Chapter 8

DATABASE BACKUP AND RECOVERY

Database Backup

- Database Backup is storage of data that means the copy of the data.
- It is a safeguard against unexpected data loss and application errors.
- It protects the database against data loss.
- If the original data is lost, then using the backup it can reconstructed.

The backups are divided into two types,

1. Physical Backup
2. Logical Backup

1. Physical backups

- Physical Backups are the backups of the physical files used in storing and recovering your database, such as datafiles, control files and archived redo logs, log files.
- It is a copy of files storing database information to some other location, such as disk, some offline storage like magnetic tape.
- Physical backups are the foundation of the recovery mechanism in the database.
- Physical backup provides the minute details about the transaction and modification to the database.

2. Logical backup

- Logical Backup contains logical data which is extracted from a database.
- It includes backup of logical data like views, procedures, functions, tables, etc.
- It is a useful supplement to physical backups in many circumstances but not a sufficient protection against data loss without physical backups, because logical backup provides only structural information.

Importance Of Backups

- Planning and testing backup helps against failure of media, operating system, software and any other kind of failures that cause a serious data crash.
- It determines the speed and success of the recovery.
- Physical backup extracts data from physical storage (usually from disk to tape). Operating system is an example of physical backup.
- Logical backup extracts data using SQL from the database and store it in a binary file.
- Logical backup is used to restore the database objects into the database. So the logical backup utilities allow DBA (Database Administrator) to back up and recover selected objects within the database.

Methods of Backup

The different methods of backup in a database are:

- **Full Backup -** This method takes a lot of time as the full copy of the database is made including the data and the transaction records.
- **Transaction Log -** Only the transaction logs are saved as the backup in this method. To keep the backup file as small as possible, the previous transaction log details are deleted once a new backup record is made.
- **Differential Backup -** This is similar to full backup in that it stores both the data and the transaction records. However only that information is saved in the backup that has changed since the last full backup. Because of this, differential backup leads to smaller files.

WHY Plan Backups?

There can be multiple reasons of failure in a database because of which a database backup and recovery plan is required. Some of these reasons are:

- A database includes a huge amount of data and transaction.
- If the system crashes or failure occurs, then it is very difficult to recover the database.

There are some common causes of failures such as,

1. System Crash
2. Transaction Failure
3. Network Failure
4. Disk Failure
5. Media Failure

- Each transaction has ACID property. If we fail to maintain the ACID properties, it is the failure of the database system.

1. System Crash

- System crash occurs when there is a hardware or software failure or external factors like a power failure.
- The data in the secondary memory is not affected when system crashes because the database has lots of integrity. Checkpoint prevents the loss of data from secondary memory.

2. Transaction Failure

- The transaction failure is affected on only few tables or processes because of logical errors in the code.
- This failure occurs when there are system errors like deadlock or unavailability of system resources to execute the transaction.

3. Network Failure

- A network failure occurs when a client – server configuration or distributed database system are connected by communication networks.

4. Disk Failure

- Disk Failure occurs when there are issues with hard disks like formation of bad sectors, disk head crash, unavailability of disk etc.

5. Media Failure

- Media failure is the most dangerous failure because, it takes more time to recover than any other kind of failures.
- A disk controller or disk head crash is a typical example of media failure.
- Natural disasters like floods, earthquakes, power failures, etc. damage the data.

6. User Error

Normally, user error is the biggest reason of data destruction or corruption in a database. To rectify the error, the database needs to be restored to the point in time before the error occured.

Hardware Protection and Type of Hardware Protection

Hardware protection is divided into 3 categories: CPU protection, Memory Protection, and I/O protection. These are explained as following below.

1. **CPU Protection:**
   CPU protection is referred to as we can not give CPU to a process forever, it should be for some limited time otherwise other processes will not get the chance to execute the process. So for that, a timer is used to get over from this situation. which is basically give a certain amount of time a process and after the timer execution a signal will be sent to the process to leave the CPU. hence process will not hold CPU for more time.

2. **Memory Protection:**
   In memory protection, we are talking about that situation when two or more processes are in memory and one process may access the other process memory. and to protecting this situation we are using two registers as:
   Bare register
   Limit register
   So basically Bare register store the starting address of program and limit register store the size of the process, so when a process wants to access the memory then it is checked that it can access or can not access the memory.

