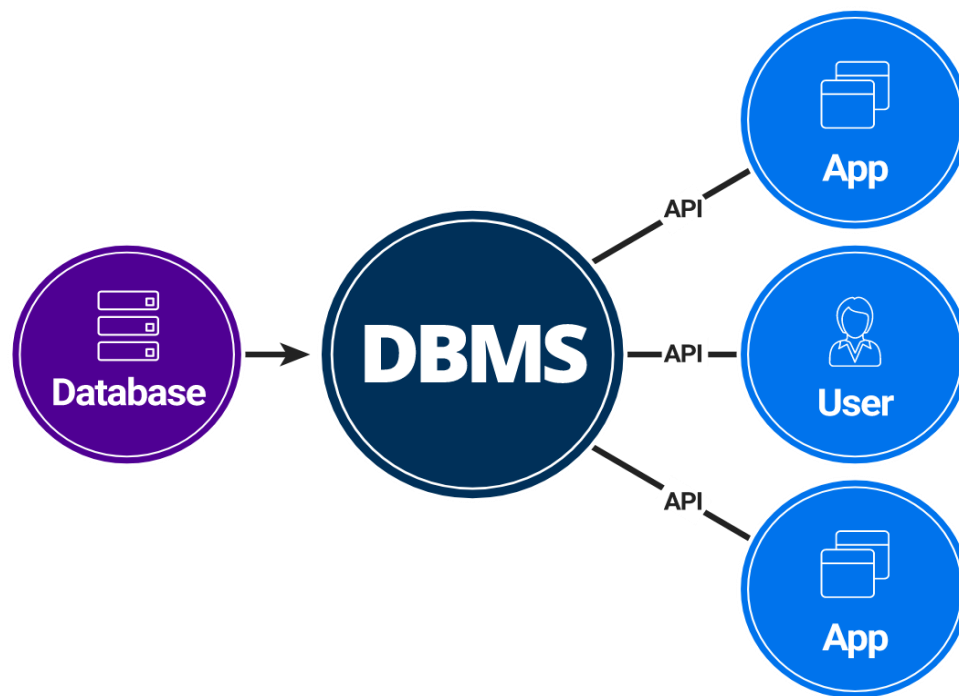


Modules_1- Introduction to Databases Systems & SQL

1.1 Introduction and Background

How were things managed before technology became a major influence in our lives? Tracking information was difficult before the employment of databases. There was plenty of room for error with the pen and paper method. It was not until the 1960s when databases were used from a computer-based format. However, most computerized databases still use the principles and methods developed in the previous age.

Databases now are used everywhere to store information. Whether it be in a customer management system or tracking bank information, databases are utilized to store the necessary data for later use. Data is structured in rows and columns featuring different fields for queries and stored in multiple tables to showcase the relationship between them. According to Oracle.com, “Databases have evolved dramatically since their inception in the early 1960s” (Oracle). In the beginning, only navigational databases, such as the hierarchical and network database, were employed. As time went on, new types of databases were created based on the needs of organizations and the management of their data.



Source: <https://www.smartsheet.com/database-management>

In the graph shown above, a database is used in a database management system (DBMS) for short, is a form of software that allows an organization to access and manipulate data that will be showcased in a form that is unable to be changed by other applications and users.

1.2 Limitations of Conventional File Processing

Files are used to store specific data for future use and recollection. When computers first became mainstream, files were stored like paper, in the form of flat files. This information was collected in notepads separated by spaces, commas, semicolons or other symbols. Organization of files was often based on their categories, consisting of only related information with specific names. The downside to this is that you were unable to open the files without using a specific coding language to edit it. While it appeared convenient at the time, it is easy to identify the many disadvantages to using this system.

1.3 Data Redundancy

One of the major problems with this system was data redundancy and inconsistency. Since the files and programs jammed into files were created by several different programmers over a long period of time, the files were certain to be in different formats, involving several different programming languages. Most of the information is also constantly duplicated due to how tedious it would be to access others' code and double check the information. For example, if a customer of a bank has two accounts, the data accompanied by these accounts would be stored in two separate files in order to satisfy both accounts as they are made. This leads to data redundancy. This would lead to bigger storage sizes for the same information, increasing the cost.

1.4 Data Accuracy

The countless copies of this data could also have discrepancies, making it impossible to know which information is accurate. Whenever a new value needs to be entered into the database, every single file with this data has to be updated to prevent this. This would lead to tedious work that wasn't 100% accurate in the end. For example, a company could have stored customer data, including name, address, and city. There could be a request in which the record of a customer who lives in a specific city is needed. In order to achieve this, a new program would need to be written and executed, and the file containing the customer's city had to be accessed. Every single customer who belonged to this city would need to be specifically selected and taken out into this new program in order to organize the data. This is neither convenient nor reliable. These copies also contributed to the difficulty involving the creation of new applications, as they may be unable to find the appropriate data. This also ensured atomicity didn't work. Atomicity is a sequence of database processes such that either all occur, or nothing occurs. This could be used to prevent updates to a database occurring only partially; however, atomicity is unable to work unless it is able to read and write to every single file, which in this structure, is extremely difficult.

