**JEREMY PETERSON DUTOYA**

**DR. MOHAMMAD AWWAL**

**PMBA-8311-RHA ENTERPRISE SECURITY**

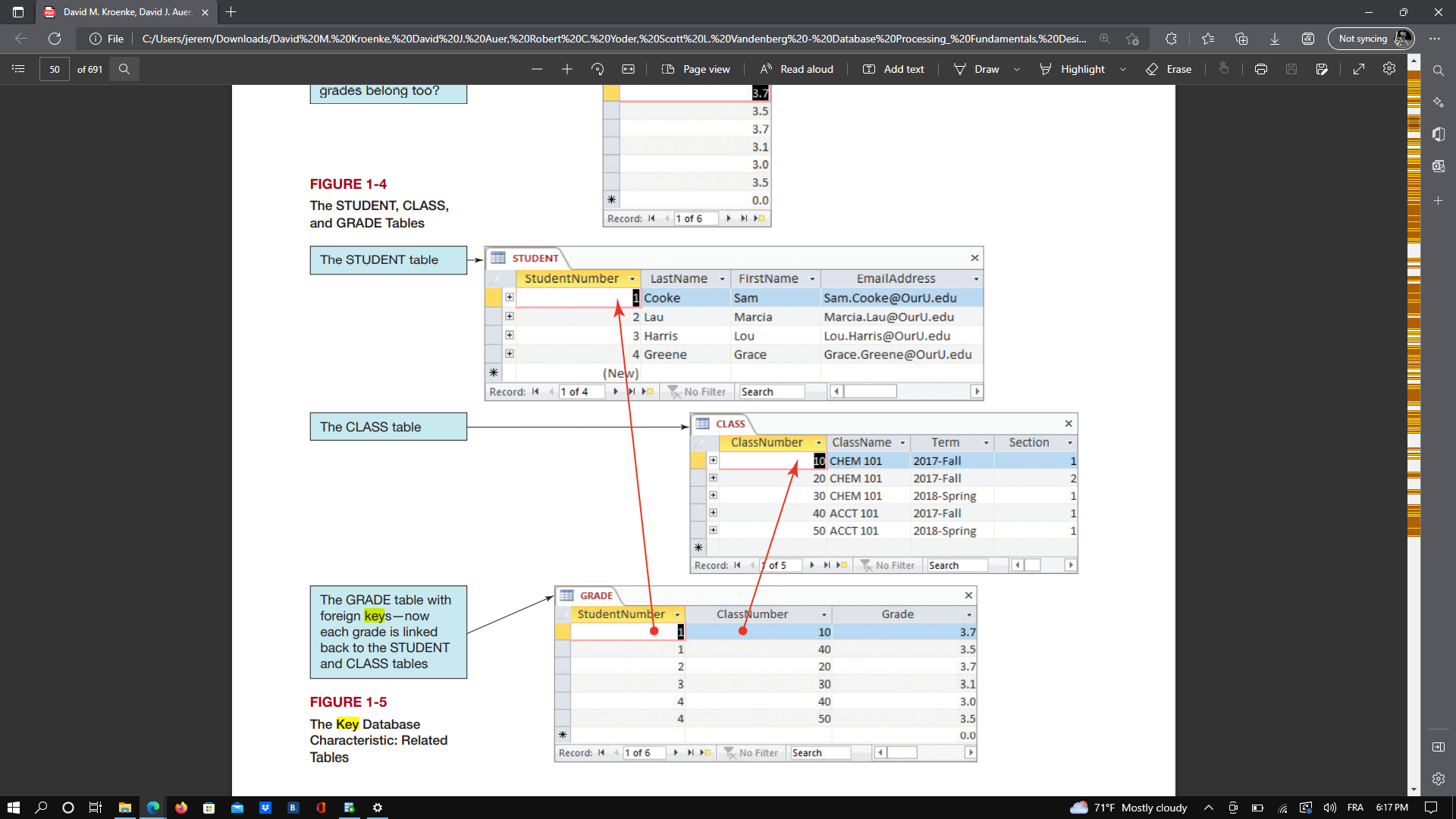
1. What is the purpose of a database and how does it address the problems related to file systems?