3. **I/O Protection:**
   So when we ensuring the I/O protection then some cases will never have occurred in the system as:
   1. Termination I/O of other process
   2. View I/O of other process
   3. Giving priority to a particular process I/O

Redundancy

Data redundancy is a condition created within a database or data storage technology in which the same piece of data is held in two separate places.

This can mean two different fields within a single database, or two different spots in multiple software environments or platforms. Whenever data is repeated, this basically constitutes data

redundancy. This can occur by accident, but is also done deliberately for backup and recovery purposes.

Hardware redundancy

Hardware redundancy is achieved by providing two or more physical copies of a hardware component. When other techniques, such as use of more reliable components, manufacturing quality control, test, design simplification, etc., have been exhausted, hardware redundancy may be the only way to improve the dependability of a system.

What Is Recovery?

- Recovery is the process of restoring a database to the correct state in the event of a failure.
- It ensures that the database is reliable and remains in consistent state in case of a failure.

Database recovery can be classified into two parts;
1. Rolling Forward applies redo records to the corresponding data blocks.
2. Rolling Back applies rollback segments to the datafiles. It is stored in transaction tables.

Database Recovery

There are two methods that are primarily used for database recovery. These are:

- **Log based recovery -** In log based recovery, logs of all database transactions are stored in a secure area so that in case of a system failure, the database can recover the data. All log information, such as the time of the transaction, its data etc. should be stored before the transaction is executed.
- **Shadow paging -** In shadow paging, after the transaction is completed its data is automatically stored for safekeeping. So, if the system crashes in the middle of a transaction, changes made by it will not be reflected in the database.

Log-Based Recovery

- Logs are the sequence of records, that maintain the records of actions performed by a transaction.
- In Log – Based Recovery, log of each transaction is maintained in some stable storage. If any failure occurs, it can be recovered from there to recover the database.
- The log contains the information about the transaction being executed, values that have been modified and transaction state.
- All these information will be stored in the order of execution.

Example:
Assume, a transaction to modify the address of an employee. The following logs are written for this transaction,
Log 1: Transaction is initiated, writes 'START' log.
Log: <T_n START>

Log 2: Transaction modifies the address from 'Pune' to 'Mumbai'.
Log: <T_n Address, 'Pune', 'Mumbai'>

Log 3: Transaction is completed. The log indicates the end of the transaction.
Log: <T_n COMMIT>

There are two methods of creating the log files and updating the database,
1. Deferred Database Modification
2. Immediate Database Modification

1. In Deferred Database Modification, all the logs for the transaction are created and stored into stable storage system. In the above example, three log records are created and stored it in some storage system, the database will be updated with those steps.

2. In Immediate Database Modification, after creating each log record, the database is modified for each step of log entry immediately. In the above example, the database is modified at each step of log entry that means after first log entry, transaction will hit the database to fetch the record, then the second log will be entered followed by updating the employee's address, then the third log followed by committing the database changes.

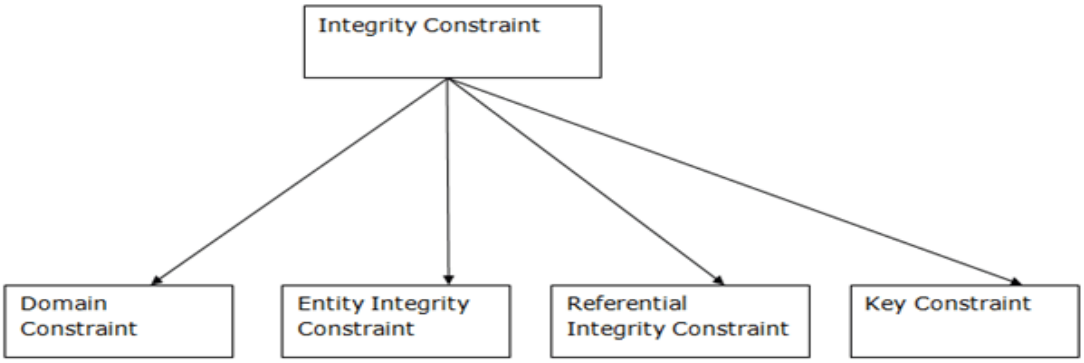
Chapter 9

Database Security And Integrity

Integrity Constraints

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to guard against accidental damage to the database.

