Database "Exam Notes

For this to work , people should post different questions (past papers).

There should be different answers for questions, as everyone is not going off one person’s answers.

REMEMBER TO PUT ANSWERS IN RED

Green for not 100% sure

Blue for another question

2012 DB Exam - <http://exampapers.cit.ie/PastExams/Computing/KWEBD_8_Y3/2013%20Semester1/KWEBD_8_Y3%20COMP8017%20Database%20Systems%203.pdf>

2013 DB Exam -

<http://exampapers.cit.ie/PastExams/Computing/KWEBD_8_Y3/2014%20Semester1/KWEBD_8_Y3%20COMP8017%20Database%20Systems%203.pdf>

**Useful short video on db admin and transaction logs**

https://www.youtube.com/watch?v=rPfncf8W1MA

**Exam Breakdown**

***5 questions do 4 [2 hours]***

**Q1 & 2** - Database Systems: General terms and concepts

**Q3** - Recovery

**Q4** - Concurrency

**Q5** - Security

**Database Recovery Questions**

(Taken from the recovery document on Blackboard)

1. **Explain the need for recovery/concurrency in the database environment**

Concurrency is the ability of a database to allow multiple users to affect multiple transactions. This is one of the main properties that separates a database from other forms of data storage like spreadsheets.

Data recovery is the process of restoring data that has been lost, accidentally deleted, corrupted or made inaccessible for some reason.

1. **Describe the main components of a recovery system**

In data management, recovery is a process that involves copying backup files from secondary storage (tape, Zip disk or other backup media) to hard disk.

1. **Explain the term transactions; what are they, how do complicate the work of the DBMS**

A transaction, in the context of a database, is a logical unit that is independently executed for data retrieval or updates. In relational databases, database transactions must be atomic, consistent, isolated and durable--summarized as the ACID acronym.

1. **Explain the need for and use of logs, checkpoints and archive databases.**

In the field of databases in computer science, a **transaction log** is a history of actions executed by a database management system to guarantee ACID properties over crashes or hardware failures.

A *checkpoint* creates a known good point from which the SQL Server Database Engine can start applying changes contained in the log during recovery after an unexpected shutdown or crash.

Data archiving is the process of moving data that is no longer actively used to a separate data storage device for long-term retention.

1. **Develop/Devise recovery procedures (Business Continuity)**

?

1. **Why is block I/O used and why is it asynchronous?**

Block I/O is ultimately what is done to a disk drive. Disks today in open systems store data in fixed blocks (usually 512 bytes) in a linear address space. Doing block I/O means that the application or file system is sending blocks to the disk drive to be written or asking for blocks using a logical block address (LBA). File systems turn file requests into block I/O. Applications (including databases) can do file I/O or they can bypass the filesystem and do block I/O (this is usually called raw I/O). Obviously it's easier to do file I/O, and you can do file sharing much easier that way. Doing block I/O may have performance advantages (in control of the buffering/caching and not having the file system overhead).

In telecommunications, **asynchronous** communication is transmission of **data**, generally without the use of an external clock signal, where **data** can be transmitted intermittently rather than in a steady stream.

For raid : http://www.tutorialspoint.com/dbms/dbms\_storage\_system.htm

**Recovery Qs**

1. **What is meant by reliability?**

* Reliability means that the system is as close to constantly available as is possible.

1. **Differentiate using examples between the terms “error” and “failure” in the context of database recovery.**

* error: explicit ROLLBACK is planned and not considered a failure
* failure: aborts any suspect transactions and effects recovery

1. **What is a backward recovery scheme, and in what applications can it be used?**

* The database is reset to some previous correct state.
* Applications that are subject to accounting or security audits

1. **State the objective of a recovery system? Briefly discuss the main factors influencing the recovery scheme?**

* must handle both single process and full system failures.

