Further Aspects of Database Management

(A.3)

Database Administrator (DBA) (A.3.1)

The role may include capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recovery. (wikipedia)

Duties:

- Installing / upgrading database application
- Modifying database structure
- Controlling and monitoring user access to the database
- Planning for backup and recovery of database information

https://en.wikipedia.org/wiki/Database_administrator

Database Administrator ctd. (A.3.1 ctd)

- Control development company's database
- Control D.B. access
- Work with Information Systems Managers to customize database solutions
- Debugging code
- Managing applications
- Securing organizational data, restoring lost data, creating new user permissions, testing modifications, merging databases, performance tuning
- Monitor database systems to ensure efficient, error-free functioning

Database administrators control the development of a company's databases to keep vital data available only to users with authorized access. DBAs work closely with information systems managers to customize database solutions to corporate needs. System DBAs oversee technical aspects of database administration, including debugging code and upgrading software. Application DBAs focus on managing a specific application working with the database. Typical daily duties for database administrators are securing organizational data, restoring lost data, creating new user permissions, testing modifications, merging old databases, and conduct performance tuning support. It's the responsibility of DBAs to continually monitor their database systems to ensure efficient, error-free functioning.

End-user interaction (A.3.2)

Queries

User interface

QBE (Query by example)

Visual Queries

Natural Language Interfaces

https://www.activequerybuilder.com/

Natural Language Interfaces to Database is a type of database interface that allows the user to access the data using natural language.
https://github.com/machinalis/quepy

Database Recovery (A.3.3)

Causes of errors in database:

- User error
- Statement failure
- Process Failure
- Network Failure
- Database Instance Failure
- Media (Disk) Failure

User Error
A database administrator can do little to prevent user errors (for example, accidentally dropping a table). Usually, user error can be reduced by increased training on database and application principles. Furthermore, by planning an effective recovery scheme ahead of time, the administrator can ease the work necessary to recover from many types of user errors.

Statement Failure
Statement failure occurs when there is a logical failure in the handling of a statement in an Oracle program. For example, assume all extents of a table (in other words, the number of extents specified in the MAXEXTENTS parameter of the CREATE TABLE statement) are allocated, and are completely filled with data; the table is absolutely full. A valid INSERT statement cannot insert a row because there is no space available. Therefore, if issued, the statement fails.

If a statement failure occurs, the Oracle software or operating system returns an error code or message. A statement failure usually requires no action or recovery steps; Oracle automatically corrects for statement failure by rolling back the effects (if any) of the statement and returning control to the application. The user can simply re-execute the statement after correcting the problem indicated by the error message.

Process Failure
A process failure is a failure in a user, server, or background process of a database instance (for example, an abnormal disconnect or process termination). When a

process failure occurs, the failed subordinate process cannot continue work, although the other processes of the database instance can continue.

The Oracle background process PMON detects aborted Oracle processes. If the aborted process is a user or server process, PMON resolves the failure by rolling back the current transaction of the aborted process and releasing any resources that this process was using. Recovery of the failed user or server process is automatic. If the aborted process is a background process, the instance usually cannot continue to function correctly. Therefore, you must shut down and restart the instance.

Network Failure

When your system uses networks (for example, local area networks, phone lines, and so on) to connect client workstations to database servers, or to connect several database servers to form a distributed database system, network failures (such as aborted phone connections or network communication software failures) can interrupt the normal operation of a database system. For example:

A network failure might interrupt normal execution of a client application and cause a process failure to occur. In this case, the Oracle background process PMON detects and resolves the aborted server process for the disconnected user process, as described in the previous section.

A network failure might interrupt the two-phase commit of a distributed transaction. Once the network problem is corrected, the Oracle background process RECO of each involved database server automatically resolves any distributed transactions not yet resolved at all nodes of the distributed database system. Distributed database systems are discussed in Chapter 33, "Distributed Databases".

Database Instance Failure

Database instance failure occurs when a problem arises that prevents an Oracle database instance (SGA and background processes) from continuing to work. An instance failure can result from a hardware problem, such as a power outage, or a software problem, such as an operating system crash. Instance failure also results when you issue a SHUTDOWN ABORT or STARTUP FORCE command.

Recovery from Instance Failure

Crash or instance recovery recovers a database to its transaction-consistent state just before instance failure. Crash recovery recovers a database in a single-instance configuration and instance recovery recovers a database in an Oracle Parallel Server configuration.

Recovery from instance failure is automatic. For example, when using the Oracle Parallel Server, another instance performs instance recovery for the failed instance. In single-instance configurations, Oracle performs crash recovery for a database when the database is restarted (mounted and opened to a new instance). The transition from a mounted state to an open state automatically triggers crash recovery, if necessary.