Types of Integrity Constraint



1. Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

Example:

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

- The entity integrity constraint states that primary key value can't be null.

- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.

Example:

EMPLOYEE

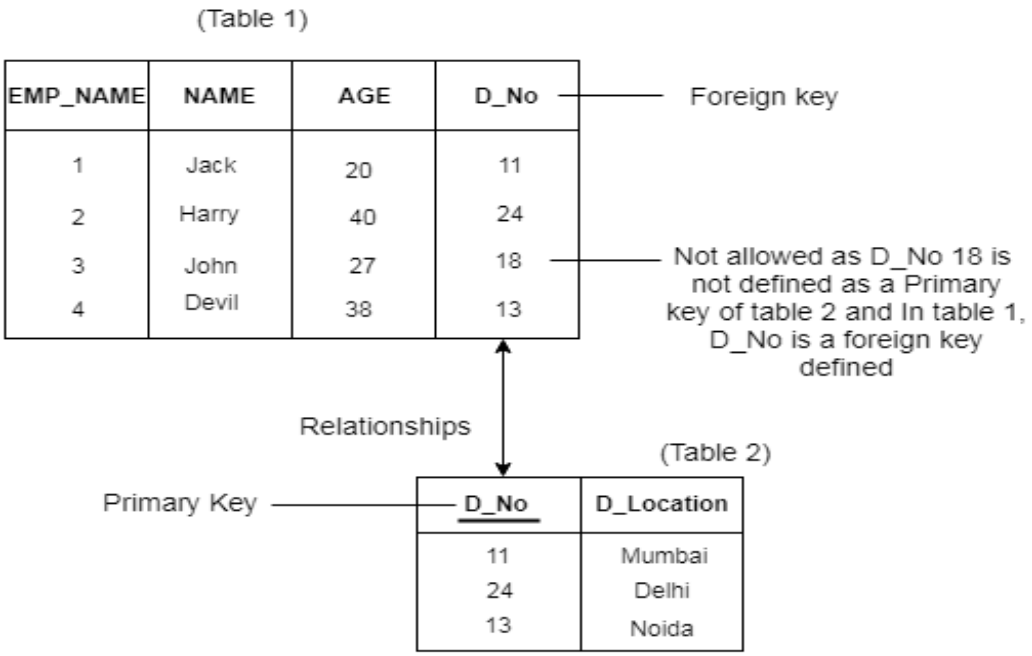
EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

3. Referential Integrity Constraints

- A referential integrity constraint is specified between two tables.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.

Example:



4. Key constraints

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.

Example:

ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

Data Integrity Restrictions

NOT NULL and CHECK

NOT NULL and CHECK integrity constraints are allowed. They are not a problem for parallel DML because they are enforced on the column and row level, respectively.

UNIQUE and PRIMARY KEY

UNIQUE and PRIMARY KEY integrity constraints are allowed.

FOREIGN KEY (Referential Integrity)

Restrictions for referential integrity occur whenever a DML operation on one table could cause a recursive DML operation on another table. These restrictions also apply when, to perform an integrity check, it is necessary to see simultaneously all changes made to the object being modified.

Delete Cascade

Deletion on tables having a foreign key with delete cascade is not parallelized because parallel execution servers attempt to delete rows from multiple partitions (parent and child tables).

Self-Referential Integrity

DML on tables with self-referential integrity constraints is not parallelized if the referenced keys (primary keys) are involved. For DML on all other columns, parallelism is possible.

Deferrable Integrity Constraints

If any deferrable constraints apply to the table being operated on, the DML operation is not executed in parallel.

Database Security and Threats

Data security is an imperative aspect of any database system. It is of particular importance in distributed systems because of large number of users, fragmented and replicated data, multiple sites and distributed control.

Threats in a Database

- **Availability loss** − Availability loss refers to non-availability of database objects by legitimate users.
- **Integrity loss** − Integrity loss occurs when unacceptable operations are performed upon the database either accidentally or maliciously. This may happen while creating, inserting, updating or deleting data. It results in corrupted data leading to incorrect decisions.
- **Confidentiality loss** − Confidentiality loss occurs due to unauthorized or unintentional disclosure of confidential information. It may result in illegal actions, security threats and loss in public confidence.

we consider database security about the following situations:

- Theft and fraudulent.
- Loss of confidentiality or secrecy.
- Loss of data privacy.
- Loss of data integrity.
- Loss of availability of data.