1. **Why are transactions the unit of recovery / concurrency and not individual user operations?**
2. **Write a note on the properties of a transaction?**

"A series of data manipulation statements that must either **fully complete** or **fully fail**, **leaving the database in a consistent state**"

* **ACID** (**Atomicity**, **Consistency**, **Isolation**, **Durability**) is a set of properties that guarantee that database transactions are processed reliably.
* **Atomic** (if the change is committed, it happens in one fell swoop; you can never see "half a change")
* **Consistent** (the change can only happen if the new state of the system will be valid; any attempt to commit an invalid change will fail, leaving the system in its previous valid state)
* **Isolated** (no-one else sees any part of the transaction until it's committed)
* **Durable** (once the change has happened - if the system says the transaction has been committed, the client doesn't need to worry about "flushing" the system to make the change "stick")

1. **What is the WALS and why is it used.**

The WALS states that the log is written and saved to stable 2nd secondary storage before any updates are made to the actual physical database.

This is crucial as log data itself **may be** built up in buffers (or disk caches). If a failure to volatile storage occurs then this log data will be lost

1. **What are checkpoints.**

A checkpoint is a point in recent time where we guarantee that any operation that is in the log is also written out to the physical database.

Checkpoints therefore are used to limit the volume of log information that has to be processed in the event of a loss of volatile memory

Checkpoints force writes any buffers (both for writes to the database and the log); so you can always come back to this point in time to start recovery in the event of failure

Checkpoints are done frequently during processing (every few minutes).

1. **What is meant by archive/backup database management**?

## Is the process of moving data that is no longer actively used to a separate data storage device for long-term retention.

1. **What is meant by archive/backup log management?**

**Log Management:** in case of failure, active transactions **must** be capable of being rolled back without human intervention.

Therefore all log records for all active transactions must be in **a single** on-line log (NB). When the current on-line log is full, the log manager switches to another active log, and dumps the first to archive. Thus the second active log should be opened as soon as the first reaches a set threshold e.g. 80%. Existing transactions use the first log, new transactions use the second.

1. **Explain the concepts of Undo/Redo recovery.**

UNDO: rectifies undesirable changes made by a transaction. Requires Before\_value in log

REDO: ensures that changes made by a transaction are correctly recorded by redoing the action(s). Requires the After\_value in the log.

1. **Non-volatile memory failure requires archival databases and archival logs. Explain.**
2. **Besides the transaction recovery log, what other logs are implemented by DBMS? What is a binary log?**

**Bin Log**

* The binary log contains “events” that describe database changes such as table creation operations or changes to table data.

**General Log**

* The general query log is a general record of what mysql is doing. The server writes information to this log when clients connect or disconnect, and it logs each SQL statement received.

**Slow Query Log**

* The slow query log consists of SQL statements that took more than [long\_query\_time](http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar_long_query_time) **(in the ini file)** seconds to execute. The minimum and default values are 1 and 10, respectively.

1. **Explain in detail the use of RAID in recovery (for) a) database data; b) control data; c) log data.**
2. **Discuss Recovery requirements impact on an N-Tier system architecture**
3. **Devise/Outline a recovery plan (Business Continuity plan)**
4. **Explain using an example how the use of transactions complicates the recovery system.**
5. **Explain using an example how the use of SLAs impacts on the recovery proc**ess.

<http://www.cio.com/article/2432490/data-management/understanding-service-level-agreements-for-database-development.html>

1. **Explain how the use of Virtualisation impacts on Recovery/**

**This is the basics for it**

Virtualisation:

Note that although **virtualisation** is a single point of failure at the physical machine level, it provides benefits for recovery in the form of containment of failures at OS, DBMS and Application levels.

downtime is significantly reduced for maintenance and system admin

downtime for recovery as the entire system configuration is now in digital form and simply treated as a critical object for protection. This complete running environment can be deployed anywhere quicker than physical install/config/data recovery. In other words the time to manually set up and config (very slow mechanical process) is replaced by a simple electronic copy and run.

all this for one answer?

this is the answer

Recovery log are processed in sequential read/write. Therefore a sequential data organisation on the disk would seem beneficial. However striping across volumes/disks forces disk head seeks (not good); in addition a full mirror is ok so long as you do not mix multiple log files on it as this again will force disk seeks to read/write the different logs.

1. **Explain how the use of RAID impacts on Recovery/ Concurrency**

As unsuccessful, failed user remedies begin to stack up, the time and expense of professional data recovery goes up right along with them, while the statistical prognosis for complete recovery goes in the opposite direction.

Failed rebuild attempts are lamentable and make successful data recovery more difficult and potentially impossible.

