

Possible changes to the database design:
1. Existing tables may be joined (denormalized) because certain attributes from two or more tables are frequently needed together.
2. For the given set of tables, there may be alternative design choices, all of which achieve 3NF or BCNF. One may be replaced by the other.
3. A relation of the form R(K, A, B, C, D, …) that is in BCNF can be stored into multiple tables that are also in BCNF by replicating the key K in each table.
4. Attribute(s) from one table may be repeated in another even though this creates redundancy and potential anomalies.
5. Apply horizontal partitioning as well as vertical partitioning if necessary.

Indications for tuning queries
▪ A query issues too many disk accesses
▪ The query plan shows that relevant indexes are not being used.

Typical instances for query tuning
1. In some situations, involving using of correlated queries, temporaries are useful.
2. If multiple options for join condition are possible, choose one that uses a clustering index and avoid those that contain string comparisons.
3. The order of tables in the FROM clause may affect the join processing.
4. Some query optimizers perform worse on nested queries compared to their equivalent un-nested counterparts.
5. Many applications are based on views that define the data of interest to those applications. Sometimes these views become an overkill.

Additional Query Tuning Guidelines:
A query with **multiple selection** conditions that are connected via OR may not be prompting the query optimizer to use any index. Such a query may be split up and expressed as a union of queries, each with a condition on an attribute that causes an index to be used.
Query have **NOT** condition may be transformed into a positive expression.
Query have Embedded SELECT blocks may be replaced by joins
If an **equality join is set up between two tables**, the range predicate on the joining attribute set up in one table may be repeated for the other table
WHERE conditions may be rewritten to utilize the indexes on multiple columns.
Single-User System: At most one user at a time can use the system.
Multiuser System: Many users can access the system concurrently.
Concurrency
- Interleaved processing: Concurrent execution of processes is interleaved in a single CPU
- Parallel processing: Processes are concurrently executed in multiple CPUs.

A Transaction: Logical unit of database processing that includes one or more access operations (read -retrieval, write - insert or update, delete).
Transaction boundaries**:** Begin and End transaction.
A database is a collection of named data items, Basic operations are read and write
Read and Write operations:
read_item(X) command includes the following steps:
1. Find the address of the disk block that contains item X.
2. Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).
3. Copy item X from the buffer to the program variable named X.
write_item(X) command includes the following steps:
1. Find the address of the disk block that contains item X.
2. Copy that disk block into a buffer in main memory (if that disk block is not already in some main memory buffer).
3. Copy item X from the program variable named X into its correct location in the buffer.
4. Store the updated block from the buffer back to disk (either immediately or at some later point in time).

What causes a Transaction to fail?
1. computer failure (system crash)
2. A transaction or system error
3. Local errors or exception conditions detected by the transaction
4. Concurrency control enforcement
5. Disk failure
6. Physical problems and catastrophes

Why Concurrency Control is needed?
• The Lost Update Problem: This occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect
• The Temporary Update (or Dirty Read) Problem: This occurs when one transaction updates a database item and then the transaction fails for some reason
• The Incorrect Summary Problem: If one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some of these records, the aggregate function may calculate some values before they are updated and others after they are updated.

A transaction is an atomic unit of work that is either completed in its entirety or not done at all, For **recovery purposes**, the system needs to keep track of when the transaction starts, terminates, and commits or aborts.
Transaction states: Active state, partially committed state, committed state, Failed state, Terminated State

Recovery manager keeps track of the following operations:
• **begin transaction**: This marks the beginning of transaction execution.
• **read or write:** These specify read or write operations on the database items that are executed as part of a transaction.
• **end transaction:** This specifies that read and write transaction operations have ended and marks the end limit of transaction execution.
• **commit transaction:** This signals a successful end of the transaction so that any changes (updates) executed by the transaction can be safely committed to the database and will not be undone.
• **rollback (or abort):** This signals that the transaction has ended unsuccessfully, so that any changes or effects that the transaction may have applied to the database must be undone.
• **undo:** Similar to rollback except that it applies to a single operation rather than to a whole transaction.
• **redo:** This specifies that certain transaction operations must be redone to ensure that all the operations of a committed transaction have been applied successfully to the database.

The System Log (or Journal): The log keeps track of all transaction operations that affect the values of database items. This information may be needed to permit recovery from transaction failures.

Types of log record :
1. **[start_transaction,T]:** Records that transaction T has started execution.
2. [**write_item,T,X,old_value,new_value]:** Records that transaction T has changed the value of database item X from old_value to new_value.
3. **[read_item,T,X]:** Records that transaction T has read the value of database item X.
4. **[commit,T]:** Records that transaction T has completed successfully, and affirms that its effect can be committed (recorded permanently) to the database.
5. **[abort,T]:** Records that transaction T has been aborted.

• **Commit Point :** A transaction T reaches its commit point when all its operations that access the database have been executed successfully and the effect of all the transaction operations on the database has been recorded in the log
• **Roll Back of transactions:** Needed for transactions that have a [start_transaction,T] entry into the log but no commit entry [commit,T] into the log.
• **Redoing transactions:** Transactions that have written their commit entry in the log must also have recorded all their write operations in the log
• **Force writing a log:** Before a transaction reaches its commit point, any portion of the log that has not been written to the disk yet must now be written to the disk. This process is called **force-writing** the log file before committing a transaction.

Properties of Transactions(ACID)
• **Atomicity:** A transaction is an atomic unit of processing; it is either performed in its entirety or not performed at all.
• **Consistency preservation:** A correct execution of the transaction must take the database from one consistent state to another.
• **Isolation:** A transaction should not make its updates visible to other transactions until it is committed;
• **Durability or permanency:** Once a transaction changes the database and the changes are committed, these changes must never be lost because of subsequent failure.