There was also a difficulty in accessing data due to the "spaghetti code" structure of this system. If a specific set of information is needed to be organized in a new way, unless it was anticipated prior to the initially being created, it was nearly impossible to achieve this. The application needed to display the information in the requested way would not have existed. This system doesn't allow data to be retrieved in a convenient manner, leading to different systems created down the line.

Integrity problems were also created due to the data values in a database needing to satisfy certain types of consistency constraints. Since most of the code involving these files is in different languages, it is almost impossible to change them all to enforce new constraints. The file system also lacks concurrent access. In modern systems, multiple users can update the data simultaneously. This is to ensure a faster response time and to improve the overall performance of the system. The involvement of multiple users may result in inconsistent data, which is normally prevented using supervision. However, in a file processing system, this supervision is lackluster due to the several applications and various languages. It all leads to the same problems in the end.

1.5 Data Security

Security is also a major issue in this system. In a database, every user in the database system shouldn't be able to access all the data. Each user should be delegated and only allowed to access specific data requiring a password of sorts. In a file processing system since different programmers add their own application, there is either a universal password or so many passwords that the information is scrambled and the people requiring it can't access it. Since every new file is only added when needed, it is difficult to constantly change the permission for each individual file in order to ensure security standards.

These disadvantages would lead many to convert to a database approach rather than a file system. A database corrected many of these errors reducing the development time and increasing the data integrity of

every file. It is true that file processing systems were full of many errors, but they are known as a stepping stone towards more perfected systems of data storage.

1.6 Advantages of Databases

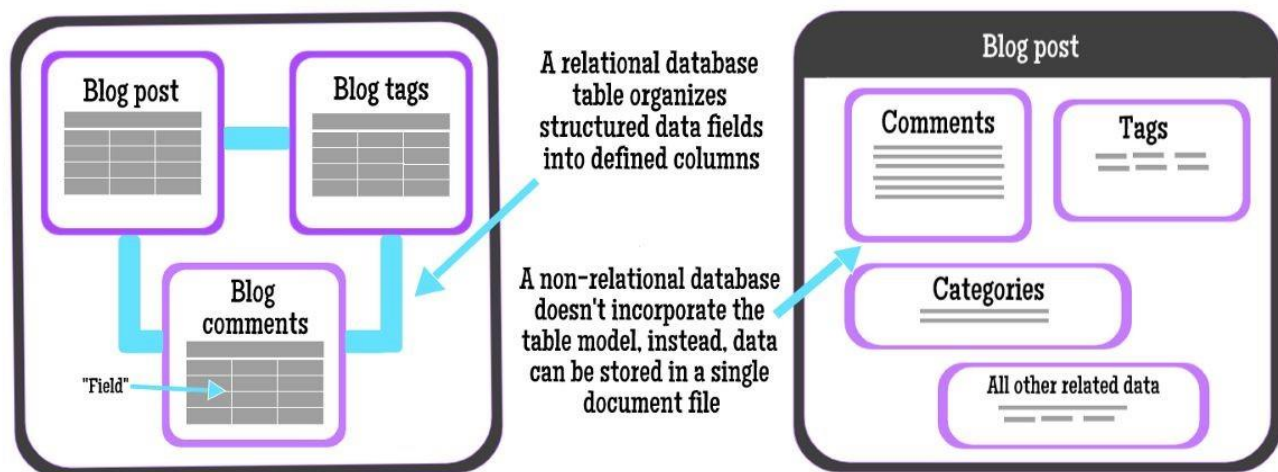
In today's world, data is prevalent in every aspect of our lives as human beings. Data is constantly being created, organized, and stored. With all this data being transferred and exchanged around the world, it is important to have an efficient and organized method to storing this data. This is where databases come in. Databases offer improved efficiency and versatility, they allow categorization and structuring of available data, and they allow multi-user access, creating an organized work environment and newer and better ways to manage data.

Efficiency comes into play specifically with businesses. Databases can handle large amounts of data as well as multiple types of data. Businesses can use databases to have data easily accessible to make operational decisions on a daily basis.

Versatility is also important in terms of accessing data. Databases can be accessed via desktop, laptop, tablet and even mobile devices. This is incredibly helpful in a time where so much importance is placed on accessing things immediately, as data can be easily retrieved at any moment. This benefit is applicable to consumers as well as businesses.