1. **Storing the query texts in the security log may be sufficient however it is not enough for transaction recovery? Explain**

Database Architecture

**Discuss the link between memory management and concurrency (multiple users accessing shared data).**

1. An in-memory database (IMDB; also main memory database system or MMDB or memory resident database) is a database management system that primarily relies on main memory for computer data storage. It is contrasted with database management systems that employ a disk storage mechanism.
2. Concurrency is the ability of a database to allow multiple users to affect multiple transactions. This is one of the main properties that separates a database from other forms of data storage like spreadsheets.

**Explain what is meant by load balancing**

Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster.

“Load balancing is a core networking solution responsible for distributing incoming traffic among servers hosting the same application content.”

-so its just giving even distribution of queries between multiple servers

**Describe the term hit rate and comment on it as a factor in performance.**

The chief measurement of a cache, which is the percentage of all accesses that are satisfied by the data in the cache. Also known as "hit ratio.”

“The term **hit rate** refers to the proportion of successful accesses in the total number of accesses on a particular item of storage. It is a critical indicator of performance. Successful access means that the required data was already available in a cache when it was requested. If you store anything in a cache/buffer, to justify its position there it should have a high hit rate. Good hit rates = good performance.”

**Storing Query data (or QEP) may be useful for improved throughput, explain?**

Storing query data can help with optimising as if there is a very commonly executed query it'll have the data stored so the access time will be quicker in cache

**Describe how CPU scheduling and memory allocation are linked.**

When a process/thread is executing, it uses RAM for the data that it is processing. Therefore, we should note, that **threads and memory allocation are linked**. A thread based system is efficient as it conserves global memory for the overall process, as threads are swapped. In other words, the parent process can manage shared memory between its threads. So in general, memory management is better and the context switching for threads are lower cost while maintaining the same parent address space and control. Because of this threads are called ‘light weight’,

scheduling is the method by which threads, processes or data flows are given access to system resources

Threads under one parent can share common data

**A DBA (**[**Database administrator**](http://en.wikipedia.org/wiki/Database_administrator)**) faced with a choice of optimizer type faces a dilemma, explain?**

When an SQL statement is run, the DBMS must decide on a query execution plan (QEP) to implement that SQL. The DBMS may run an optimizer program to implement the SQL statement in an efficient way i.e. it develops the QEP.

**Explain how the same query may have different executions (or costs of executing) .**

If the execution plan was the same and one was slow, it would be database load, hardware, locking/blocking, etc.

However, if the execution plans are different something is different between the two databases.

**Explain, using an example, the basis of query optimisation.**

Query optimization is a function of many relational database management systems. The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans.

**example:**

*Select \* FROM College DB where department =”business”* will result in a different QEP than *department=”computing”* because the number of records for each is different. So we would use an index on computing but NOT business

**What is meta data, and why is accessing meta data in the data dictionary critical for system performance?**

Meta data is “data about data”. An item of metadata describes the specific characteristics about an individual data item. In databases, metadata describes the structural components of tables and their elements. For example, metadata about an element could include data types, name of data, size and many more characteristics about that element. It would also give information about the tables the database is storing, information, such as length of fields, number of columns, where the tables are located and other pertinent information. One of the main uses for met data is to provide a link between the information creator and the information users. Meta data allows the users to speed up the search for individual data. This is done by being able to set parameter for searches.

**What is an SQL dialect? Why it is important for issues such as portability and data migration?**

The dialect of the database is simply a term that defines the specific features of the SQL language that are available when accessing that database.

**Explain using examples why a DBA must know about software developer (programmer) requirements during install/config.**

DBMS may have a setting that relates to the version of SQL supported. Many older programs have code that does not adhere to significant SQL standard releases. The administrator must liaison/agree with software developers that all programs that interact with the database adhere to a SQL standard before configuring/installing the DBMS.

**Describe an N-tier architecture for a database system? IS this recommended in all cases?**

**A three tier architecture is effectively spreading the overall processing load** over 3 machines. Any tier can be expanded but in data intensive applications the data tier is commonly more complex. This an N-tier architecture. Within the data server element of the overall architecture you may for instance be able to run multiple instances of the database on the same machine (or on different machines). Each instance can be configured specifically for the particular data it stores. This is moving into the area of distributed databases (note the terms federated databases, clouds or clustered databases relating to a co-ordinated set of data servers).

