[[1]](#footnote-1)

Teresa Ventaja

Higher Diploma in Science in Data Analytics

Data normalization, retrieval & warehousing

Introduction to Data Analytics

INDEX

[1. Data Normalisation 2](#_Toc40689854)

[2. Data Warehousing & Data Marts 3](#_Toc40689855)

[3. Data Security 4](#_Toc40689856)

[4. Data Retrieval 5](#_Toc40689857)

[BIBLIOGRAPHY 7](#_Toc40689858)

# 1. Data Normalisation

When we organize data, some ways are more appropriate than others. Unwanted modification anomalies can happen if the database is not well structured. These can happen upon addition, change or elimination of data. To prevent this from happening, normalization is the process of dividing a table into multiple ones, each of which containing a single theme[[2]](#footnote-2). It is a multi-step process that puts data into tabular forms, eliminating redundancy and inconsistent dependency from the tables[[3]](#footnote-3). Duplicated data occupy space in the disk and creates maintenance issues[[4]](#footnote-4). Other reasons to normalize is to simplify queries and make it easier to sort and search data[[5]](#footnote-5).

E. F. Codd created this normalization system[[6]](#footnote-6). There are 4 common forms of normalization, and additional forms, such as BCNF. **They are progressive, so that the higher form level must first satisfy the roles for the lower level, and so on**. They abbreviate as 1NF, 2NF, 3NF, BCNF and 4NF, in that order[[7]](#footnote-7): The rules to create the forms can be found below[[8]](#footnote-8)[[9]](#footnote-9):

**1NF**

* Eliminating duplicate groups in individual tables.
* Creating a separate table for each set of related data.
* Identifying each set of related data with a primary key.

**2NF**

* Creating separate tables for sets of values that apply to various records.
* Relating these tables with a foreign key [records should not depend on anything other than a table's primary key (a compound key, if necessary)].

**3NF**

* Eliminate fields that do not depend on the key (no transitive dependency)

In general, when the contents of a group of fields may apply to more than one single record in the table, we should consider placing those fields in a separate table. Adhering to the third normal form, while theoretically desirable, is not always practical. Many small tables may degrade performance or exceed open file and memory capacities. It is recommended to apply third normal form only to data that changes frequently.

**Boyce and Codd Normal Form (BCNF)**

It is a more structured version of a 3NF table which does not have multiple overlapping candidate keys. It must meet these conditions:

* R must be in 3rd Normal Form
* For each functional dependency (X → Y), X should be a super Key.

**4NF**

* The table doesn't have Multi-Valued Dependency[[10]](#footnote-10).

4NF is rarely used outside of academic circles[[11]](#footnote-11).

But normalization also has some **disadvantages**[[12]](#footnote-12)[[13]](#footnote-13):

* Need of usage of table joins and complicated queries, due to the data not being duplicated. That makes the queries more complex and loading time is longer.
* Indexing work more inefficiently with joins, which again makes loading time longer.
* Need for detailed design and analysis, which can be a difficult task and requires knowing the purpose of the database in details
* The duplicated data will be stored as lines of codes rather than the real data. Therefore, there is a need to go to the lookup table, making system performance slower.

# 2. Data Warehousing & Data Marts

First, I am going to talk about the difference between a data warehouse (DW) and a database. A data warehouse is created to store a vast amount of historical data, while a database is set to update real-time data enabling quick access to ongoing business processes[[14]](#footnote-14). One database source an application in a one-to-one relationship, while in DW the relationship is one to many, since they aggregate historical data from multiple applications. Examples of applications that serve as data sources are the social media or a CRM. DW are expensive to scale, so they are detached from frontend applications to support scalability. DW use case include the execution of data mining to extract insights or carrying out market research, ultimately aiming to improve the company performance[[15]](#footnote-15).

A drawing of a face

Description automatically generated

Initially, the concept was invented by Inmon Bill, who had written about various topics on warehouse. The data comes from the relational databases and it can be raw, unstructured, semi-structured or complex. The data is modified and processed so that with the use of BI tools, spreadsheets and SQL servers it can be accessed[[16]](#footnote-16)[[17]](#footnote-17). Types of DW are[[18]](#footnote-18):

**1. Enterprise Data Warehouse**

It is a database that unifies diverse functional areas, centralizing all the data from different sources. It provides a single location where analytical tools can be used to identify trends. It helps to reduce expensive downtime with machine learning approaches, which may occur due to faulty configurations. When it is implemented on a relatively small scale, it structures data consistently.