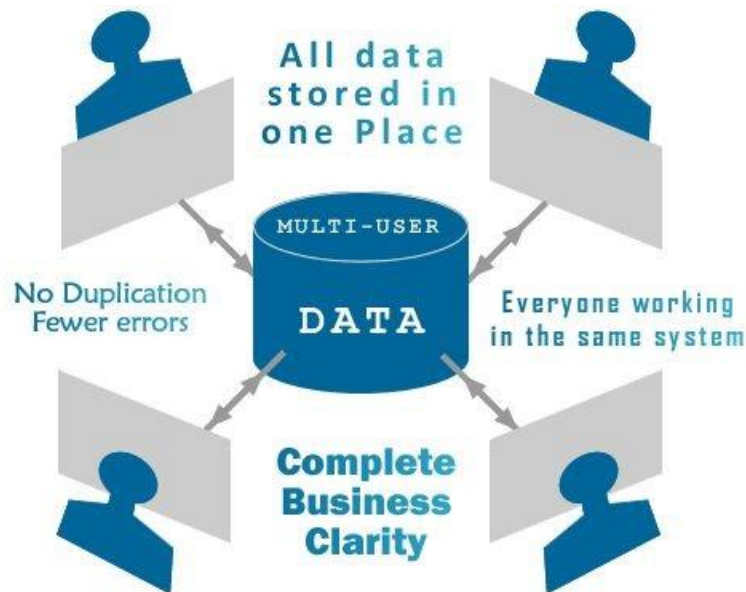
Categorization and organization are both major advantages. They allow the structuring of information in ways that are easily understandable and accessed. Certain DBMS allow relationships between entities in order to simplify the organization of data.

Relational Vs. Non-relational Databases



Source: Liz Parody (Databases for Front-End Developers;Medium.com)

Accessing data in a multitude of ways by multiple different users is also a huge advantage that databases have; this is called multi-access. Multi-access is what allows multiple authorized users to have access to the same data. For example, a human resources manager at a company will have access to the same set of potential hires at a certain location as the general manager of that same location. The picture below visually describes the relationship between this shared data and the users that have access to it. (WD, 2005)



Source: <https://www.workingdata.co.uk/spreadsheets-vs-databases-round-1-multi-user/>

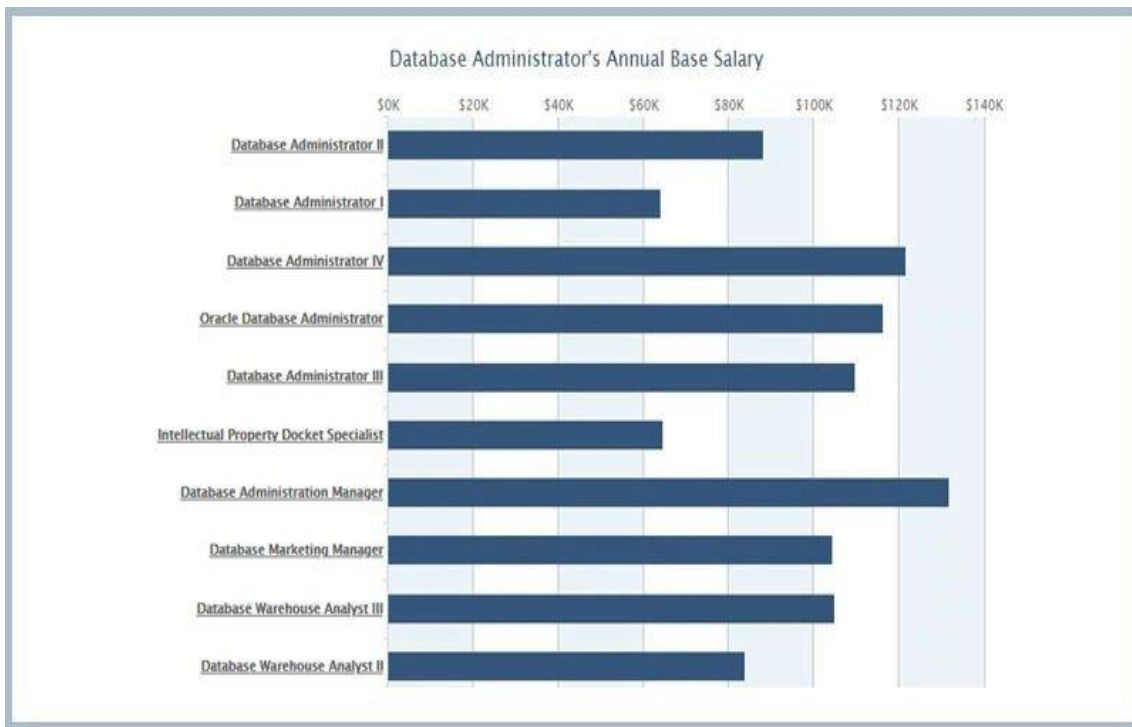
Databases offer businesses a smoother operating work situation. The implementation of a database management language such as SQL (Structured Query Language) allows businesses to access and modify data that is stored in a relational database.

Databases are constantly being used and accessed in new ways. With all the advantages that databases offer, uses will continue to grow. The accessibility, versatility and efficiency that a database can provide when paired with a DBMS is the reason why so many successful businesses are using them to this day.