**What is thread and what is its role in making a database robust ( or fault tolerant) (or load balanced)**

Threads also enable a more robust system essential where demands of high availability are becoming a standard feature of many business systems. Usually this demand is prompted by web interfaces for transaction systems such as sales, ordering etc. Reliability means that the system is as close to constantly available as is possible.

**In a database there is far more data that can fit in RAM (processing memory); Describe how ‘good’ memory management is essential for efficient DBMS.**

http://en.wikibooks.org/wiki/Design\_of\_Main\_Memory\_Database\_System/Overview\_of\_DBMS

**Explain why multiple instances of a database might be used?**

Multiple instances against single (same) database is required in such environment where that database is distributed among different systems (either on LAN or WAN) forming cluster so users can access the data transparently. Each instance can be configured specifically for the particular data it stores.

Different types of applications require different configurations of the database to run efficiently. Rather than tuning the single database to give acceptable performance to all applications, one solution is to install multiple instances of the DB server, configuring each specifically to suit a limited subset of applications. Multiple instances are used for load balancing and targeted configuration for optimum performance; Instances can also be used in development and test environments.

**Explain what is the role of the administrator in Data/File Management on 2nd storage devices.**

In MySql these are called storage engines, but in general they are called file organisations. A database administrator might find it useful to change the file organisation and process the table in a different way i.e re-organise the table and therefore process it using a different internal subprogram e.g. change from a Btree to a Hash file

**Why not create indexes on all attributes? Should a QEP always use an index if available?**

Reason you don't use indexes for every table is your index is suppose to be for quick access

If you have every table indexed its the same size as the amount of data you got

Data Migration/Cleansing/Portability

**Describe and explain the following terms:**

**Data migration** is the process of transferring data between storage types, formats, or computer systems. It is a key consideration for any system implementation, upgrade, or consolidation.

**Data import** , Allowing you to import tables and database to your dbma system.

**Data export** allowing you to export your database and tables in excel,csv and sql files. It is vendor locked.

**Data cleansing,** data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database.

**data loading/unloading,**

|  |  |
| --- | --- |
| **Unloading** | **Copying database data to external text files for consumption by another database or another application (such as a spreadsheet application). The text files are in an industry-standard format such as tab-delimited or comma-delimited (CSV).** |
| **Loading** | **Copying data into the database from external text files that are in either a standard delimited format or in any of the formats that are supported by the SQL\*Loader utility.** |

**Database portability** Portability and data migration are closely related. SQL code that is portable means that it’ll work on any DBMS, so you can port or migrate your programs to a new system without encountering SQL dialect errors.

**legacy system** , In the general context, it can refer to any of the older database technologies. In a more specific context, it can refer to a database system that was inherited by a team from previous project owners.

**Data extraction** is the act or process of retrieving data out of (usually unstructured or poorly structured) data sources for further data processing or data storage (data migration).

**ETL**, In computing, Extract, Transform and Load (ETL) refers to a process in database usage and especially in data warehousing that: Extracts data from homogeneous or heterogeneous data sources Transforms the data for storing it in proper format or structure for querying and analysis purpose .Loads it into the final target (database, more specifically, operational data store, data mart, or data warehouse)

2013 Exam Paper

**Explain the principle components of a data import utility**

Command default values.

Character set used in the data file (source) and the database (destination). A Given DBMS e.g. MySql may not support all character sets. Windows ASCII, Unicode, Vendor specific.

Source/Destination specification e.g. path names, any restrictions on where data can be loaded from (source) etc.

Effects of performance; impact on concurrent transactions (reader, writers). This implies that the destination database / table already exists and contains data.

Option to delay execution of the import to a quiesced state of the database.

Need flexibility in handling various text formats/structures. Fields separated by commas, spaces, tabs. Lines ended by carriage return, carriage return & line feed, special control characters e.g. control Z.

Ability to skip file headers e.g. the column names in a spreadsheet, if so we may need to be able to skip the header. Note the header would confuse any format/structure analysis tool.