Characterizing Schedules Based on Recoverability:
• Transaction schedule or history
• A schedule (or history) S of n transactions T1, T2, …, Tn:

Schedules classified on recoverability
1. Recoverable schedule
2. Cascadeless schedule
3. Schedules requiring cascaded rollback
4. Strict Schedules

Characterizing Schedules Based on Serializability
1. **Serial schedule:** schedule S is serial if, for every transaction T participating in the schedule, all the operations of T are executed consecutively in the schedule.
2. **Serializable schedule:** A schedule S is serializable if it is equivalent to some serial schedule of the same n transactions.
3. **Result equivalent:** Two schedules are called result equivalent if they produce the same final state of the database
4. **Conflict equivalent:** Two schedules are said to be conflict equivalent if the order of any two conflicting operations is the same in both schedules
5. **Conflict serializable:** A schedule S is said to be conflict serializable if it is conflict equivalent to some serial schedule S'.
6. **Being serializable is not the same as being serial**
7. **Being serializable implies that the schedule is a correct schedule**
8. **Serializability is hard to check.**

https://www.youtube.com/watch?v=abvQe1Le4h8

شرح للسؤال الرابع الواجب الثاني عملي شابتر 21

CHAPTER – 22

Purpose of Concurrency Control

1) To enforce Isolation among conflicting transactions.
2) To preserve database consistency through consistency preserving execution of transactions.
3) To resolve read-write and write-write conflicts.

Two-Phase Locking Techniques

Locking is an operation which secures

- permission to Read
- permission to Write a data item for a transaction.

Unlocking is an operation which removes these permissions from the data item.

Lock and Unlock are Atomic operations.

Two locks modes: (a) shared (read) (b) exclusive (write).

- **Shared mode: shared lock (X)**

More than one transaction can apply share lock on X for reading its value but no write lock can be applied on X by any other transaction.

- **Exclusive mode: Write lock (X)**

Only one write lock on X can exist at any time and no shared lock can be applied by any other transaction on X.

Lock Manager: Managing locks on data items.
Lock table: Lock manager uses it to store the identify of transaction locking a data item, the data item, lock mode and pointer to the next data item locked.

A transaction is well-formed if:

It must lock the data item before it reads or writes to it.
It must not lock already locked data items and it must not try to unlock a free data item.

Lock conversion

- Lock upgrade: existing read lock to write lock
- Lock downgrade: existing write lock to read lock

Two-Phase Locking Techniques: The algorithm

1) **Locking (Growing) Phase:**
   A transaction applies locks (read or write) on desired data items one at a time.
2) **Unlocking (Shrinking) Phase:**
   A transaction unlocks its locked data items one at a time.

- **Requirement:** For a transaction these two phases must be mutually exclusively, that is, during locking phase unlocking phase must not start and during unlocking phase locking phase must not begin.

Two-phase policy generates two locking algorithms

1) **Conservative:**

Prevents deadlock by locking all desired data items before transaction begins execution.

2) **Basic:**

Transaction locks data items incrementally. This may cause deadlock which is dealt with.

Strict: A more stricter version of Basic algorithm where unlocking is performed after a transaction terminates (commits or aborts and rolled-back). This is the most commonly used two-phase locking algorithm.

Deadlock prevention

- A transaction locks all data items it refers to before it begins execution.
- This way of locking prevents deadlock since a transaction never waits for a data item.
- The conservative two-phase locking uses this approach.

Deadlock detection and resolution

- deadlocks are allowed to happen. The scheduler maintains a wait-for-graph for detecting cycle. If a cycle exists, then one transaction involved in the cycle is selected (victim) and rolled-back.
- A wait-for-graph is created using the lock table. As soon as a transaction is blocked, it is added to the graph

Deadlock avoidance

- There are many variations of two-phase locking algorithm.
- Some avoid deadlock by not letting the cycle to complete.
- That is as soon as the algorithm discovers that blocking a transaction is likely to create a cycle, it rolls back the transaction.
- Wound-Wait and Wait-Die algorithms use timestamps to avoid deadlocks by rolling-back victim.

Starvation

- Starvation occurs when a particular transaction consistently waits or restarted and never gets a chance to proceed further.
- In a deadlock resolution it is possible that the same transaction may consistently be selected as victim and rolled-back.
- This limitation is inherent in all priority based scheduling mechanisms.
- In Wound-Wait scheme a younger transaction may always be wounded (aborted) by a long running older transaction which may create starvation.

Timestamp based concurrency control algorithm

Timestamp

- A monotonically increasing variable (integer) indicating the age of an operation or a transaction. A larger timestamp value indicates a more recent event or operation.
- Timestamp based algorithm uses timestamp to serialize the execution of concurrent transactions.

Multiversion concurrency control techniques

- maintains a number of versions of a data item and allocates the right version to a read operation of a transaction and a read operation in this mechanism is never rejected.
- Side effect: Significantly more storage (RAM and disk) is required to maintain multiple versions.

Multiversion Two-Phase Locking Using Certify Locks

Concept

- Allow a transaction T' to read a data item X while it is write locked by a conflicting transaction T.
- This is accomplished by maintaining two versions of each data item X where one version must always have been written by some committed transaction. This means a write operation always creates a new version of X.

Multiversion Two-Phase Locking Using Certify Locks

- **Steps**

1) X is the committed version of a data item.
2) T creates a second version X' after obtaining a write lock on X.
3) Other transactions continue to read X.
4) T is ready to commit so it obtains a certify lock on X'.
5) The committed version X becomes X'.
6) T releases its certify lock on X', which is X now.

Note:

In multiversion 2PL read and write operations from conflicting transactions can be processed concurrently. This improves concurrency but it may delay transaction commit because of obtaining certify locks on all its writes. It avoids cascading abort but like strict two phase locking scheme conflicting transactions may get deadlocked.

Validation (Optimistic) Concurrency Control Schemes

- In this technique only at the time of commit serializability is checked and transactions are aborted in case of non-serializable schedules.
- **Three phases:** Read phase / Validation phase / Write phase

S= shared / X= exclusive / Is= intention-shared =read / Ix= intention-exclusive =write / Six= shared-intention-exclusive

شرح عملي شبتر 22 !
ارجعوا للشبتر ولمثال الاسايمنت ..

https://www.youtube.com/watch?v=3t4IdYV-Kho&list=PLO6V6DeYbPNElVPIhQ9iHYzOwBINtekzN&index=36

https://www.youtube.com/watch?v=LF3ga_NJIvA&feature=share

CHAPTER – 23

Purpose of Database Recovery
- To bring the database into the last consistent state, which existed prior to the failure.
- To preserve transaction properties (Atomicity, Consistency, Isolation and Durability).