1.7 Costs and Risks of Database Approach

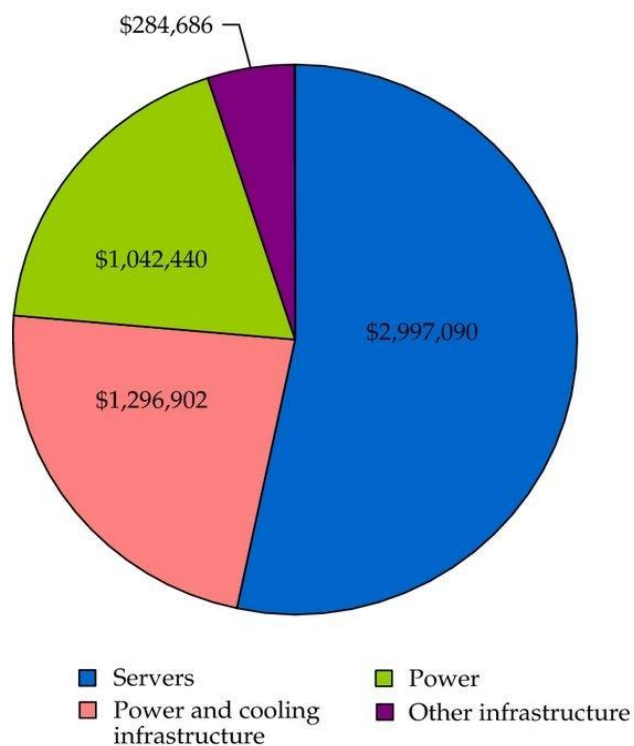
There are obviously many advantages that benefit those who implement a database approach. Organization, efficiency and structure are all some positive elements that can be attributed to the database approach. However, there is always give and take, and there are some risks and costs involved with the database approach as well.

For example, when you decide to implement a database system, you now require personnel who know how to implement and maintain this system. This will most definitely be a significant cost that will be directly attributed to the implementation of the system. There also lies the cost in training individuals who may be new to your system that has already been implemented, and this will not be cheap either. The graph below illustrates the cost one may be dealing with personnel-wise on an annual basis when implementing a Database system.



Source: <https://education.oracle.com/oracle-dba-salaries-guide>

Another significant cost of the Database approach is the cost of installation and maintenance. When implementing a new database system, it is costly to pay personnel to install and operate it, especially if it is a large and complex database. Installation isn't where the cost stops; maintenance is needed in order to keep that system running, and as you want to expand and maintain, over time, you will require additional hardware. The chart below displays the different costs of operating a given datacenter per month.



Source: https://www.researchgate.net/figure/Monthly-costs-of-the-data-center_fig3_258385511

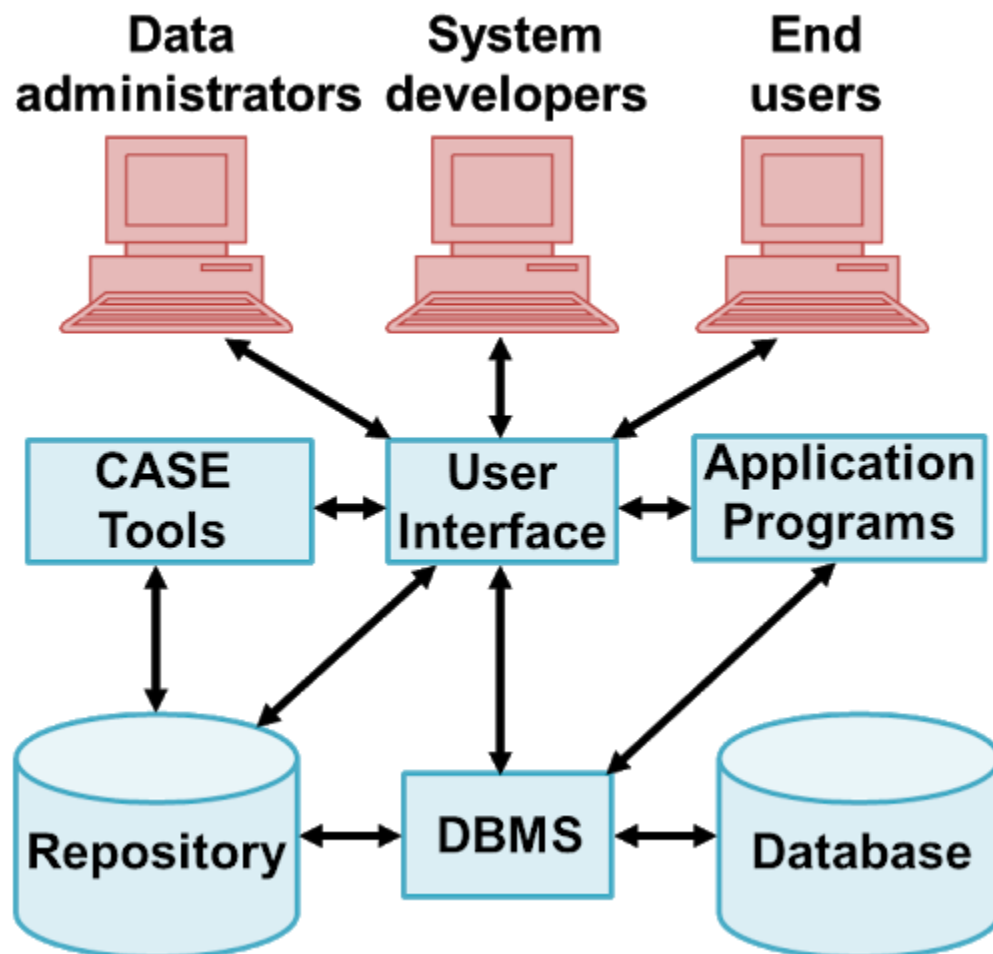
One must also account for the cost of migration: the cost to transfer the data and functionality of the previous file system over to the new database system accurately and without loss. It may seem as though it is a simple concept, but it is very difficult in a lot of cases and thus costs a substantial amount of money and time to execute.