**2. Operational Data Store**

It stores the operations that are currently being performed before the data is moved to the data warehouse for a longer time. It helps to contrast data from various sources to perform analytical tasks. It works effectively for reduced amounts of data and simple queries, acting as a temporary storage for recent information. Therefore, it is preferred for day-to-day tasks like storing transactions.

**3. Data Mart**

It stores data from a specific functional area, holding a subset of the data contained in the data warehouse. It helps in improving user responses and, in data analysis, decreases the volume of data, making the research easier. It has an easy implementation and is cost-effective in comparison with a complete DW. The data can be more simply controlled because it is partitioned. Data Marts can be dependent, independent or hybrid.

A **trend** in DW is the **shift to the in-cloud environment**, where companies can benefit from a higher scalability and flexibility, diverse pricing options in accordance with companies’ technical needs and budgets, and data availability (automatic backups and fault tolerance). Another trend is the change to DW as a Service, whose main benefit is addressing the challenge of implementation, minimizing data administration efforts. It eliminates the cost of hardware and software acquisition and maintenance and the cost of having a DW team as well. Lastly, Big Data tools integrated into the DW architecture is trendy as well, a huge step towards accurate forecasting and boosting profit[[19]](#footnote-19).

# 3. Data Security

Data Security (DS) is a set of activities aiming to protect data, the most essential business asset, from unwanted or unauthorized use. It is a team effort[[20]](#footnote-20) and involves the protection of databases, files and accounts on a network by different techniques and controls, depending on regulatory compliance requirements and their sensitivity. It is a method to put in place when saving any sort of data, consisting in evaluating and reducing the risk on the storage and treatment of data.

Integrity, confidentiality and availability are the main components of DS. Things to be considered about DS are the location of the data and who have access to it. Alerting and monitoring are essential not only to comply with regulations, but also to identify unusual tasks and distrustful accounts on a timely manner[[21]](#footnote-21). The types of DS controls include[[22]](#footnote-22):

* **Authentication** – it corroborates if a user’s credentials match those saved on the database. Nowadays it is extended to combine different methods, such as biometrics, passwords and security tokens. Authentication and authorization are usually defined as “*access control*”.
* **Access Control** – There are 3 methods: the discretionary is the least restrictive, assigns access based on user or group identity; role-based allows access based on organizational job, accessing only to specific data, and mandatory access control, having a system administrator who is responsible for strictly control access to all data.
* **Backups & Recovery** – it requires copying the data and storing it on an autonomous system, in case it is necessary to recover lost data. It helps in the unfortunate event of a breach, system error, data corruption, etc.
* **Encryption** – it protects data by employing an algorithm (a cipher) with an encryption key, turning text into encrypted ciphertext, not readable for unauthorized users. It protects “data at rest” (the one that is stored) and the “data in transit” (when it is exchanged). Encryption keys must be backup offsite with restrictive access.
* **Data Masking** – it consists in hiding data with proxy characters, by masking digits. The data is readable when it is received by an authorized user.
* **Tokenization** – it implies the substitution of sensitive data with random characters that are not reversible algorithmically. A database lookup table deposits the relationship between data and its token equivalents. Tokens are used as a replacement, while the real data is located on a platform separately.
* **Deletions & Erasure** – while deletion is permanent and irreversible, erasure is the process of hiding data so it can be retrieved easily.

# 4. Data Retrieval

Data retrieval (DR) is the ability to effortlessly select data for attribute editing, updating, querying, analysis or display. It is based on the structure of the Database Management System and the software usually provides the command interface. Querying is equivalent to retrieve data, typically a subset defined by the user. Data subsets are also called logical views. Frequently the querying is closely associated with the data analysis and manipulation subsystem[[23]](#footnote-23).

A database system would be useless without the ability of writing or extracting the stored data. The most used SQL statement to retrieve data is the SELECT keyword, which can extract multiple columns by simply specifying the columns names separated by a comma. Using the LIMIT statement, we can limit the number of results returned, followed by a number (number of rows to be retrieved). To avoid duplicate values in our result, we can use the DISTINCT statement[[24]](#footnote-24). There are other clauses which can be used to sort and specify the data we want to get, such as ORDER BY, LIKE/NOT LIKE or COUNT.