Types of Failure The database may become unavailable for use due: **(مُهِم)**
1) Transaction failure: Transactions may fail because of incorrect input, deadlock, incorrect synchronization.
2) System failure: System may fail because of addressing error, application error, operating system fault, RAM failure, etc.
3) Media failure: Disk head crash, power disruption, etc.

Transaction Log
- For recovery from any type of failure data values prior to modification (BFIM - BeFore Image) and new value after modification (AFIM – After Image).
- These values and other information is stored in a sequential file called Transaction log.

Data Update (مُهِم)
1) Immediate Update: a data item is modified in cache , the disk copy is updated.
2) Deferred Update: All modified data items in the cache is written after a transaction ends its execution or after a fixed number of transactions completed their execution.
3) Shadow update: The modified version of a data item does not overwrite its disk copy but is written at a separate disk location.
4) In-place update: The disk version of the data item is overwritten by cache version.

Data Caching
- Data items to be modified are first stored into database cache by the Cache Manager (CM) and after modification they are flushed (written) to the disk.
- The flushing is controlled by:

Pin-Unpin: Instructs the operating system not to flush the data item.
Modified: Indicates the AFIM of the data item.

Transaction Roll-back (Undo) and Roll-Forward (Redo)
- To maintain atomicity, a transaction's operations are;
  Undo: Restore all BFIMs on to disk (Remove all AFIMs).
  Redo: Restore all AFIMs on to disk.
- Database recovery is achieved either by performing only Undos or only Redos or by a combination of the two.

Write-Ahead Logging Write-Ahead Logging (WAL) protocol.
- When in-place update (immediate or deferred) is used then log is necessary for recovery and it must be available to recovery manager.
- WAL states that
- For Undo: BFIM must be written to the log and the log must be saved on a stable store (log disk).
- For Redo: Before a transaction executes its commit operation, all its AFIMs must be written to the log and the log must be saved on a stable store.

Checkpointing
Time to time (randomly or under some criteria) the database flushes its buffer to database disk to minimize the task of recovery.

The following steps defines a checkpoint operation:
1) Suspend execution of transactions temporarily.
2) Force write modified buffer data to disk.
3) Write a [checkpoint] record to the log, save the log to disk.
4) Resume normal transaction execution.

✓ During recovery redo or undo is required to transactions appearing after [checkpoint] record.

Possible ways for flushing database cache to database disk:
1) Steal: Cache can be flushed before transaction commits.
2) No-Steal: Cache cannot be flushed before transaction commit.
3) Force: Cache is immediately flushed (forced) to disk.

4) No-Force: Cache is deferred until transaction commits.

These give rise to four different ways for handling recovery:

1) Steal/No-Force (Undo/Redo)
2) No-Steal/No-Force (Redo/No-undo)
3) Steal/Force (Undo/No-redo)
4) No-Steal/Force (No-undo/No-redo)

Recovery Scheme Deferred Update (No Undo/Redo)

The data update goes as follows:

- A set of transactions records their updates in the log.
- At commit point under WAL scheme these updates are saved on database disk.
- After reboot from a failure the log is used to redo all the transactions affected by this failure. No undo is required because no AFIM is flushed to the disk before a transaction commits.
- Deferred Update in a single-user system

There is no concurrent data sharing in a single user system. The data update goes as follows:

- A set of transactions records their updates in the log.
- At commit point under WAL scheme these updates are saved on database disk.
- After reboot from a failure the log is used to redo all the transactions affected by this failure. Deferred Update with concurrent users

Two tables are required for implementing this protocol:

- Active table: All active transactions are entered in this table.
- Commit table: Transactions to be committed are entered in this table.

During recovery, all transactions of the commit table are redone and all transactions of active tables are ignored since none of their AFIMs reached the database

Recovery Techniques Based on Immediate Update

Undo/No-redo Algorithm

- In this algorithm AFIMs of a transaction are flushed to the database disk under WAL before it commits.
- For this reason the recovery manager undoes all transactions during recovery.
- No transaction is redone.
- It is possible that a transaction might have completed execution and ready to commit but this transaction is also undone.

Undo/Redo Algorithm (Single-user environment)

- Recovery schemes of this category apply undo and also redo for recovery.
- In a single-user environment no concurrency control is required but a log is maintained under WAL.
- Note that at any time there will be one transaction in the system and it will be either in the commit table or in the active table.
- The recovery manager performs:

Undo of a transaction if it is in the active table.
Redo of a transaction if it is in the commit table.

Undo/Redo Algorithm (Concurrent execution)

- Recovery schemes of this category applies undo and also redo to recover the database from failure.
- In concurrent execution environment a concurrency control is required and log is maintained under WAL.
- Commit table records transactions to be committed and active table records active transactions. To minimize the work of the recovery manager checkpointing is used.

The recovery performs:
Undo of a transaction if it is in the active table.
Redo of a transaction if it is in the commit table.

Shadow Paging

- The AFIM does not overwrite its BFIM but recorded at another place on the disk. Thus, at any time a data item has AFIM and BFIM (Shadow copy of the data item) at two different places on the disk.
- To manage access of data items by concurrent transactions two directories (current and shadow) are used.

The ARIES Recovery Algorithm The ARIES Recovery Algorithm is based on:

- WAL (Write Ahead Logging)
- Repeating history during redo: ARIES will retrace all actions of the database system prior to the crash to reconstruct the database state when the crash occurred.
- Logging changes during undo: It will prevent ARIES from repeating the completed undo operations if a failure occurs during recovery, which causes a restart of the recovery process.