There are also the costs and risks involved in needing specific backup and recovery systems. In a shared corporate database, there will be large amounts of data being stored. However, there must be backup data in case of software, hardware, or human error. If the data is not backed up, depending on the use of the database, the results could be catastrophic.

Lastly, there is organizational conflict. When implementing a large database with large complexity, it is common for people within the organization to have opposing views on how the data should be stored and how the system should be running. The costs and risks here lie in hiring strong leadership. In order to reach agreements on data definitions and how responsibilities are delegated for accurate data maintenance, the leadership must be strong and defined.

1.8 Components of a Database Environment

A database environment has 5 major components for functionality. The components necessary are: people, hardware, software, data, and procedures (OwlGen, 2019).



Source: http://cdn.wagmob.com/subject/G124/html/introduction00to00dbms_1.html

When it comes to people, there are different roles needed to help build the overall database environment. These roles would include but are not limited to system and database administrators, database designers, programmers, analysts, and end users (OwlGen, 2019).

The system administrator is in charge of setting up and managing the system and server. They are needed to make sure there are no server crashes or any missing information within the database (Gite, 2014). Database administrators ensure the physical database is working properly through monitoring the performance and also managing security access and other standards (“What are the functions of a database administrator”). Database designers and programmers code all queries, relationships, and data and make sure they are stored properly within the database management system. Analysts review all the data the designers and programmers have implemented. Finally, end users are the ones that utilize the database management system and make the system more usable for other users.

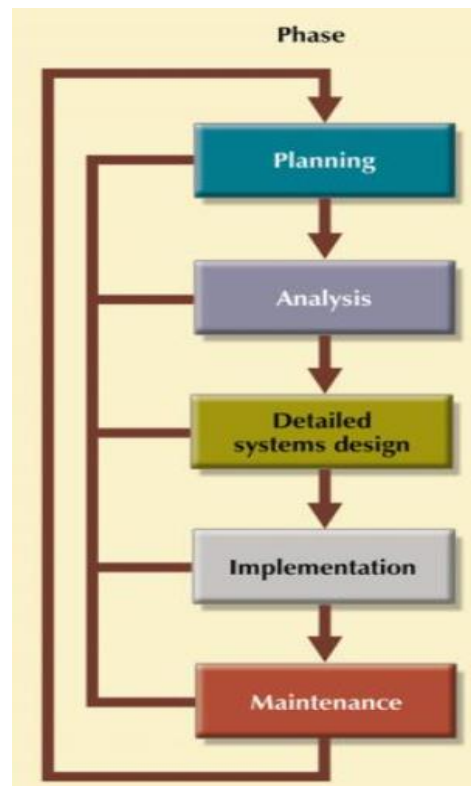
Hardware and software are the items that make the database environment come to life. Hardware includes the actual computer itself and any sort of networking components needed. Software includes the operating system and any sort of programs needed to build and administer the database (OwlGen, 2019).

Finally, data and procedures go hand in hand. Data includes things like the actual database needed to function in the environment as well as any business procedures and/or rules that manage the system. The procedures are implemented to structure the overall design on how the database should work and regulate all the data that should be going in and coming out of the database (OwlGen, 2019).

1.9 Database Systems Development Life Cycle

The database life cycle (DBLC) consists of six phases. These phases include database primary study planning, analysis, detailed System design, (prototyping), implementation and loading, testing and evaluation, operation, maintenance and evolution.

In the database primary study, the researcher examines the current systems operations in the company to determine how and why the current system isn’t sustainable. The objective of this study is to analyze the company status, define problems and constraints, define purpose, and define the scope and boundaries. Each section can be broken down in order to further understand the usefulness behind creating this study.



