Modules_1- Basic Concepts – SQL, NoSQL, & Access

1.1 Introduction and Background

We live in competitive environment, where the metadata, data, and information with its efficient management is the most critical objectives of most organizations. Database systems with its tools, such as Structure Query Language (SQL) and No SQL create the efficiency, consistency, and reduce redundancy were to simplifies the managing of the data by extracting useful information in a timely manner. Therefore, the success of organization is dependent on the reliable tools and technologies to provide accurate, reliable in a timely manner to business to operate and make sufficient decisions.

This allowed to keep 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.

With knowing that billions of people using Databases in 21 century to store information, they need reliable and efficient system such as SQL, relational database, and NoSQL and Non-relational database to meet the rapidly growing need to handle large amount of data. 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.

1.2 Relational (SQL) and Non-Relations (NoSQL) Database Systems

There are two primary types of Databases

- The SQL relational database system
- The NoSQL Non-relational database system

A rational database was developed/proposed by E. F. Codd in 1970 where focused on relational model of data and used to maintain relational database management systems (DBMS). It is traditional structured database regarding names, numbers, and addresses, etc., (Figure_1). The popular tools that relational database uses are Structure Query language (SQL) where data storing is structured as rows and columns have very specific data reducing overhead, maintain efficiencies, and reduces data residencies. For Example, in Figrue_1 using UML (*Unified Modeling Language*) diagram STUDENT class may define the association with CLSS number and GRADE objects and with each other in different tables.

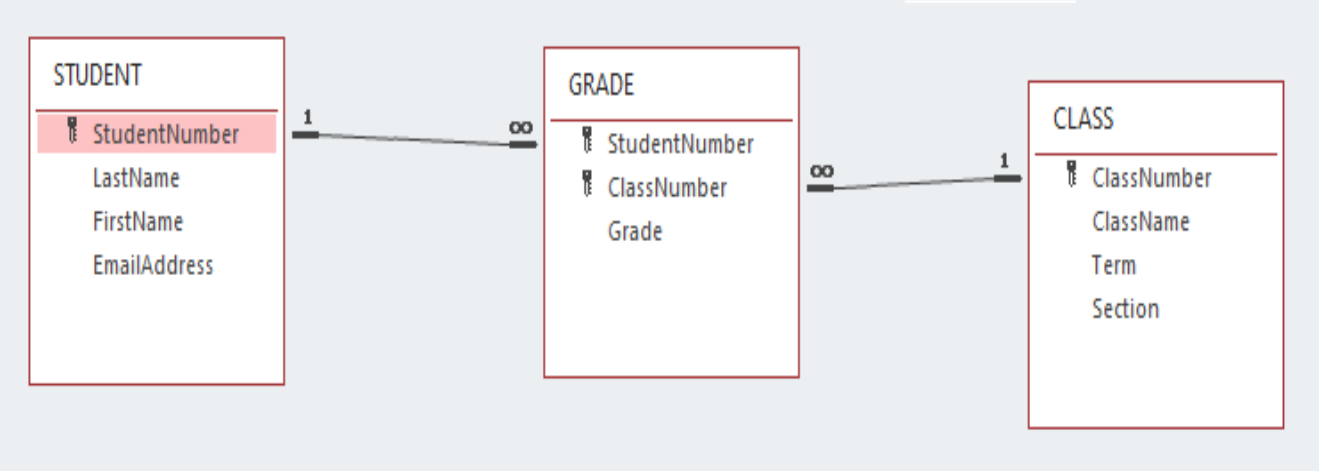


Figure_1 –Relational Database using Student Grade and Class Number

A non-relational database is more contemporary databased toward documentations and distributed process. There is common misconception that the NoSQL means "No SQL", but actually means "Not Only SQL", with the alternative to SQL database that apply SQL-like query concepts it covers any database that is not a traditional relational database with loosely ended structure. The motivation behind the No SQL relational database is simplified design, parallel scaling, and greater control over the availability of data specialized for types of data, which makes it more efficient and better performing than relational database.