The ARIES recovery algorithm consists of three steps:

1) Analysis: step identifies the dirty (updated) pages in the buffer and the set of transactions active at the time of crash
2) Redo: necessary redo operations are applied.
3) Undo: log is scanned backwards and the operations of transactions active at the time of crash are undone in reverse order.

The Log and Log Sequence Number (LSN)

A log record written for: (a) data update (b) transaction commit (c) transaction abort (d) undo (e) transaction end

A unique LSN is associated with every log record.

- LSN increases monotonically and indicates the disk address of the log record it is associated with.
- each data page stores the LSN of the latest log record corresponding to a change for that page.

A log record stores

(a) the previous LSN of that transaction (b) the transaction ID (c) the type of log record.

A log record stores:

1) Previous LSN of that transaction: It links the log record of each transaction. It is like a back pointer points to the previous record of the same transaction
2) Transaction ID
3) Type of log record

For a write operation the following additional information is logged:

1) Page ID for the page that includes the item
2) Its offset from the beginning of the page
3) AFIM of the item
4) Length of the updated item
5) BFIM of the item

For efficient recovery following tables are also stored in the log during checkpointing:

Transaction table: Contains an entry for each active transaction, with information such as transaction ID, transaction status and the LSN of the most recent log record for the transaction.

Dirty Page table: Contains an entry for each dirty page in the buffer, which includes the page ID and the LSN corresponding to the earliest update to that page.

A checkpointing does the following:

- Writes a begin_checkpoint record in the log
- Writes an end_checkpoint record in the log. With this record the contents of transaction table and dirty page table are appended to the end of the log.
- Writes the LSN of the begin_checkpoint record to a special file. This special file is accessed during recovery to locate the last checkpoint information.

The following steps are performed for recovery

Analysis phase: Start at the begin_checkpoint record and proceed to the end_checkpoint record. Access transaction table and dirty page table are appended to the end of the log

Redo phase: Starts from the point in the log up to where all dirty pages have been flushed, and move forward to the end of the log.

Undo phase: Starts from the end of the log and proceeds backward while performing appropriate undo.

Recovery in multidatabase system

A multidatabase system is a special distributed database system where one node .

CHAPTER– 24

Introduction to Database Security Issues

Types of Security

1) Legal and ethical issues
2) System-related issues
3) Policy issues
4) The need to identify multiple security levels

Threats to databases

1) Loss of integrity
2) Loss of availability
3) Loss of confidentiality

To protect databases against these types of threats four kinds of countermeasures can be implemented:

1) Access control
2) Inference control
3) Flow control
4) Encryption

Two types of database security mechanisms:

1) Discretionary security mechanisms
2) Mandatory security mechanisms

✓ The security mechanism of a DBMS must include provisions for restricting access to the database as a whole This function is called access control and is handled by creating user accounts and passwords to control login process by the DBMS.

✓ The countermeasures to statistical database security problem is called inference control measures.

✓ Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called covert channels

✓ A final security issue is data encryption, which is used to protect sensitive data (such as credit card numbers) that is being transmitted via some type communication network.

✓ The data is encoded using some encoding algorithm.

✓ An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting algorithms (or keys) to decipher data.

Database Security and the DBA The database administrator (DBA) is the central authority for managing a database system

The DBA's responsibilities include

1) granting privileges to users who need to use the system
2) classifying users and data in accordance with the policy of the organization

- The DBA is responsible for the overall security of the database system.
- The DBA has a DBA account in the DBMS
- Sometimes these are called a system or superuser account

These accounts provide powerful capabilities such as:

Account creation/ privilege granting / Privilege revocation / Security level assignment

- If any tampering with the database is suspected, a database audit is performed
- A database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period.
- A database log that is used mainly for security purposes is sometimes called an audit trail.

Discretionary Access Control Based on Granting and Revoking Privileges

- The typical method of enforcing discretionary access control in a database system is based on the granting and revoking privileges.

Types of Discretionary Privileges

The account level:

- At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.

The relation level (or table level):

- At this level, the DBA can control the privilege to access each individual relation or view in the database.
- The privileges at the account level apply to the capabilities provided to the account itself and can include
- the CREATE SCHEMA or CREATE TABLE privilege, to create a schema or base relation;
- the CREATE VIEW privilege;

- the ALTER privilege, to apply schema changes such adding or removing attributes from relations;
- the DROP privilege, to delete relations or views;
- the MODIFY privilege, to insert, delete, or update tuples;
- and the SELECT privilege, to retrieve information from the database by using a SELECT query.

- The second level of privileges applies to the relation level
- This includes base relations and virtual (view) relations.
- The granting and revoking of privileges generally follow an authorization model for discretionary privileges known as the access matrix model where
- The rows of a matrix M represents subjects (users, accounts, programs)
- The columns represent objects (relations, records, columns, views, operations).
- Each position M(i,j) in the matrix represents the types of privileges (read, write, update) that subject i holds on object j.
- To control the granting and revoking of relation privileges, each relation R in a database is assigned and owner account, which is typically the account that was used when the relation was created in the first place.
- The owner of a relation is given all privileges on that relation.
- In SQL2, the DBA can assign and owner to a whole schema by creating the schema and associating the appropriate authorization identifier with that schema, using the CREATE SCHEMA command.
- The owner account holder can pass privileges on any of the owned relation to other users by granting privileges to their accounts.

In SQL the following types of privileges can be granted on each individual relation R:

SELECT (retrieval or read) privilege on R:

- Gives the account retrieval privilege.
- In SQL this gives the account the privilege to use the SELECT statement to retrieve tuples from R.

MODIFY privileges on R:

- This gives the account the capability to modify tuples of R.
- In SQL this privilege is further divided into UPDATE, DELETE, and INSERT privileges to apply the corresponding SQL command to R.
- In addition, both the INSERT and UPDATE privileges can specify that only certain attributes can be updated by the account.

REFERENCES privilege on R:

- This gives the account the capability to reference relation R when specifying integrity constraints.
- The privilege can also be restricted to specific attributes of R.
- Notice that to create a view, the account must have SELECT privilege on all relations involved in the view definition.