Crash or instance recovery consists of the following steps:

Rolling forward to recover data that has not been recorded in the datafiles, yet has been recorded in the online redo log, including the contents of rollback segments. This is called cache recovery.
Opening the database. Instead of waiting for all transactions to be rolled back before making the database available, Oracle allows the database to be opened as soon as cache recovery is complete. Any data that is not locked by unrecovered transactions is immediately available.
Marking all transactions system-wide that were active at the time of failure as DEAD and marking the rollback segments containing these transactions as PARTLY AVAILABLE.
Rolling back dead transactions as part of SMON recovery. This is called transaction recovery.
Resolving any pending distributed transactions undergoing a two-phase commit at the time of the instance failure.
As new transactions encounter rows locked by dead transactions, they can automatically roll back the dead transaction to release the locks. If you are using Fast-Start Recovery, just the data block is immediately rolled back, as opposed to the entire transaction.

Additional Information:
See the Oracle8i Parallel Server Setup and Configuration Guide for a discussion of instance recovery.

See Oracle8i Tuning for a discussion of instance recovery tuning.

Media (Disk) Failure
An error can arise when trying to write or read a file that is required to operate an Oracle database. This occurrence is called media failure because there is a physical problem reading or writing to files on the storage medium.

A common example of media failure is a disk head crash, which causes the loss of all files on a disk drive. All files associated with a database are vulnerable to a disk crash, including datafiles, online redo log files, and control files.

The appropriate recovery from a media failure depends on the files affected.

Additional Information:
See the Oracle8i Backup and Recovery Guide for a discussion of recovery methods.

How Media Failures Affect Database Operation
Media failures can affect one or all types of files necessary for the operation of an Oracle database, including datafiles, online redo log files, and control files.

Database operation after a media failure of online redo log files or control files depends on whether the online redo log or control file is multiplexed, as recommended. A multiplexed online redo log or control file simply means that a second copy of the file is maintained. If a media failure damages a single disk, and you have a multiplexed online redo log, the database can usually continue to operate without significant interruption. Damage to a non-multiplexed online redo log causes database operation to halt and may cause permanent loss of data. Damage to any control file, whether it is multiplexed or non-multiplexed, halts database operation once Oracle attempts to read or write the damaged control file (which happens frequently, for example at every checkpoint and log switch).

Media failures that affect datafiles can be divided into two categories: read errors and write errors. In a read error, Oracle discovers it cannot read a datafile and an operating system error is returned to the application, along with an Oracle error indicating that the file cannot be found, cannot be opened, or cannot be read. Oracle continues to run, but the error is returned each time an unsuccessful read occurs. At the next checkpoint, a write error will occur when Oracle attempts to write the file header as part of the standard checkpoint process.

If Oracle discovers that it cannot write to a datafile and Oracle is archiving the filled online redo log files, Oracle returns an error in the DBWn trace file and takes the datafile offline automatically. Only the datafile that cannot be written to is taken offline; the tablespace containing that file remains online.

If the datafile that cannot be written to is in the SYSTEM tablespace, the file is not taken offline. Instead, an error is returned and Oracle shuts down the instance. The reason for this exception is that all files in the SYSTEM tablespace must be online in order for Oracle to operate properly. For the same reason, the datafiles of a tablespace containing active rollback segments must remain online.

If Oracle discovers that it cannot write to a datafile, and Oracle is not archiving the filled online redo log files, the DBWn background process fails and the current instance fails. If the problem is temporary (for example, the disk controller was powered off), crash or instance recovery usually can be performed using the online redo log files, in which case the instance can be restarted. However, if a datafile is permanently damaged and archiving is not used, the entire database must be restored using the most recent cold backup.

Recovery of Read-Only Tablespaces
Recovery is not needed on read-only datafiles during crash or instance recovery. Recovery during startup verifies that each online read-only file does not need any media recovery. That is, the file was not restored from a backup taken before it was made read-only. If you restore a read-only tablespace from a backup taken before the tablespace was made read-only, you cannot access the tablespace until you complete

media recovery.

Database Recovery ctd... (A.3.3 ctd.)

DBMS Recovery systems

- Rollback (undoing)
- Deferred update
- Immediate update
- Caching / Buffering
- Shadow Paging

Backup Techniques

- Full database backup
- Differential backup
- Transaction log backup

https://www.geeksforgeeks.org/database-recovery-techniques-in-dbms/

Undoing – If a transaction crashes, then the recovery manager may undo transactions i.e. reverse the operations of a transaction. This involves examining a transaction for the log entry write_item(T, x, old_value, new_value) and setting the value of item x in the database to old-value.There are two major techniques for recovery from non-catastrophic transaction failures: deferred updates and immediate updates.