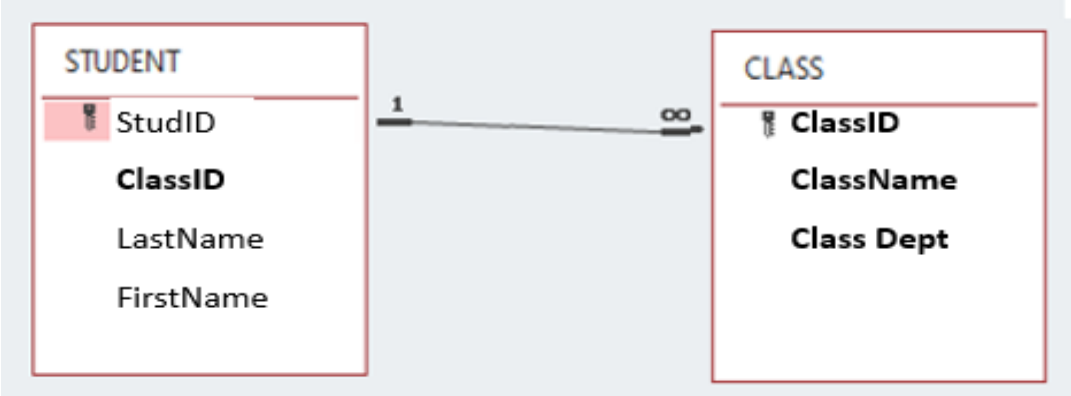


Figure_2 –Non-Relational Database using Student Grade and Class Number

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 Databases Structure

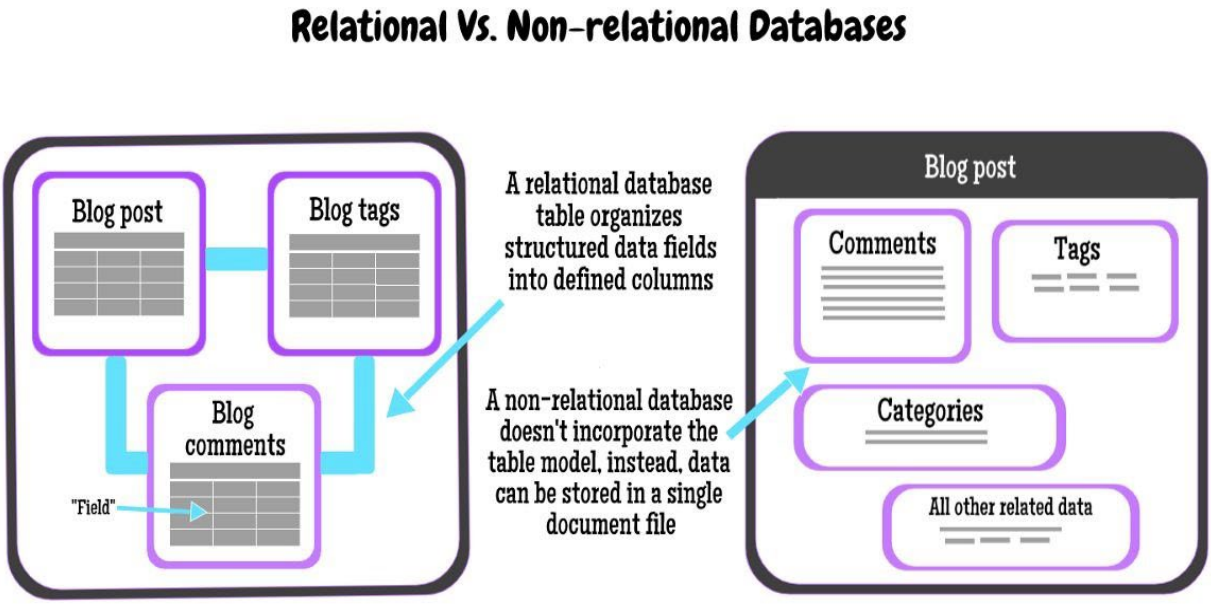
As Data is constantly being created, organized, and stored. With all this data being transferred and exchanged around the world, it is important to have an efficient and organized method to storing this data. This is where databases come in. Databases offer improved efficiency and versatility, they allow 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.