من سلايد 25 الى سلايد 30 مهم جدا ارجعو للسلايدات جات نفس الامثلة بالفاينل وكان عليها 10 درجات

Comparing Discretionary Access Control and Mandatory Access Control

- ✓ In discretionary access control (DAC), the owner of the object specifies which subjects can access the object. This model is called discretionary because the control of access is based on the discretion of the owner.
- ✓ Most operating systems such as all Windows, Linux, and Macintosh and most flavors of UNIX are based on DAC models.
- ✓ **In mandatory access control (MAC),** the system (and not the users) specifies which subjects can access specific data objects.
- ✓ The MAC model is based on security labels. Subjects are given a security clearance (secret, top secret, confidential, etc.), and data objects are given a security classification (secret, top secret, confidential, etc.). The clearance and classification data are stored in the security labels, which are bound to the specific subjects and objects.
- ✓ Examples of the MAC-based commercial systems are SE Linux and Trusted Solaris.

- ✓ **E-Commerce environments require elaborate policies that go beyond traditional DBMSs.**
- ✓ **In an e-commerce environment the resources to be protected are not only traditional data but also knowledge and experience.**
- ✓ **The access control mechanism should be flexible enough to support a wide spectrum of heterogeneous protection objects.**
- ✓ **A related requirement is the support for content-based access-control.**
- ✓ **A credential is a set of properties concerning a user that are relevant for security purposes**
- ✓ **Statistical databases are used mainly to produce statistics on various populations.**
- ✓ **The database may contain confidential data on individuals, which should be protected from user access.**
- ✓ Users are permitted to retrieve statistical information on the populations, such as averages, sums, counts, maximums, minimums, and standard deviations.
- ✓ **A population is a set of tuples of a relation (table) that satisfy some selection condition.**
- ✓ **Statistical queries involve applying statistical functions to a population of tuples.**
- ✓ **A covert channel allows a transfer of information that violates the security or the policy and allows information to pass from a higher classification level to a lower classification level through improper means**
- ✓ **Covert channels can be classified into two broad categories:**
- ✓ **Storage channels do not require any temporal synchronization, in that information is conveyed by accessing system information or what is otherwise inaccessible to the user.**
- ✓ **Timing channel allow the information to be conveyed by the timing of events or processes.**

Encryption and Public Key Infrastructures

- Encryption is a means of maintaining secure data in an insecure environment.
- Encryption consists of applying an encryption algorithm to data using some prespecified encryption key.
- The resulting data has to be decrypted using a decryption key to recover the original data.

The Data and Advanced Encryption Standards

- The Data Encryption Standard (DES) is a system developed by the U.S. government for use by the general public.
- It has been widely accepted as a cryptographic standard both in the United States and abroad.

DES algorithm is a careful and complex combination of two of the fundamental building blocks of encryption:

- substitution and permutation (transposition).
- Plaintext (the original form of the message) is encrypted as blocks of 64 bits.

The two keys used for public key encryption are referred to as the public key and the private key.
the private key is kept secret, but it is referred to as private key rather than a secret key (the word used in conventional encryption to avoid confusion with conventional encryption).

A public key encryption scheme, or infrastructure, has six ingredients:

- Plaintext: This is the data or readable message that is fed into the algorithm as input.
- Encryption algorithm: The encryption algorithm performs various transformations on the plaintext.
- Public and private keys: These are pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
- A public key encryption scheme, or infrastructure, has six ingredients:
- Plaintext: This is the data or readable message that is fed into the algorithm as input.
- Ciphertext: This is the scrambled message produced as output.
- Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext

- ✓ Public key is made for public and private key is known only by owner.
- ✓ A general-purpose public key cryptographic algorithm relies on one key for encryption and a different but related key for decryption

- ✓ digital signature is an example of using encryption techniques to provide authentication services in e-commerce applications.
- ✓ A digital signature is a means of associating a mark unique to an individual with a body of text.
- ✓ A digital signature consists of a string of symbols.
- ✓ Signature must be different for each use.

CHAPTER 25

(التعريف مُهم)

A distributed database (DDB) collection of multiple logically related database distributed over a computer network, and a distributed database management system as a software system that manages a distributed database while making the distribution transparent to the user.

(فواند قواعد البيانات الموزعة سؤال مُهم مُمكن تعداد او تعداد مع شرح)

Advantages of distributed database (DDB)

1) **Management of distributed data with different levels of transparency**
- This refers to the physical placement of data (files, relations)

2) **Distribution and Network transparency**
- Users do not have to worry about operational details of the network.
- Location transparency; refers to freedom of issuing command from any location without affecting it's working.
- Naming transparency; allows access to any names object from any location.

3) **Replication transparency**
- It allows to store copies of a data at multiple sites.
- This is done to minimize access time to the required data.

4) **Fragmentation transparency**
- Allows to fragment a relation:
- horizontally (create a subset of tuples of a relation)
- vertically (create a subset of columns of a relation).

5) **Increased reliability and availability**
- Reliability refers to system live time, system is running efficiently most of the time.
- **Availability is the probability that the system is continuously available during a time interval.**
- A distributed database system has multiple (computers) if one fails then others are available to do the job.

6) **Improved performance**
- **distributed DBMS fragments the database to keep data closer to where it is needed most.**
- This reduces data management (access and modification)

7) **Easier expansion (scalability):**
- Allows new nodes (computers) to be added anytime without chaining the entire configuration.

Data Fragmentation Split a relation into logically related and correct parts.
A relation can be fragmented in two ways: Horizontal and Vertical Fragmentation

من سلايد 12 الى سلايد 15 جُزئية عملية.. رابط شرح لها:
من الدقيقة 22 الـ Fragmentation
https://youtu.be/0tagJGW39fw

Fragmentation schema
- ✓ A definition of a set of fragments (horizontal or vertical or horizontal and vertical)
- ✓ includes all attributes and tuples in the database that satisfies the condition that the whole database can be reconstructed from the fragments by applying some sequence of UNION (or OUTER JOIN) and UNION operations.