The Database Life Cycle (DBLC)

Analyzing the company situation	Pertains to defining the general conditions within a company, including its organization structure and its mission. In order to correctly do this, the designer must discover what the company's operation components are, the way they function, and how they interact.
Defining Problems and Constraints	Pertains to the discovery of issues within the company, formally and informally. These problems may appear unstructured; however, problems are usually connected, allowing the designer to overcome them by the end of the process.
Defining Objectives	Is a part of the new proposed database system showing that it is designed to solve the major problems identified previously?
Defining the Scope and Boundaries	<p>Pertains to the engineer recognizing the existence of their limits: scope and boundaries. The system's scope shows the extent of the design according to the requirements. The system also is connected to limits known as boundaries which are external. These boundaries are set by the accompanying hardware and software.</p> <p>Database design is the second phase focusing on the design of the database that supports company operations and objectives in the future. This can be viewed as the most critical DBLC phase.</p>
Implementation and Loading	Pertain to a series of instructions when dealing with the creation of tables, attributes, etc. in the domain. In this phase, the design specifications are installed, creating the exact database required by the parent company. This can be done in 3 phases.
Install the DBMS	Installing a new instance of a DBMS in the system on a server.
Creating the DBMS	Creates the table spaces and file groups accompanied by the database.
Loading and Converting Data	<p>After the database is created, the data must enter the new tables. This requires them to be merged and imported from other databases or the ones previously used in order to ensure the same data is relayed into the newer, better system.</p> <p>Testing and evaluation pertain to the decision made to ensure integrity, security, performance, and recoverability of the database. Following the plans laid out previously, this fine-tunes the database to ensure that it performs as expected. This phase is also divided into three phases, making it easy to follow and accurately test the functionality of the database.</p>
Test the Database	During this step, the database is tested to ensure it has the integrity and security required by the company. This is enforced through the proper use of primary and foreign key rules.
Fine-Tune the Database	This is the editing of the database with the results of the previous step in mind. If no fine-tuning is required, this step can be skipped.
Evaluating the Database	The database must be reviewed thoroughly to ensure that the data contained is protected against loss, promoting the use of a backup.
Operation	The second to last step identifying that the database is fully functional. At this point, the database is complete, and the new system has space to evolve as needed by the developers.

Maintenance and Evolution	The final step. This step is directed by the database administrator allowing them to perform routine maintenance activities regarding the database. Some of these activities include Backup, Corrected Maintenance, Adaptive Maintenance and the Assignment of access permissions to welcome new users and edit old users.
----------------------------------	--

All together these steps make up the Database Life cycle and ensure that a fully functional database is created, allowing for around the clock maintenance within the company and promoting a highly efficient system the meets the guidelines presented at the beginning of the process.

1.10 A Database and its Data Relationships

Database tables are structured to store data, but a database is not complete unless it also shows the relationships among the tables. To see why this is important, examine Figure 1-4 below (Kroenke, Auer, Vandenberg, Yoder, 2018) the database includes all of the basic data shown together with a GRADE table. Unfortunately, the relationships among the data are missing.

In this format, the GRADE data are useless. It would be the equivalent to a sports commentator who simply announced: “Now for tonight’s baseball scores: 2–3, 7–2, 1–0, and 4–5.” The scores are useless without knowing the teams that earned them. Thus, a database contains both data and the relationships among the data.

This demonstrates is imperative characteristic of database processing. Each row in a table is distinctively identified by a **primary key**, and the values of these keys are used to create the relationships between the tables. For example, in the STUDENT table StudentNumber serves as the primary key. Each value of StudentNumber is unique and identifies a particular student. Thus, StudentNumber 1 identifies Sam Cooke. For example, ClassNumber in the CLASS table identifies each class. If the numbers used in primary key columns such as StudentNumber and ClassNumber are repeatedly created and assigned in the database itself, then the key is also called a **surrogate key** (Kroenke, Auer, Vandenberg, Yoder, 2018).

STUDENT		
Field Name	Data Type	Description (Optional)
StudentNumber	AutoNumber	Surrogate key for STUDENT
LastName	Short Text	Student's last name
FirstName	Short Text	Student's first name
EmailAddress	Short Text	Student's email address

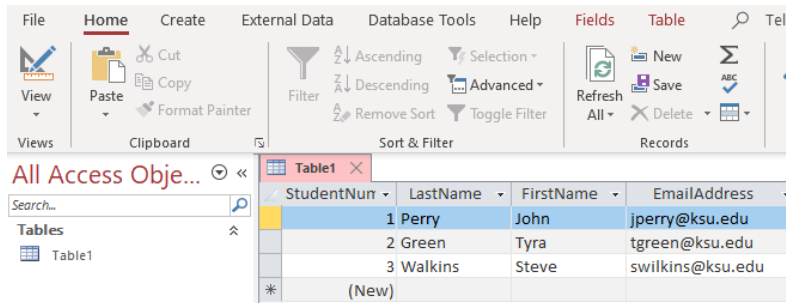
Figure 1-1: Sample Microsoft Access Student Record

Figure 1-2 shows each row in a table in specifically known by a **primary key**, and value of those keys that are used to create a relationship between the tables, such as student **IDNumber** (primary key). If the numbers used StudentNumber and ClassNumber column and generate and assigned in the database, then the key is also called a **surrogate key**.