Ability to process data as it is imported e.g. calculate, skip fields, assign to specific table column names

Error handling: Errors may occur during the import process e.g. end of line encountered (i.e. not enough text in the input line); too much data on a line for the destination table definition. Blanks and Null values.

Security implications: privileges required to execute the import command (run the utility). Irrespective of the assigned privileges to users, the entire DBMS may be set up to allow/disallow data import (this would over ride the user privilege grants)

**Give examples of the application of data migration**

1. **Data migration** is the process of transferring **data** between storage types, formats, or computer systems. It is a key consideration for any system implementation, upgrade, or consolidation.

**Perform a data migration to include a data cleanse & data import**

**Explain, using an example, what concatenated fields in a database means, and comment on why concatenated fields are an issue for the data migration.**

Concatenation, in the context of databases, refers to the joining together two or more things into a large one. In database parlance, the things being joined are generally two table fields which may be from the same or different tables.

In relational database system design, we do not need concatenated values. W**e can break up the composite value and store the elements in separate columns. We then can create a composite key using the attributes themselves.** The database designer must decide whether to keep the composite codes in one column (design A below) or break up the elements into separate attributes using string functions, and import the set of attributes into the database (Design B below). This is an example of a transformation that could be programmed for regular migration.

**Explain the term multi-threading.**

1. a technique by which a single set of code can be used by several processors at different stages of execution.

**Explain the fundamental techniques used to implement SQL joins efficiently**

### Join Indexes - did this in labs

### Join algorithms

Three fundamental algorithms for performing a join operation exist: nested loop join, sort-merge join and hash join.

**Stored Procedures**

Stored Procedures are a batch of SQL statements that can be executed in a couple of ways. Most major DBMs support stored procedures; however, not all do. You will need to verify with your particular DBMS help documentation for specifics. As I am most famillar with SQL Server I will use that as my samples

In a database management system ([DBMS](http://searchsqlserver.techtarget.com/definition/database-management-system)), a stored procedure is a set of Structured Query Language ([SQL](http://searchsqlserver.techtarget.com/definition/SQL)) statements with an assigned name that's stored in the database in compiled form so that it can be shared by a number of programs. The use of stored procedures can be helpful in controlling access to data ([end-users](http://whatis.techtarget.com/definition/end-user) may enter or change data but do not write procedures), preserving [data integrity](http://searchdatacenter.techtarget.com/definition/integrity) (information is entered in a consistent manner), and improving productivity (statements in a stored procedure only need to be written one time).

^^ benefits = security (SQL injection attacks) + stored procedures can be fully optimized for fast execution + compiled

2013 Exam Paper

**Q1 Database Systems: General terms and concepts**  **25 Marks**

**Indexing and Index Management**

**Explain why you would not recommend creating indexes on every attribute in a table?**

**(12 Marks)**

Index is an extra cost in space and maintenance.

It is supposed to be a fast access of data but if you have an index on every attribute it is the same size as the amount of data you get.

**Explain why a Query processor/optimiser may not opt to use an index on a column even though it is available? Mention the role of Statistics.**

**(13 Marks)**

The role for the optimiser when making the QEP might be:

If an index on Dept column exists then

Use Index

If you force a QEP to use an index and you end up reading a large percentage of the records of the table in from the disk, then the cost of reading both, the index and the data file becomes a performance issue due to excessive disk seek.

**Q2. Database Systems: General terms and concepts 25 Marks**

**Threads**

**Explain the term multi-threading.**

**(13 Marks)**

Multi-threading relates to CPU scheduling. A thread is a management unit of work for running on the CPU. Let’s say you have 3 processes running on the computer: an email client, an internet browser and a DBMS. All three have equal importance so they are all scheduled for CPU time equally.