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)

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.

1.6 Characteristic of Relational (SQL) and Non-Relational (NoSQL) Database

SQL is one of the most used to interface with relational databases. The script SQL statements/ programming language that enable the users to design **query** with one or more tables to get the information they want. Therefore, **Relational databases** model data as record in rows and tables with logical links between them were more traditional types of numbers, addresses, etc.

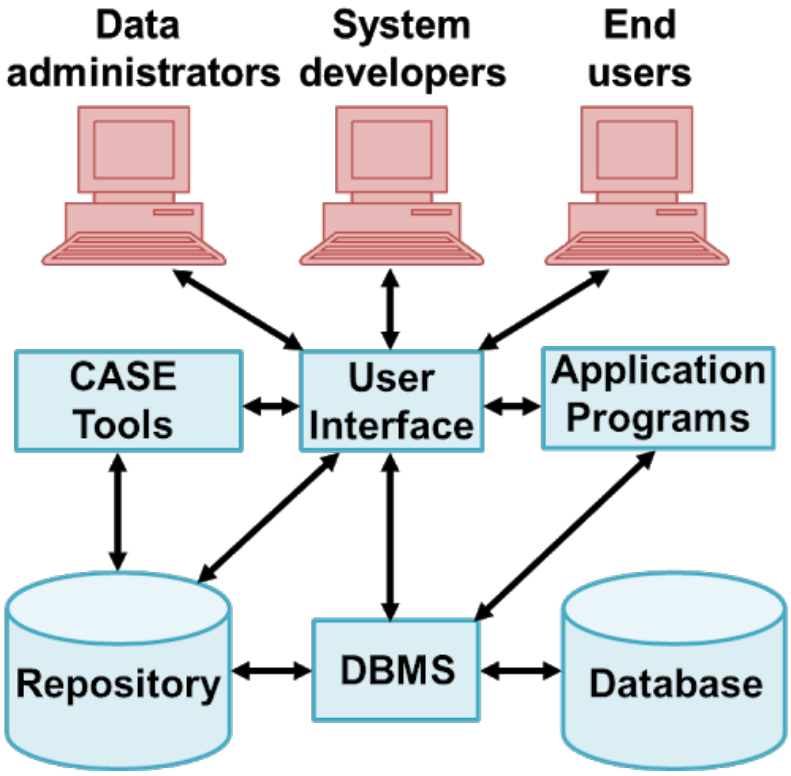
NoSQL are not faster than SQL but, it is all depend on the context. NoSQL covers any database that is not a traditional relational database management system (RDBMS).

- The motivation behind NoSQL is mainly simplified design,
- NoSQL databases are more specialized for types of data, which makes them more efficient and better performing than RDBMS servers in most instances.
- NoSQL seeks to break away from the traditional structure of relational databases.

Relational (SQL) Databases	Non-Relational (NoSQL) Databases
SQL are relational Database management system are Table-based, where each record is a structured row	**NoSQL are distributed database management system and specialized storage solutions**, e.g., document-based, key-value pairs, graph databases, columnar storage.
SQL are vertically scalable that typically scaled by increasing the power of the hardware, and predefined **schema** for each table, where changes allowed but usually blocking *(expensive in distributed and live environments)*	**NoSQL are Horizontally scalable**, as is scaled by increasing the databases servers in the pool of resources to reduce the load **Schema-less**, schema-free, schema change is dynamic for each document, suitable for semi-structured or **un-structured data.**
Not suitable for hierarchical data storage and can be used for complex queries	**Best suitable for hierarchical data** storage and not good for complex queries.

1.7 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*](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.