These listed circumstances mostly signify the areas in which the organization should focus on reducing the risk that is the chance of incurring loss or damage to data within a database.

Data Security Requirements

Database security requirements arise from the need to protect data:

First, from accidental loss and corruption, and Second, from deliberate unauthorized attempts to access or alter that data. Secondary concerns include protecting against undue delays in accessing or using data, or even against interference to the point of denial of service. .

Ensure that physical damage to the server doesn’t result in the loss of data.

Prevent data loss through corruption of files or programming errors.

These requirements are dynamic. New technologies and practices continually provide new arenas for unauthorized exploitation, as well as new ways for accidental or deliberate misuse to affect even stable products and environments.

Today's evolution involves a globally changing technological and cultural environment, in which security concerns necessarily affect both the use of existing solutions and the development of new ones.

Database Users

Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

1. Database Administrator (DBA):

DBA Stands for Database Administrator.

It is a person or a team, who is responsible for managing the overall database management system.

It is the leader of the database. It is like a super user of the system.

It is responsible for the administration of all the three levels of the database.

DBA is responsible for:

- Deciding the instances for the database.
- Defining the Schema
- Liaising with Users
- Define Security
- Back-up and Recovery
- Monitoring the performance

2. Database Designers:
Database designers design the appropriate structure for the database, where we share data.

3. System Analyst:
System analyst is responsible for the design, structure and properties of database. All the requirements of the end users are handled by system analyst. Feasibility, economic and technical aspects of DBMS is the main concern of system analyst.

4. Application Programmers:
Application programmers are computer professionals, who write application programs.

These application programs are written in programming languages like COBOL or PL (Programming Language 1), Java and fourth generation language. These programs meet the user requirement and made according to user requirements. Retrieving information, creating new information and changing existing information is done by these application programs.

5. Naïve Users / Parametric Users:
Any user who does not have any knowledge about database can be in this category. There task is to just use the developed application and get the desired results. For example: Clerical

staff in any bank is a naïve user. They don't have any dbms knowledge but they still use the database and perform their given task.

6. Sophisticated Users:
Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. These users interact with the database but they do not write programs

7. Casual Users / Temporary Users:
These types of users communicate with the database for a little period of time.

These users have great knowledge of query language. Casual users access data by entering different queries from the terminal end. They do not write programs but they can interact with the system by writing queries.

Protecting Data within Database

Authorization is permission given "to user, program, or process to access an object or set of objects. The type of data access granted to a user can be read-only, or read and write. Privileges specify the type of Data Manipulation Language (DML) operations like SELECT, INSERT, UPDATE, DELETE, etc., which the user can perform upon data.

The two methods by which the access control is done are by using privileges and roles.

• Privilege

• Role

Privileges

A privilege is permission to access a named object in a prescribed manner; for example, permission to query a table. Privileges can be granted enable a particular user to connect to the database (create a session); select rows from someone else's table; or execute someone else's stored procedure.

Database privileges

A privilege is a right to execute a particular type of SQL statement or to access another user's object. Some examples of privileges include:

• The right to connect to the database (create a session)

• The right. to create a table

• The right to select rows from another user's table

• The right to execute another user's stored procedure

Privileges are granted to users so that these can accomplish tasks required for their job. You should grant a privilege only to user who absolutely requires the privilege to accomplish necessary work. Excessive granting of unnecessary privileges can lead to compromised security.

System privileges

A system privilege is the right to perform a particular action, on a particular type of object. For example, the privileges to create tables and to delete the (OWS of any table in a database are system privileges. In many commercial database management systems there are hundreds of distinct system privileges.

Object privileges

An object privilege is a privilege or right to perform a particular action on a specific table, view, sequence, procedure, function, or package. For example, the privilege to delete rows from the table DEPT is an object privilege.

Object privileges granted for a table, view, sequence, procedure, function, or package apply whether referencing the base object by name or using a synonym.