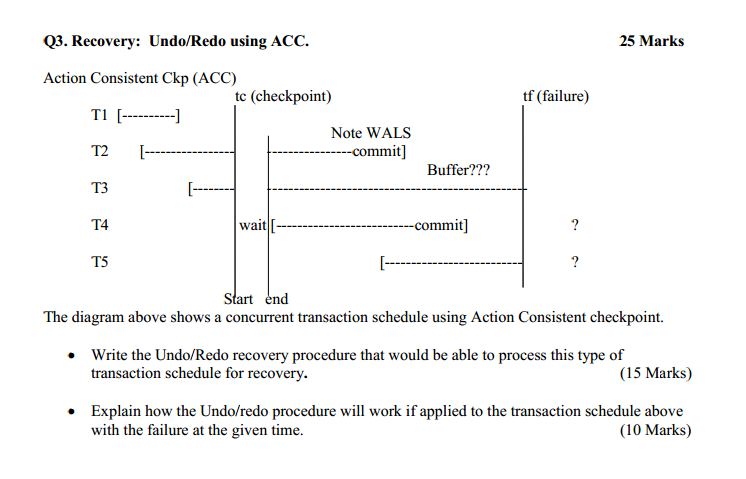
However, what if one of these processes is very important and it has a lot of work to do?

How can you allocate more CPU time to it? Threads allow you do just that. The system administrator can configure the system so that the DBMS is allocated more threads than the others e.g. we give 3 threads to the DBMS. The CPU is then managed by scheduling the threads not the full process. So the DBMS will get 3 times more CPU than each of the others ( Email, Browser)

**Explain how multi-threading is of significance to DBMS architecture e.g. load balancing, availability.**

Load balancing - allows many users/transactions to be served at the same time/simultaneously avoiding queuing for CPU time by spreading the work over more cores. Processes with higher priority don’t block lower priority processes.

8



**Recovery Procedure (Undo/Redo)** (general recovery procedure; buffers independent, ACC)

Using the above example: at restart time the following recovery procedure is used.

1. retrieve the last checkpoint record

2. create two lists called UNDO and REDO. Set UNDO equal to the list of all transactions given in the checkpoint record.

3. search through the log from the checkpoint time forward.

4. If a BEGIN TRANSACTION log entry is found for transaction T, add T to the UNDO list.

5. If a COMMIT entry is found for transaction T, move T from UNDO to REDO.

Note : ROLLBACK and implications.

6. When the end of the log is reached both list identify the transactions that require partial changes to be undone and unpropagated modifications for committed transactions(i.e. not logged) to be redone.

In the above example T3 and T5 must be undone. T2 and T4 redone.

**Q4. Concurrency:**

**Data Access Protocol(DAP): There are two versions of the DAP, one basic or weak, and the second strong.**

**a)State the basic DAP**

1. before reading you must acquire an S lock on the object. For update you must acquire an X lock. **Lock Promotion** : to cover the common case of ‘look first then update’, you can acquire the S lock and then **promote** the S to an X lock. (never hold S and X for the same object as X subsumes S)

2. if a request of any type cannot be granted for incompatibility reasons then the requesting transaction will go into a WAIT state until the request can be granted

3. commit or rollback releases all locks held by a transaction.

**b)Explain using an example how this basic DAP is weak in terms of concurrency? That is, devise concurrent transactions to show that database integrity can be compromised if the weak version of the DAP is used**

The Uncommitted Dependency Problem & locking

The original problem was that TA2 was allowed see the value of 10 which was a partial updated value from TA1 i.e. the value is unstable as it is subject to possible failure. For locking to work we must prevent the read by TA2 until the value of cost is stable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Time** | **TA1** | **TA2** | **Cost** |
| **t1** | **begin trans** |  | **20** |
| **t2** | **read(cost) S lock OK** |  | **20** |
| **t3** | **Update cost=cost-10 X lock OK** |  | **20** |
| **t4** | **write(cost)** | **begin trans** | **10** |
| **t5** |  | **read(cost) S lock No** | **10** |
| **t6** | **rollback** | **wait** | **20** |
| **t7** |  | **S lock granted Read(cost)** | **20** |
| **t8** |  | **commit** | **20** |

**c) Explain how the strong version of DAP prevents the problem(s) outlined in part b above**

Holding locks until commit/rollback ensures that any other transaction cannot read value (uncommitted dependency/inconsistent analysis) or update value (lost update) before actual one finishes its purpose.

**Q5. Security: 25 Marks**

**Discuss Authentication of users for Access Control ( types and effectiveness)**

**(17 Marks)**

Something we know: Hard to steal unless improperly stored. Effectiveness increases with password change policy in place(password time expiry, complexity, usage of different passwords)

Passwords: Too complicated password can lower productivity if used in wrong environment(e.g. password protected sale terminals in restaurant)

Passphrase:

Something we are: Hard to replicate. Not easily changed if compromised.