Allocation schema
- ✓ it describes the distribution of fragments to sites of distributed databases. It can be fully or partially replicated or can be partitioned.

Data Replication
- ✓ Database is replicated to all sites.
- ✓ In full replication the entire database is replicated and in partial replication some selected part is replicated to some of the sites.
- ✓ Data replication is achieved through a replication schema.

Data Distribution (Data Allocation)
- ✓ relevant only in the case of partial replication or partition.
- ✓ The selected portion of the database is distributed to the database sites.

Types of Distributed Database Systems (الانواع مُهمة ولازم نميز بينهم)

1) **Homogeneous** All sites of the database system have identical setup, i.e., same database system software.
- The underlying operating system may be different.
- For example, all sites run Oracle or DB2, or Sybase or some other database system.
- The underlying operating systems can be a mixture of Linux, Window, Unix, etc.

2) Heterogeneous

- **Federated:** Each site may run different database system but the data access is managed through a single conceptual schema.
- This implies that the degree of local autonomy is minimum. Each site must adhere to a centralized access policy.
- **Multidatabase:** There is no one conceptual global schema. For data access a schema is constructed dynamically as needed by the application software.

Federated Database Management Systems Issues

1) Differences in data models: Relational, Objected oriented, hierarchical, network.
2) Differences in constraints: Each site may have their own data accessing and processing constraints.
3) Differences in query language: use SQL

من سلايد 20 الي 28 جزئية عملية وسؤال مُكرر لونق انسر بفايئل سابق نفس المعطيات!
رابط شرح نفس مثال السلايد..
https://youtu.be/JhMHWIlCJRs?list=PLO6V6DeYbPNElVPIhQ9iHYzOwBINtekzN

What is the problem Distributed Databases

1) Dealing with multiple copies of data items
The concurrency control must maintain global consistency.
2) Failure of individual sites
Database availability must not be affected due to the failure of one or two sites and the recovery scheme must recover them before they are available for use.
3) Communication link failure
This failure may create network partition which would affect database availability even though all database sites may be running.
4) Distributed commit
A transaction may be fragmented, and they may be executed by a number of sites.
5) Distributed deadlock
transactions are processed at multiple sites, two or more sites may get involved in deadlock.

Transaction management:

- Concurrency control and commit are managed by this site.
- In two phase locking, this site manages locking and releasing data items.

Advantages:

- An extension to the centralized two phase locking so implementation and management is simple.
- Data items are locked only at one site but they can be accessed at any site.

Disadvantages:

- All transaction management activities go to primary site
- If the primary site fails, the entire system is inaccessible.

Primary Copy Technique:

- In instead of a site, a data item partition is designated as primary copy.
- To lock a data item just the primary copy of the data item is locked.

Advantages:

- Since primary copies are distributed at various sites, a single site is not overloaded with locking and unlocking requests.

Disadvantages:

- Identification of a primary copy is complex. A distributed directory must be maintained, possibly at all sites.

Recovery from a coordinator failure

- In both approaches a coordinator site or copy may become unavailable. This will require the selection of a new coordinator.

Primary site approach with no backup site:

- Aborts and restarts all active transactions at all sites. Elects a new coordinator and initiates transaction processing.

Primary site approach with backup site:

- Suspends all active transactions, designates the backup site as the primary site and identifies a new back up site.

Primary and backup sites fail or no backup site:

- Use election process to select a new coordinator site

(قراءة فقط)

- Clients reach server for desired service, but server does reach clients.
- The server software is responsible for local data management at a site, much like centralized DBMS software.
- The client software is responsible for most of the distribution function.
- The communication software manages communication among clients and servers.

(مُهم)

The processing of a SQL queries in Client-Server Database Architecture :

1) Client parses a user query and decomposes it into a number of independent sub-queries.
2) Each subquery is sent to appropriate site for execution.
3) Each server processes its query and sends the result to the client.
4) The client combines the results of subqueries and produces the final result.

CHAPTER 26

مُمكن يجي سؤال عملي عن الترايقر مثال: اسايمنت 4 السؤال الأول

Triggers are executed when a specified condition occurs during insert/delete/update
Triggers are action that fire automatically based on these conditions.
Triggers follow an Event-condition-action (ECA) model
Event : Database modification E.g., insert, delete, update
Condition: Any true/false expression / Optional: If no condition is specified then condition is always true
Action: Sequence of SQL statements that will be automatically executed

CREATE or ALTER TRIGGER

CREATE TRIGGER <name>	Creates a trigger
ALTER TRIGGER <name>	Alters a trigger (assuming one exists)
CREATE OR ALTER TRIGGER <name>	Creates a trigger if one does not exist

- Alters a trigger if one does exist
- Works in both cases, whether a trigger exists or not

AFTER Executes after the event
BEFORE Executes before the event
INSTEAD OF Executes instead of the event
Note that event does not execute in this case
Triggers can be
Row-level FOR EACH ROW specifies a row-level trigger
Statement-level Default (when FOR EACH ROW is not specified)
Row level triggers Executed separately for each affected row
Statement-level triggers Execute once for the SQL statement

- Any true/false condition to control whether a trigger is activated on not
- Absence of condition means that the trigger will always execute for the even
- Otherwise, condition is evaluated
- before the event for BEFORE trigger
- after the event for AFTER trigger

Action can be One SQL statement A sequence of SQL statements enclosed between a BEGIN and an END
- Action specifies the relevant modifications

INSTEAD OF triggers are used to process view modifications
An active database allows users to make the following changes to triggers (rules)
- Activate / Deactivate / Drop

An event can be considered in 3 ways
- Immediate consideration
- Deferred consideration
- Detached consideration

Immediate consideration; Part of the same transaction and can be one of the following depending on the situation
Before – After - Instead of

Deferred consideration Condition is evaluated at the end of the transaction
Detached consideration Condition is evaluated in a separate transaction

Temporal Database Concepts Time Representation, Calendars, and Time Dimensions
Time is considered ordered sequence of points in some granularity
A calendar organizes time into different time units for convenience.
Accommodates various calendars Gregorian (western)/Chinese / Islamic / Hindu / Jewish