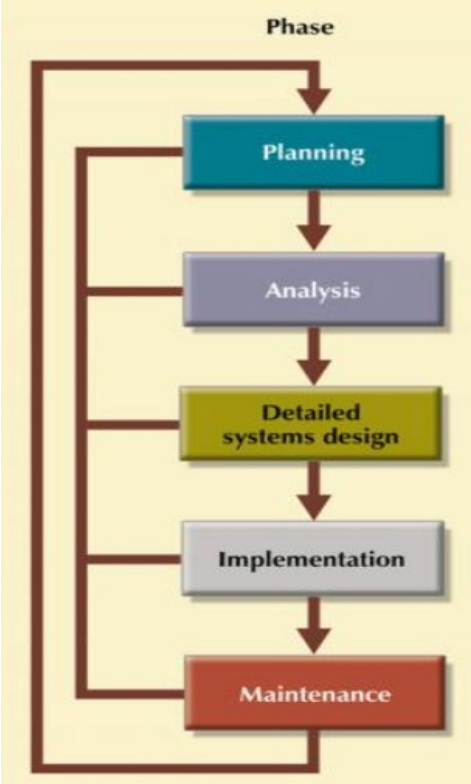
Hardware and software are the items that make the database environment come to life. Hardware includes the actual computer itself and any sort of networking components needed. Software includes the operating system and any sort of programs needed to build and administer the database (OwlGen, 2019).

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).

1.8 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)

1.9 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 3 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).

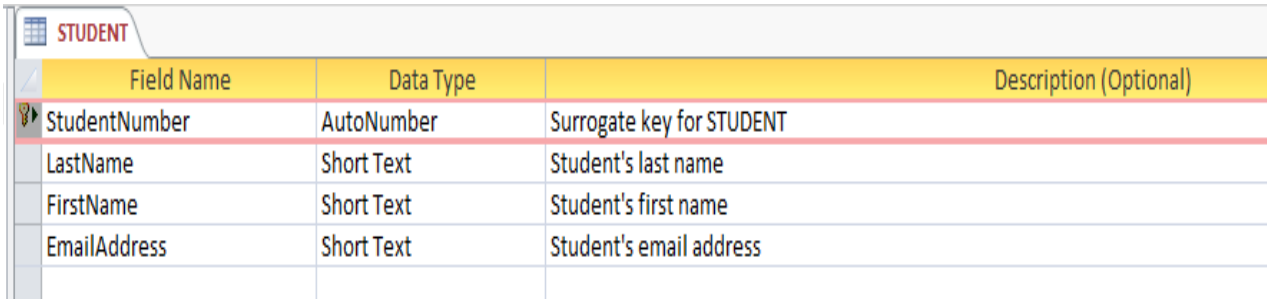


Figure 3: Sample Microsoft Access Student Record

Figure 4 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**.