Retina scan:

Voice command:

Finger print:

Something we have: Can be stolen. Easy to put implement and withdraw(removing items from past employees).

Swipe cards:

Physical key:

**Proliferation of rights**

**(8 Marks)**

Ownership vs Admin

Owner is usually the person who creates the data

*Also gives access rights to others which is not always a good idea as* ***too many*** *people get access to the database.*

Has every type of access

Ownership is very hard to control particularly if the owner of data can pass on access rights to others. This issue is known as the proliferation of rights control.

**Examinations 2012/2013**

**The main features of MySQL**

* **Written in C and C++. Tested with a broad range of different compilers.**
  + can be run on multiple OS
* **The MySQL server design is multi-layered with independent modules.**
  + allows for installing modules as required, saving space and performance
* **Fully multi-threaded using kernel threads.**
  + improved performance, allows for fault tolerance(independent threads), load balancing
* **The server is available as a separate programs for use in a client/server networked environment. It is also available as a library that can be embedded into standalone i.e. used in isolation or in environments where no network is available.**
  + Functionality can be implemented into different application(using libraries) without the need of setting up full system.

**Q.1. Explain the features of MySQL listed above.**

**Q.2. Concurrency**

1. **“Locks should be always maintain until end of transaction” Discuss your opinion of this statement.**

This is the weak version of Data Access Protocol. The lock should be maintained because if the locks are released before the end(commit/rollback) the other transaction may try to access/chan ge the same value. That would cause inconsistency in the value between these transactions.

The locks should be maintain.

Properties of locks

* If a transaction 1 has an X lock on a tuple p, a request of any type from any transaction 2 will be denied
* If T1 has an S lock on p then requests from T2 for an :
  + S lock will be granted
  + X lock will be denied

1. **What is lock granularity ? Why is it of significance in concurrency control ?**

It deals with the cost of implementing locks depending upon the space and time.

Space refers to data structure in DBMS for each lock.

Time refers to handling of lock request and release.

A too-fine granularity will increase the frequency of locks requests and locks releases which will add additional instructions.

**Granularity hierarchy (parent-child lock hierarchy)**

DB -----high level lock; simple admin and conflict check but low concurrency

Table

Table/tuple in notes but open to any 2 levels of hierarchy.

Tuple

Field------ expensive admin and compatibility check but high concurrency

**Trade-off:** the finer the granularity of the locks, the more concurrency; but the coarser (higher level) the fewer locks and therefore less admin/overhead but less concurrency.

The admin cost relates to storing and processing the current set of locks on the database. In a database with tables of millions of rows this is considerable work. In addition the issue is further complicated each time new lock requests are made. The DBMS must have some way to check if the new lock request conflicts with any existing locks. For example, if a transaction wants to X lock an entire table;

1. **Lock escalation and lock promotion**

Lock escalation : accumulates locks as you use them but trade in when you hit the pre-set limit.

To escalate means to trade in a set of lower level child locks for 1 parent lock.

Lock promotion : to cover the common case of “look first then update”, you can acquire the S lock and then promote the S to an X lock.

**Q.3. Recovery**

1. **As the DBA, you are responsible for the location and size of the on-line log. In your opinion what are the issues that influence your decisions in these matters.**

The on-line log is only for recovery for current transactions ; the Disk is safe but some or all active transactions have failed.

Usually held on direct access storage device for :

* Speed : low recovery time
* Selective e.g. an individual current transaction may fail and require rollback

Log size : log stored in a file, can’t have infinite file size ; no use of file extends.

-How often do you perform checkpoints

-How many users and size of traffic

**b. Checkpoints**

**As the DBA, setting the frequency of checkpoints is an important issue. Explain**

-availability(more frequent checkpoints, the less available db)

-less frequent checkpoints mean more space needs to be allocated for logs

-less frequent checkpoints mean more complex recovery

**Describe two mechanisms by which the frequency of the checkpoints might be controlled.**

TCC:checkpoint is done when no transaction is happening. Not good for busy applications or long transactions

-ACC: allows the checkpoint to be done in the middle of transaction but it must hold undo information for them.