Point events;
- Single time point event E.g., bank deposit
- Series of point events can form a time series data

Duration events
- Associated with specific time period .
- Time period is represented by start time and end time

Transaction time ; The time when the information from a certain transaction becomes valid

Bitemporal database ; Databases dealing with two time dimensions

- A single complex object stores all temporal changes of the object
- Time varying attribute An attribute that changes over time E.g., age
- Non-Time varying attribute An attribute that does not changes over time E.g., date of birth

Spatial Database Concepts; Keep track of objects in a multi-dimensional space / Maps / Geographical Information Systems (GIS) / Weather

Typical Spatial Queries

- **Range query:** Finds objects of a particular type within a particular distance from a given location
- **Nearest Neighbor query:** Finds objects of a particular type that is nearest to a given location
- **Spatial joins or overlays**: Joins objects of two types based on some spatial condition (intersecting, overlapping, within certain distance, etc.)

R-trees

- Technique for typical spatial queries
- Group objects close in spatial proximity on the same leaf nodes of a tree structured index
- Internal nodes define areas (rectangles) that cover all areas of the rectangles in its subtree.

Quad trees Divide subspaces into equally sized areas

Types of multimedia data are available in current systems

- Text: May be formatted or unformatted.
- Graphics: Examples include drawings and illustrations that are encoded using some descriptive standards
- Images: Includes drawings, photographs
- Animations: Temporal sequences of image or graphic data.
- Video: A set of temporally sequenced photographic data for presentation at specified rates
- Structured audio: A sequence of audio components comprising note, tone, duration.
- Audio: Sample data generated from aural recordings in a string of bits in digitized form.
- Composite or mixed multimedia data: A combination of multimedia data types such as audio and video which may be physically mixed to yield a new storage format or logically mixed while retaining original types and formats.

Declarative Language; Language to specify rules

Inference Engine (Deduction Machine)

- Can deduce new facts by interpreting the rules
- Related to logic programming

Speciation consists of:

Facts Similar to relation specification without the necessity of including attribute names

Rules Similar to relational views (virtual relations that are not stored)

Predicate has; a name / a fixed number of arguments

Convention: Constants are numeric or character strings

Variables start with upper case letters E.g., SUPERVISE(Supervisor, Supervisee)

A formula can have quantifiers ; Universal / Existential

In clausal form, a formula must be transformed into another formula with the following characteristics

- All variables are universally quantified
- Formula is made of a number of clauses where each clause is made up of literals connected by logical ORs only
- Clauses themselves are connected by logical ANDs only.

There are two main alternatives for interpreting rules: Proof-theoretic / Model-theoretic

Proof-theoretic

- Facts and rules are axioms
- Ground axioms contain no variables
- Rules are deductive axioms
- Deductive axioms can be used to construct new facts from existing facts
- This process is known as theorem proving

Model; An interpretation for a specific set of rules

A program is safe if it generates a finite set of facts

- Fact-defined predicates (or relations) Listing all combination of values that make a predicate true
- Rule-defined predicates (or views)

CHAPTER 28

Definitions of Data Mining

- The discovery of new information in terms of patterns or rules from vast amounts of data.
- The process of finding interesting structure in data.
- The process of employing one or more computer learning techniques to automatically analyze and extract knowledge from data.

Data Warehousing

- The data warehouse is a historical database designed for decision support.
- Data mining can be applied to the data in a warehouse to help with certain types of decisions.
- Proper construction of a data warehouse is fundamental to the successful use of data mining.

The KDD process model comprises six phases

- Data selection / Data cleansing / Enrichment / Data transformation or encoding
- Data mining / Reporting and displaying discovered knowledge

Goals of Data Mining and Knowledge Discovery (PICO)

- Prediction: Determine how certain attributes will behave in the future.
- Identification: Identify the existence of an item, event, or activity.
- Classification: Partition data into classes or categories.
- Optimization: Optimize the use of limited resources.

Types of Discovered Knowledge

Association Rules / Classification Hierarchies / Sequential Patterns / Patterns Within Time Series / Clustering

Association rules are frequently used to generate rules from market-basket data

The general algorithm for generating association rules is a two-step process.

1) Generate all itemsets that have a support exceeding the given threshold. Itemsets with this property are called large or frequent itemsets.
2) Generate rules for each itemset

Two properties are used to reduce the search space for association rule generation.

Downward Closure A subset of a large itemset must also be large

Anti-monotonicity A superset of a small itemset is also small.

This implies that the itemset does not have sufficient support to be considered for rule generation.

The Apriori algorithm was the first algorithm used to generate association rules.

The Apriori algorithm uses the general algorithm for creating association rules together with downward closure and anti-monotonicity.

Apriori algorithm https://www.youtube.com/watch?v=WgaYxdld6xQ

The sampling algorithm selects samples from the database of transactions that individually fit into memory.

The Frequent-Pattern Tree Algorithm reduces the total number of candidate itemsets by producing a compressed version of the database in terms of an FP-tree.

The algorithm consists of two steps:

Step 1 builds the FP-tree.

Step 2 uses the tree to find frequent itemsets.

FP TREE https://www.youtube.com/watch?v=nS_YP7kCO3Y&feature=youtu.be

The Partition Algorithm
- Divide the database into non-overlapping subsets.
- Treat each subset as a separate database where each subset fits entirely into main memory.

Complications seen with Association Rules
- The cardinality of itemsets in most situations is extremely large.
- Association rule mining is more difficult when transactions show variability in factors such as geographic location and seasons.
- Data quality is variable; data may be missing, erroneous, conflicting, as well as redundant.

Classification is
the process of learning a model that is able to describe different classes of data.

Clustering Unsupervised learning or clustering builds models from data without predefined classes.
The goal is to place records into groups where the records in a group are highly similar to each other and dissimilar to records in other groups.

Additional Data Mining Methods
Sequential pattern analysis
Time Series Analysis
Regression
Neural Networks
Genetic Algorithms

Sequential Pattern Analysis
- Transactions ordered by time of purchase form a sequence of itemsets.
- The problem is to find all subsequences from a given set of sequences that have a minimum support.