Roles

A role is a mechanism that can be used to provide authorization. A single person or a group of people can be granted a role or a group of roles. One role can be granted in turn to other roles. By defining different types of roles, administrators can manage access privileges much more easily.

Database management systems provide for easy and controlled privilege management through roles. Roles are named groups of related privileges that you grant to users or other roles. Roles are designed to ease the administration of end-user system and object privileges.

The following properties of roles allow for easier privilege management vvithin a database:

• **Reduced privilege administration** - Rather them explicitly granting the same set of privileges to several uses, you can grant the privileges for a group of related users to a role. Then, only the role needs to be granted to each member of the group.

• **Dynamic privilege management** - If the privileges of a group must change, only the privileges of the role need to be modified.

• **Selective availability of privileges** - You can selectively enable or disable the roles granted to a user. This allows specific control of a user's privileges in any given situation.

• **Application-specific security** - you can protect role use with a password. Applications can be created specifically to enable a role when supplied the correct password. Users cannot enable the role if they do not know the password.

Granting and Revoking Privileges and Roles

You grant or revoke privileges and roles from users or other roles using the SQL commands GRANT and REVOKE. Who can grant and revoke the privileges and roles? The answer is "it depends on the DBMS". For example, in Oracle, a..'1yus.cr with the GRANT ANY ROLE system privilege can grant or revoke any role to or from other users or roles of the database. IJ:1 most database management systems, the Database Administrators (DBAs) and Security Officers will have the necessary powers to grant and revoke the privileges and roles to the users.

GRANT Command

The GRANT command gives users privileges to base tables and views.

The syntax of this command is as follows:

GRANT privileges ON object to users [with GRANT Option]

Here, object is either a base table or a view. If 'with GRANT Option' is specified, it means that the recipient has authority to grant the privileges that were granted to him to another user.

Examples

• Grant the SELECT authority on the EMP table to all users.

GRANT SELECT ON EMP TO PUBLIC;

• Grant the SELECT, DELETE and UPDATE authority on DEPT table to user 'AJAY'.

GRANT SELECT, DELETE, UPDATE ON DEPT TO AJAY;

• Grant the SELECT.' DELETE and UPDATE authority with the capability to grant those privileges to others users on DEPT table to user 'AJAY'.

GRANT SELECT, UPDATE ON DEPT TO AJAY WITH GRANT OPTION;

• Grant ALL privileges on EMP table to user 'DEEP'.

GRANT ALL ON EMP TO DEEP;

• Give the system privileges for creating tables and views to 'AJAY'.

GRANT CREATE TABLE, CREATE VIEW TO AJAY

• Grant the UPDATE authority on the SAL column of the EMP to user 'AJAY'.

GRANT UPDATE (SAL) ON EMP TO AJAY;

REVOKE Command

REVOKE is a complementary command to GRANT that allows the withdrawal of privileges. The syntax of the REVOKE command is as follows:

REVOKE [GRANT OPTION FOR] privileges

ON object FROM users {Restrict I Cascade}

The command can be used to REVOKE either a privilege or just the GRANT option on a privilege (by using the optional GRANT OPTION FOR Clause). When a user executes a REVOKE command with the CASCADE keyword, the effect is to withdraw the named privileges or GRANT option from all users who currently hold these privileges solely through a GRANT command that was previously executed by the same user who is now executive the REVOKE command. If these users received the privileges with the GRANT option and passed it along, those recipients will also lose their privileges as a consequence of the REVOKE command unless they also received these privileges independently. RESTRICT keyword is a complementary to CASCADE keyword.

Examples

• Revoke the system privileges for creating table from 'AJAY'.

REVOKE CREATE TABLE FROM AJAY;

• Revoke the SELECT privileges on EMP table from *'AJAY'.*

REVOKE SELECT ON EMP FROM AJAY;

• Revoke the UPDATE privileges on EMP table from all users.

REVOKE UPDATE ON EMP FROM PUBLIC;

• Remove ALL privileges on EMP table from user 'AJAY'.

REVOKE ALL ON EMP FROM AJAY;

• Remove DELETE and UPDATE authority on the SAL and JOB columns of the EMP table from user 'AJAY'.

REVOKE DELETE, UPDATE (SAL, JOB) ON EMP FROM AJAY;