To retrieve data from multiple tables, we can use the statement JOIN, and its versions (INNER, LEFT, RIGHT and CROSS joins). When obtaining information from several tables, we need to specify how records in one table can be matched to records in the other, by mentioning the name of the columns (using the ON clause)[[25]](#footnote-25). When a query needs to be used multiple times, we can save it into the database server and name it. This is called a view and it can be created by using the statements CREATE VIEW. It does not store data and it can be used with the SELECT command to run the underlying query. Views have multiple advantages, such as simplifying complex queries, adding consistency to the business logic, enhancing security by limit which data user can access and enabling backward compatibility in legacy systems[[26]](#footnote-26).

Another interesting DR method is using a Stored Procedure. This technique makes repetitive tasks faster by storing the procedure in the server, they are portable and usable on any MySQL platform, and they can be migrated[[27]](#footnote-27). It avoids duplicating database code, saving time and effort when updates are introduced, tune the performance of the queries, and can add new operations for logging, security, etc. We can create one using the command CREATE PROCEDURE and run it by the keyword CALL, adding the name of it[[28]](#footnote-28).

Finally, a TRIGGER is a command that runs automatically when we perform an INSERT, UPDATE or DELETE operation. It is linked to a table and we can specify an action and whether it will be invoked before or after the defined operation. Some advantages are referential integrity[[29]](#footnote-29), storing the value of a derived column when an UPDATE happens, inserting user actions into audit tables. They permit keeping validation rules in the database, so that the data can be used in multiple applications without breaking the business logic, which helps reducing the response time[[30]](#footnote-30).

# BIBLIOGRAPHY

Bekker, A. (2019): “*Trends on the Data Warehouse Implementation Market*” [Online] Available at: <https://www.scnsoft.com/blog/data-warehouse-implementation> (Accessed 17/05/2020)

1. Buckbee, M. (2020): “*Data Security: Definition, Explanation and Guide*” [Online] Available at: <https://www.varonis.com/blog/data-security/> (Accessed 17/05/2020)

Buckey, D. J. (1997): “*Data Retrieval and Querying*” [Online] Available at: <http://planet.botany.uwc.ac.za/nisl/GIS/GIS_primer/page_31.htm> (Accessed 17/05/2020)

EssentialSQL (2020): “*Database Normalization (Explained in Simple English)”* [Online] Available at: <https://www.essentialsql.com/get-ready-to-learn-sql-database-normalization-explained-in-simple-english/> (Accessed 17/05/2020)

Frankenfield, J. (2019): “*Data Warehousing*” [Online] Available at: <https://www.investopedia.com/terms/d/data-warehousing.asp> (Accessed 17/05/2020)

Guru99 (2020): “*What is Data Warehouse? Types, Definition & Example”* [Online] Available at: <https://www.guru99.com/data-warehousing.html> (Accessed 17/05/2020)

IBM (2020): “*Data security*” [Online] Available at: <https://www.ibm.com/security/data-security> (Accessed 17/05/2020)

Informatica (2020): “*What is Data Warehousing?”* [Online] Available at: <https://www.informatica.com/services-and-training/glossary-of-terms/data-warehousing-definition.html> (Accessed 17/05/2020)

1. Looker (2020): “*Data Security*” [Online] Available at: <https://looker.com/definitions/data-security> (Accessed 17/05/2020)

Microsoft (2020): “*Description of the database normalization basics”* [Online] Available at: <https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description> (Accessed 17/05/2020)

Morpheus Data (2015): “*Pros and Cons of Database Normalization*” [Online] Available at: <https://morpheusdata.com/blog/2015-02-17-pros-cons-db-normalization> (Accessed 17/05/2020)

MySQL (2020): “3.3.4.9 Using More Than one Table”, *Oracle* [Online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/multiple-tables.html> (Accessed 17/05/2020)

MySQL (2020): “6.1.5 Working with Stored Procedures”, *Oracle* [Online] Available at: <https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-procedures.html> (Accessed 17/05/2020)

MySQLTUTORIAL (2009): *“MySQL Views*” [Online] Available at: <https://www.mysqltutorial.org/mysql-views-tutorial.aspx/> (Accessed 17/05/2020)

Ndungu, F. (2019): “*How To Manage and Use MySQL Database Triggers on Ubuntu 18.04”* [Online] Available at: <https://www.digitalocean.com/community/tutorials/how-to-manage-and-use-mysql-database-triggers-on-ubuntu-18-04> (Accessed 17/05/2020)