Time Series Analysis
- Time series are sequences of events.
- Time series analysis can be used to identify the price trends of a stock or mutual fund.
- Time series analysis is an extended functionality of temporal data management.

A regression equation
estimates a dependent variable using a set of independent variables and a set of constants.

A neural network
- is a set of interconnected nodes designed to imitate the functioning of the brain.
- Node connections have weights which are modified during the learning process.
- Neural networks can be used for supervised learning and unsupervised clustering.

Genetic learning is based on the theory of evolution.
- An initial population of several candidate solutions is provided to the learning model.

Data Mining Applications
1) Marketing Marketing strategies and consumer behavior
2) Finance Fraud detection, creditworthiness and investment analysis
3) Manufacturing Resource optimization
4) Health Image analysis, side effects of drug, and treatment effectiveness

CHAPTER 29

(شابتر نظري)

Definitions data warehouse as:
"A subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions."

Applications that data warehouse supports are:
1) **OLAP (Online Analytical Processing)**
used to describe the analysis of complex data from the data warehouse.
2) **DSS (Decision Support Systems)** also **known as EIS (Executive Information Systems)** supports organization's leading decision makers for making complex and important decisions.
3) **Data Mining** is used for knowledge discovery, the process of searching data for unanticipated new knowledge.

Data Warehouse processing involves
1) Cleaning and reformatting of data 2) OLAP 3) Data Mining

Comparison Data Warehouses with Traditional Databases
- ✓ Data Warehouses are mainly optimized for appropriate data access.
- ✓ Traditional databases are transactional and are optimized for both access mechanisms and integrity assurance measures.
- ✓ Data warehouses emphasize more on historical data as their main purpose is to support time-series and trend analysis.
- ✓ Compared with transactional databases, data warehouses are nonvolatile.
- ✓ In transactional databases transaction is the mechanism change to the database. By contrast information in data warehouse is relatively coarse grained and refresh policy is carefully chosen, usually incremental.

Characteristics of Data Warehouses
1) Multidimensional conceptual view
2) Unlimited dimensions and aggregation levels
3) Dynamic sparse matrix handling
4) Multi-user support
5) Transparency
6) Consistent reporting performance
7) Generic dimensionality
8) Unrestricted cross-dimensional operations
9) Client-server architecture
10) Accessibility
11) Intuitive data manipulation
12) Flexible reporting

The sheer volume of data is an issue, based on which Data Warehouses could be classified as follows.
Enterprise-wide data warehouses: They are huge projects requiring massive investment of time and resources.
Virtual data warehouses: They provide views of operational databases that are materialized for efficient access.
Data marts: These are generally targeted to a subset of organization, such as a department.

Advantages of a multi-dimensional model
- Multi-dimensional models lend themselves readily to hierarchical views in what is known as roll-up display and drill-down display.
- The data can be directly queried in any combination of dimensions, bypassing complex database queries

Multi-dimensional schemas are specified using:
Dimension table : It consists of tuples of attributes of the dimension.
Fact table : Each tuple is a recorded fact. This fact contains some measured or observed variable (s) and identifies it with pointers to dimension tables

Two common multi-dimensional schemas are
Star schema: Consists of a fact table with a single table for each dimension
Snowflake Schema: It is a variation of star schema, in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them

Fact Constellation is a set of tables that share some dimension tables.
Indexing ; Data warehouse also utilizes indexing to support high performance access.

The Design of a Data Warehouse involves following steps.

- Acquisition of data for the warehouse.
- Ensuring that Data Storage meets the query requirements efficiently.
- Giving full consideration to the environment in which the data warehouse resides

Acquisition of data for the warehouse

- The data must be extracted from multiple, heterogeneous sources.
- Data must be formatted for consistency within the warehouse.
- The data must be cleaned to ensure validity
- Difficult to automate cleaning process
- Back flushing, upgrading the data with cleaned data

Functionality of a Data Warehouse

- Functionality that can be expected:
- Roll-up: Data is summarized with increasing generalization
- Drill-Down: Increasing levels of detail are revealed
- Pivot: Cross tabulation is performed
- Slice and dice: Performing projection operations on the dimensions.
- Sorting: Data is sorted by ordinal value.
- Selection: Data is available by value or range.
- Derived attributes: Attributes are computed by operations on stored derived values.

Warehouse vs. Data Views

Views and data warehouses are alike in that they both have read-only extracts from the databases.
However, data warehouses are different from views in the following ways:

- Data Warehouses exist as persistent storage instead of being materialized on demand.
- Data Warehouses are not usually relational, but rather multi-dimensional.
- Data Warehouses can be indexed for optimization.
- Data Warehouses provide specific support of functionality.
- Data Warehouses deals huge volumes of data that is contained generally in more than one database.

- **مُلخص مادة الـ داتا بيس مانمجمنت سستم..**
- **يغطي 90% من المادة..**
- **من صفحة 1 الى 5 جُزئية الميد تيرم ..**
- **ومابعدها الى النهاية جزئية الفاينل ..**
- **الاجزاء العملية موجود تحتها روابط لفديوهات تساعدكم بفهم نفس الفكرة او مثال مُقارب لها ..**
- **لا تهملوا الجزئية العملية ابداً احياناً يكون منها اللونق انسر ..**
- **أي عنوان بالهايلايت الاصفر مُكرر بفاينل او ميد سابق ..**
- **مُمكن اكون مو منتبهة على جزئية عملية وما حطيتها ضروري تتأكدوا من كل شابتر تذاكروه انو مخلصين العملي ..**
- **ارجعوا مروا على السلايدات من باب التأكيد ..**
- **هذا رابط تجميع اسئلة فاينل سابق ..**

https://drive.google.com/file/d/0B1Y8gE_ocJiOR3E2VkJiU0lJc1E/view

كُل التوفيق .. الله يرزقنا واياكم الدرجات العالية في الدنيا والأخرة ..
تُقى أكراد ..