Table1						
StudentNun	LastName	FirstName	EmailAddress	IDNumber	ClassNumbe	Click to Add
1	Perry	John	jperry@ksu.edu	0006783490	CSE1100	
2	Green	Tyra	tgreen@ksu.edu	0003214567	ENG2250	
3	Walkins	Steve	swilkins@ksu.edu	0009525412	CS4200	
(New)						

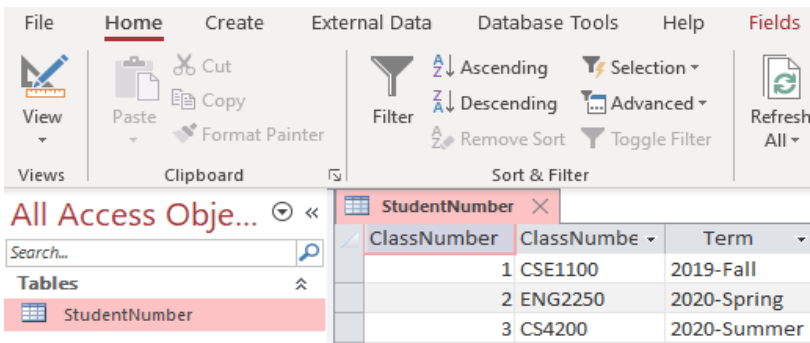
Figure 1-2 The Primary key and Surrogate key

In the table below shows when more than one column in a table are merged to form of the primary key, is known as a **composite key**. In the GRADE column, StudentNumber and ClassNumber each now serve as a foreign key. A **foreign key** provides a **relationship** or link between two tables. Figure 1-3 shows a **Microsoft Access 2016** point of view of the tables and their relationships.



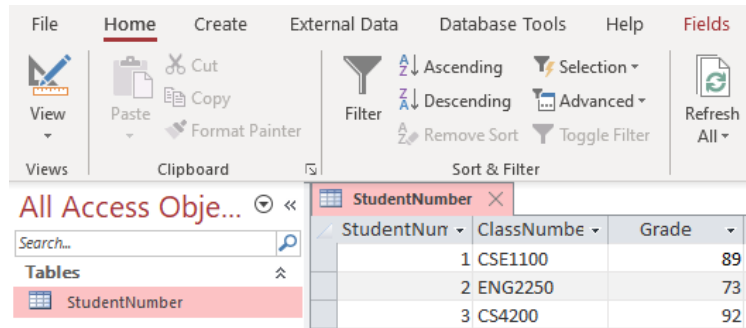
StudentNum	LastName	FirstName	EmailAddress
1	Perry	John	jperry@ksu.edu
2	Green	Tyra	tgreen@ksu.edu
3	Walkins	Steve	swilkins@ksu.edu
(New)			

StudentNumber Table



ClassNumber	ClassNumbe	Term
1	CSE1100	2019-Fall
2	ENG2250	2020-Spring
3	CS4200	2020-Summer

ClassNumber Table



StudentNum	ClassNumbe	Grade
1	CSE1100	89
2	ENG2250	73
3	CS4200	92

Figure 1-3: The Grade table with *foreign keys* – link to Student ClassNumber Table

Single-User and Multi-user Database Applications

Figure 1-4 shows the greater database application, part of a **customer relationship management (CRM)** system, which manages customers and their contacts, purchases, support requests, and so forth. The CRM system uses software to support a larger company, which may include anywhere from 500 rows to 10 million or more.

An **enterprise resources planning (ERP)** system is an information system that affects every department in a company, including sales, inventory, planning purchasing and other business purposes. **SAP** (System, Applications & Products in Data Processing) is the vendor used with ERP applications for large companies.

Application	User	Number of Users	Standard Size	Comments
Customer appointment (Doctor dentist)	Manger	20-15	500 rows	Marketing software
Customer relationship Management (CRM)	Senior Manger	10-15	10 million rows	Vendors applications such as Oracle
Data mining	Business Analysts	1-5	1000 to million rows	Data extracted and use by statistical data mining tools.

Figure 1-4 shows a larger database application

What is Microsoft Access?

Microsoft Access is not just a database management system (DBMS) but is also a **personal database system**. Microsoft Access is a combination of the relational Microsoft Jet Database engine with a **graphical user interface (GUI)** and software-development tools.

Microsoft Access is one of the office suites that is intended for individuals and small works groups such as interact with application through data entry process forms, generate reports, run the queries.

1.11 - Concise Summary:

Databases are the foundation of structuring data. When they were first implemented in the 1960s, data became easier to manage and structure. A database is now necessary to hold and manage user data and other personal information. Organizations utilize a database management system, or DBMS, for manipulating and storing the data into the databases while also managing the relationships between the data itself.

As computers and other technology began to become introduced, it became much simpler to manage data and store it in the DBMS. A database has many advantages, including efficiency, versatility, categorization, and organization to name a few. However, there are associated costs and risks to databases. The organization is now required to put funding towards training employees in managing and updating the DBMS as well as for general upkeep of the management system in order for it to remain stable. Various components of the DBMS to enable this include the people, the hardware, the software, the data itself, and the procedures needed to keep the database organized and well managed.

Finally, the life cycle of the database can be defined by six main phases: database initial study, database design, implementation and loading, testing and evaluation, operation, and maintenance and evolution. It is important to consider what needs to be implemented in a database, but also what role it can play for based on the needs of the organization

1.12 Extended Resources

- ❖ This link gives a detailed description on how database systems are managed and worked through. The speaker brings emphasis on the ER diagram and their relationship in structuring databases and queries. They also discuss database implementation in SQL Servers and briefly define what SQL (Structured Query Language) is.