Deferred update – This technique does not physically update the database on disk until a transaction has reached its commit point. Before reaching commit, all transaction updates are recorded in the local transaction workspace. If a transaction fails before reaching its commit point, it will not have changed the database in any way so UNDO is not needed. It may be necessary to REDO the effect of the operations that are recorded in the local transaction workspace, because their effect may not yet have been written in the database. Hence, a deferred update is also known as the No-undo/redo algorithm

Immediate update – In the immediate update, the database may be updated by some operations of a transaction before the transaction reaches its commit point. However, these operations are recorded in a log on disk before they are applied to the database, making recovery still possible. If a transaction fails to reach its commit point, the effect of its operation must be undone i.e. the transaction must be rolled back hence we require both undo and redo. This technique is known as undo/redo algorithm.

Caching/Buffering – In this one or more disk pages that include data items to be updated are cached into main memory buffers and then updated in memory before being written back to disk. A collection of in-memory buffers called the DBMS cache is kept under control of DBMS for holding these buffers. A directory is used to keep track of which database items are in the buffer. A dirty bit is associated with each buffer, which is 0 if the buffer is not modified else 1 if modified.

Shadow paging – It provides atomicity and durability. A directory with n entries is constructed, where the ith entry points to the ith database page on the link. When a transaction began executing the current directory is copied into a shadow directory. When a page is to be modified, a shadow page is allocated in which changes are made and when it is ready to become durable, all pages that refer to original are updated to refer new replacement page.

——-Backup———
Full database backup – In this full database including data and database, Meta information needed to restore the whole database, including full-text catalogs are backed up in a predefined time series.

Differential backup – It stores only the data changes that have occurred since last full database backup. When same data has changed many times since last full database backup, a differential backup stores the most recent version of changed data. For this first, we need to restore a full database backup.

Transaction log backup – In this, all events that have occurred in the database, like a record of every single statement executed is backed up. It is the backup of transaction log entries and contains all transaction that had happened to the database. Through this, the database can be recovered to a specific point in time. It is even possible to perform a backup from a transaction log if the data files are destroyed and not even a single committed transaction is lost.

Integrated Databases (A.3.4)

Database integrated into application or operating system

iOS and Android both have SQLite3 integrated for applications to use

Database combining data from multiple sources

- Provides a consistent view of various disconnected sources
- Data aggregation
- Database merging
- Growing under Big Data

Discuss one company acquiring another. How does the parent company merge data into their own systems.

What complications might there be?
How can this be overcome?
What procedures do you think a company should use?

Database Uses (A.3.5)

Where are databases in use?

- Stock control
- Police records
- Health records
- Employee data
- Search engines
- Social media
- Video games

What kind of security considerations for each

Security Responsibilities (A.3.6)

- What responsibilities do companies have to protect your data?
- When you use a service, who owns data about you?
- Should you know what data is collected about you?
- What should a company be able to do with data about you

- Data Protection Act
- Computer Misuse Act
- PDPA

"If you're not paying for a product, you ARE the product"

myactivity.google.com

https://www.gov.uk/data-protection (Data protection act, UK)

https://www.itpro.co.uk/it-legislation/28174/what-is-the-computer-misuse-act (Computer Misuse Act)

https://law.moj.gov.tw/ENG/LawClass/LawAll.aspx?pcode=I0050021 (PDPA)

Database access (A.3.7)

Should a database be accessible by

- Law Enforcement?
- Government?
- Should the search be connected to a legal proceeding?
- Under what conditions?

Past Exam Questions

- Discuss one issue related to privacy that might be raised if the data in this database were accessed by the police.
- Discuss one advantage and one disadvantage of using a database in this online information service.
- With specific reference to the data stored, outline two methods to ensure the privacy of the employee's data in the research firm's database.
- Describe two ways that a database management system (DBMS) can be used to promote data security.
- Discuss an ethical issue that could arise if external agencies could obtain and analyse the phone company's data.
- Explain one way that the DBA at ShowTime can ensure the anonymity of the customers is maintained.

Data Matching vs Data Mining (A.3.8)

Data mining is concerned with the development of techniques for discovering novel and useful knowledge in large databases.

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.

Data matching is the process of identifying records that correspond to the same entities, such as patients or customers, across disparate databases.

Data Matching is the task of finding records that refer to the same entity.

https://en.wikipedia.org/wiki/Data_mining
https://www.youtube.com/watch?v=EH3bp5335IU

https://medium.com/neuronio/what-is-data-matching-9478c80da888