Panoply (no date): “*The Difference Between a Data Warehouse and a Database”* [Online] Available at: <https://panoply.io/data-warehouse-guide/the-difference-between-a-database-and-a-data-warehouse/> (Accessed 17/05/2020)

1. Patil, A. (2019): “*Types of Data Warehouse”* [Online] Available at: <https://www.educba.com/types-of-data-warehouse/> (Accessed 17/05/2020)

Shanardi, A. (2018): “*Drawbacks of Normalization*” [Online] Available at: <https://sis.binus.ac.id/2018/01/17/drawbacks-of-normalization/> (Accessed 17/05/2020)

Shedor, T. (2020): “*MySQL Stored Procedure*” [Online] Available at: <https://www.w3resource.com/mysql/mysql-procedure.php> (Accessed 17/05/2020)

Studytonight (2020): “*Normalization of Database*” [Online] Available at: <https://www.studytonight.com/dbms/database-normalization.php> (Accessed 17/05/2020)

Taylor, A. G (2006).: *“SQL for Dummies*”, 6th Edition, Indiana (USA):Wiley Publishing

TechopediA (2014): “*Fourth Normal Form (4NF)”* [Online] Available at: <https://www.techopedia.com/definition/19453/fourth-normal-form-4nf> (Accessed 17/05/2020)

Techotopia (no date): “*Retrieving Data From a MySQL Database*” [Online] Available at: <https://www.techotopia.com/index.php/Retrieving_Data_From_a_MySQL_Database> (Accessed 17/05/2020)

1. Coderre, D. (2019) Available at: <https://www.wegalvanize.com/fraud/analytics-detect-prevent-fraud/> [↑](#footnote-ref-1)
2. Taylor, A. G (2006) [↑](#footnote-ref-2)
3. Studytonight (2020) [↑](#footnote-ref-3)
4. Microsoft (2020) [↑](#footnote-ref-4)
5. EssentialSQL (2020) [↑](#footnote-ref-5)
6. He proposed that “*users deal, not with relations which are domain-ordered, but with relationships which are their domain-unordered counterparts.2 To accomplish this, domains must be uniquely identifiable at least within any given relation, without using position (…) In order to discuss a preferred way (or normal form), we must first introduce a few additional concepts (active domain, primary key, foreign key, nonsimple domain) and establish some links with terminology currently in use in information systems programming*”. [↑](#footnote-ref-6)
7. EssentialSQL (2020) [↑](#footnote-ref-7)
8. Microsoft (2020) [↑](#footnote-ref-8)
9. Studytonight (2020) [↑](#footnote-ref-9)
10. An example of a multivalued dependency: a table containing 3 elements – jobs, managers and employees (independent elements). Being a multivalued dependency, one item depends on more than one element, in this example, the job depends on both managers and employees. [↑](#footnote-ref-10)
11. TechopediA (2014) [↑](#footnote-ref-11)
12. Shanardi, A. (2018) [↑](#footnote-ref-12)
13. Morpheus Data (2015) [↑](#footnote-ref-13)
14. Frankenfield, J. (2019) [↑](#footnote-ref-14)
15. Panoply (no date) [↑](#footnote-ref-15)
16. Guru99 (2020) [↑](#footnote-ref-16)
17. Informatica (2020) [↑](#footnote-ref-17)
18. Patil, A. (2019) [↑](#footnote-ref-18)
19. Bekker, A. (2019) [↑](#footnote-ref-19)
20. IBM (2020) [↑](#footnote-ref-20)
21. Buckbee, M. (2020) [↑](#footnote-ref-21)
22. Looker (2020) [↑](#footnote-ref-22)
23. Buckey, D. J. (1997) [↑](#footnote-ref-23)
24. Techotopia (no date) [↑](#footnote-ref-24)
25. MySQL (2020) [↑](#footnote-ref-25)
26. MySQLTUTORIAL (2009) [↑](#footnote-ref-26)
27. Shedor, T. (2020) [↑](#footnote-ref-27)
28. MySQL (2020) [↑](#footnote-ref-28)
29. The trigger can be used to save a record with multiple related tables without losing the link to the root table. [↑](#footnote-ref-29)
30. Ndungu, F. (2019) [↑](#footnote-ref-30)