<https://www.youtube.com/watch?v=n75iPNrzN-o>

- ❖ This article briefly describes the seven commonly used types of database management systems, explains the origins on how they were structured, and describes how each DBMS is used.

<https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/>

- ❖ This article explains the seven best practices on how to protect and secure databases from hackers and other attackers.

<https://www.esecurityplanet.com/network-security/6-database-security-best-practices.html>

- ❖ This article briefly explains some of the common malpractices in designing a database and database management systems.

<https://www.toptal.com/database/database-design-bad-practices>

- ❖ This article goes into some of the specific types of user interfaces in database management systems. They are designed by mostly UI developers to utilize the information given in the database. They also explain each of the different types to give an idea of which is the best for the needs of each DBMS.

<https://www.geeksforgeeks.org/interfaces-in-dbms/>

- ❖ Identifying Database Table Relationships

One of the huge advantages of a relational database is that, once you have your data held in clearly defined, compact tables, you can connect or relate the data held in different tables. There are three types of relationships between the data you are likely to encounter at this stage in the design: one-to-one, one-to-many, and many-to-many. To be able to identify these relationships, you need to examine the data and have an understanding of what business rules apply to the data and tables. If you're not sure, it can be helpful to meet with someone who does have a thorough knowledge of the data.

<https://condor.depaul.edu/gandrus/240IT/accesspages/relationships.htm>

- ❖ Deciding on Tables and Fields for your Database Design:

Each table in your database should hold the information on one subject. You might think of a subject as a collection of related information with common characteristics. For example, if you were creating a database to hold information about the operation of your ice cream stand, you might have an IceCream table. If you decided to sell sundaes as well as cones, you might add a Toppings table. Then, to associate ice cream and toppings in particular combinations and record the prices, you might add a Sundaes table.

<https://condor.depaul.edu/gandrus/240IT/accesspages/tables-fields.htm>

1.13 References:

Introduction to the Module. (n.d.). Retrieved from

https://www.cs.uct.ac.za/mit_notes/database/htmls/chp01.html#risks-of-the-database-approach

Costs and Risks of Database Approach. (2010). Retrieved from <http://www.smartclass.co/2011/02/costs-and-risks-of-database-approach.html>

Hooda, S. (2019, September 17). What are the Advantages of a Database Management System? Retrieved from <https://www.goskills.com/Development/Articles/Advantages-of-database-management-system>

Keuffel, W. "Battle of the Modeling Techniques." *DBMS Magazine* (August 1996).

Kroenke, D., Auer, D., Vandenberg, S., Yoder, R., Database Processing, Fundamental, Design, and Implementation, Fifteenth, Edition, 2018.

Kroenke, D. "Waxing Semantic: An Interview." *DBMS Magazine* (September 1994).

Kioko, O. N., & Bilal. (2011, March 28). Disadvantages of conventional file-processing system Retrieved from <http://punarvasi.com/disadvantages-of-conventional-file-processing-system/>

Multi-user environments: Spreadsheets vs Databases. (2016, April 11). Retrieved from <https://www.workingdata.co.uk/spreadsheets-vs-databases-round-1-multi-user>

Oracle Database Administrator (DBA) Salary Guide. (n.d.). Retrieved from <https://education.oracle.com/oracle-dba-salaries-guide>

O. (2019, December 20). What are the Components of Database System Environment? Retrieved from <https://www.owlgen.in/what-are-the-components-of-database-system-environment/>

Parody, L. (2019, September 27). Databases for Frontend Developers. Retrieved from <https://medium.com/@lizparody/databases-for-frontend-developers-2a4e9f16b7b4>

Rahul, & Rahul. (2019, December 27). File Processing System - File System - Disadvantages of File Processing. Retrieved from <https://www.tutorialcup.com/dbms/file-processing-system.htm>

Sir, B. (2017, July 26). Disadvantages of File Oriented System. Retrieved from <http://niravbaldha.blogspot.com/p/file-oriented-system.html>

Thiru. (n.d.). Thiru. Retrieved from <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/506-database-development-life-cycle.html>

Vivek Gite. (2014, January 19). What is The Role Of the System Administrator? Retrieved from <https://www.cyberciti.biz/faq/what-is-the-role-of-the-system-administrator/>

WD, (2005) Retrieve from <https://www.workingdata.co.uk/spreadsheets-vs-databases-round-1-multi-user/>

Wang, X., Wang, Y., & Zhu, H. (2012). Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its Genetic Algorithm. *Mathematical Problems in Engineering*, 2012 , 1–16. doi: 10.1155/2012/589243

What are the functions of the Database Administrator. (n.d.). Retrieved March 22, 2020, from http://www.pkirs.utep.edu/cis4365/Tutorials/Database%20Administration/8.00700/1_multipart_xF8FF_2_tutorial.htm

What is a database? (n.d.). Retrieved from <https://www.oracle.com/database/what-is-database.html>