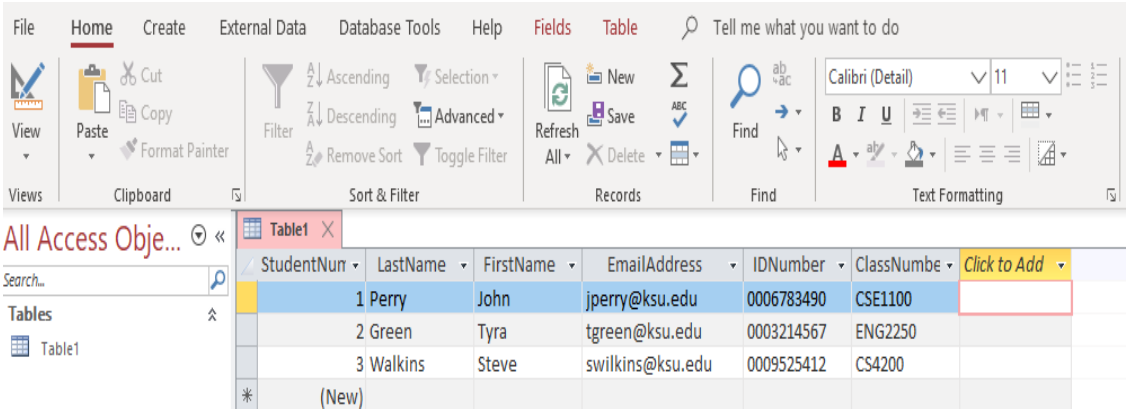
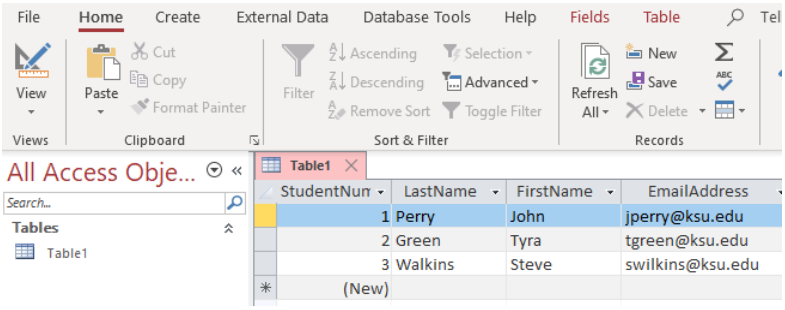
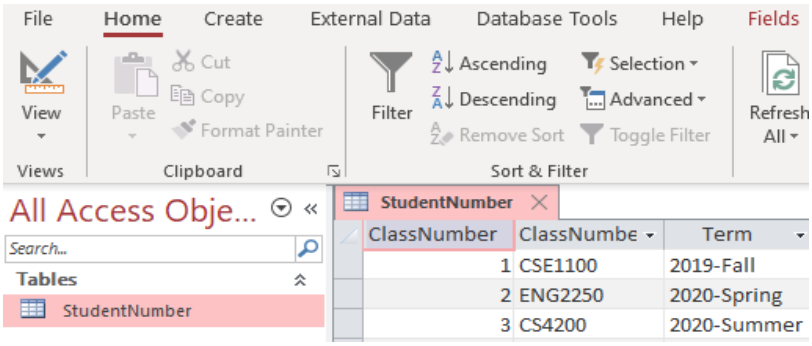


Figure 4 The Primary key and Surrogate key

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 5 shows a **Microsoft Access 2016** point of view of the tables and their relationships.



StudentNumber Table



ClassNumber Table

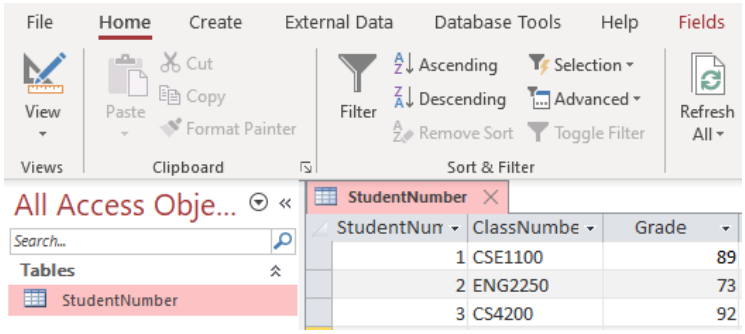


*Figure 5: The Grade table with **foreign keys** – link to Student ClassNumber Table*

Single-User and Multi-user Database Applications

Figure 6 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 6 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.

1.10 Concise Summary:

While relational or SQL databases can provide minimal data redundancy and stability, they are also traditional, structured method of storing data where data is structured in a horizontal way allowing for growth of data as it cause issues due to their structured configuration. The other hand, NoSQL are complex queries going to be required a great deal of data that requires quick scaling and will be no need for data analytics via BI (Business Intelligent) or Warehouse.

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

1.11 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-syste ms/

- This site explains the Battel of relational and non-relational databases | SQL vs NoSQL Explained. In this video I explain the difference and help you decide which database to use in which type of application. The Cloud Girl, June 2021.
  https://www.youtube.com/watch?v=ORxMMo7it_Y

- 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 Ice Cream 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

1.12 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

Dayley, B. (2020), No SQL with MongoDB in 24 Hours, Sam Teaches 24 Hours.

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-sy stem

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

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

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

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

Oracle. (2022). MySQL: MySQL shell 8.0: 3.1 MySQL shell commands. https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-commands.html

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

Singh, S. K. (2009). Database systems: Concepts, design, and applications. https://learning.oreilly.com/library/view/database-systems-concepts/9788177585674

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-develo pment-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