A database is a collection of interrelated information and data that has been structured for efficient retrieval. A database's main function is to store, retrieve, update and manipulate data, and thanks to the DBMS, it is possible to ensure an environment that is both convenient and efficient to do so. Before database systems were created, the traditional method was the processing of file systems. The traditional file-processing approach had numerous issues, including data redundancy, inconsistent data, isolated data, multiple users access problems, issues with the atomicity of the data, and poor data security and integrity. One of the biggest problems was rightfully data redundancy. Due to the fact that data is kept in several files, duplicate data may exist in each of these files. It is quite challenging to fix this error because each file is independent of the others, and if it is discovered, it will take time and effort to fix. It could also increase the risk of using obsolete data Thanks to the functionality of the DBMS, it is possible to add a constraint on certain attributes that will ensure that each record is UNIQUE and will make it impossible to add the same data twice: this saves both time and storage = save money for the company. To remain general and present the bigger picture, through the DBMS, important functions are provided to the database to address the above-mentioned problems. Thanks to these functions provided with the DBMS the database can be accessed simultaneously by several users, increasing the productivity through concurrent access; Data integrity is ensured because data must meet constraints (Primary Key, Null, UNIQUE, CHECK, FOREIGN KEY) that are imposed when the tables are created; Data security is improved as the data is kept safe from theft, modification, and destruction through role-based access control that gives the appropriate authorization to users.

1. Why do today's Web applications and smartphone apps need a database?

A database is essential to the creation of web and smartphone applications. In fact, it is probably the most crucial component when creating an application. "Data is the foundation of a web application" so you need a database to store it and a DBMS to manipulate it. User data, session data, and other application data are all stored there. The database serves as the main storage location for this information. Relational databases, object-relational databases, and NoSQL databases are just a few of the databases that web applications employ to store data. There are a ton of companies whose performance and development depend solely on the quality of their database. You see, databases are crucial for internet enterprises. These days, databases are utilized for various things, including keeping track of orders and inventories, creating consumer profiles, and monitoring financial records. The majority, if not the entirety of contemporary web apps are database-based. The database contains data about users, products, orders, and other things. A database is a central repository for user data and business logic, making it a crucial part of any website. If you have ever used Facebook it will be easier for you to understand: a database is used to record all of your posts, comments, "likes," and other information you give to the app such as pictures for example. Everything that you post is originally saved in the database before being shown on your feed. Same thing for Twitter, Instagram or any other app you can think of: a database contains all of the user tweets, pictures, likes, retweets, scores, games, purchases, saved items, preferences, drafts, etc... Every important component of the application is initially being saved in the database before being presented to you.

1. Use the STUDENT and GRADE tables in Figures 1-5 (Chapter 1).



* 1. What is a primary key and what are the primary keys of the tables? Do you think that any of the primary key can be surrogate keys? Are any of these primary keys composite keys?

SQL databases place a great deal of emphasis on primary keys as they provide each row in a database table with a special ID. You can use a column to generate a primary key once you've found one that differs in value for every row in the database and that can serve to serve to identify each row of the table. Primary keys are also very important for the concept of normalization that we will learn later in the class For now let's retain that the primary key can be thought of as the column-based "main identifier" for each row in the database. When using primary keys, a technical check is made to see if the primary key column in each row contains a UNIQUE and NOT NULL value. For instance, the primary key constraint will reject a new row if you attempt to insert it with a duplicate value in the primary key column. Primary keys are very important to ensure the concept of data redundancy and integrity that I explained in the previous question. In addition, the key is also known as a surrogate key if the numbers used in primary key columns are automatically created and assigned in the database itself.

→ Primary key for the Student Table is StudentNumber (is also a surrogate key)

→ Primary key for the Class Table is ClassNumber (is also a surrogate key)

→ Primary key for the Grade Table is also a Composite key of the Foreign Keys StudentNumber and ClassNumber.

* 1. what is a foreign key? Which table contains the foreign key?

In SQL databases, foreign keys play a critical role in enforcing data consistency. They often use primary keys to create a virtual bridge connecting two database tables together. There is much more to explain about the foreign keys: to keep it simple we will just explain that once the student and grade tables have a foreign key constraint, we may consider Student to be the parent table and Grade to be the child table. The database must make sure that each entry in the child table always corresponds to a valid record in the parent table. This is the underlying principle of all foreign key validations. To answer the question explicitly, in our example, the table containing the foreign keys is Grade.

1. What problems can occur when more than one user accesses a database? How does a DBMS address those problems?

There is always a danger that one user's work can conflict with another's when more than one user uses a database system, these problems have different names: the lost update problem, the concurrent update problem, and the inconsistent read problem. For instance, a problem arises when an entity is updated by one user but is used simultaneously by another user in a query before it is changed or returned to its original value. Imagine that one session is updating certain entries while another is performing the count function to some of the records. The aggregate function may determine some values before the data have been updated and then have a false result. To avoid such conflicts, specific concurrency-control procedures are used to coordinate activities on the database. First, there is a hierarchy: It takes a lot of time, in addition to technical expertise to support a database with a large number of users. Usually, a database administration office is in charge of providing support and has more authorization on the database system than anyone else. The database administrator is also referred to as the office manager. So this shows that there is a hierarchy. Second, concurrent transactions are those that are processing two transactions against a database simultaneously. Although it could appear to users that simultaneous actions are being completed concurrently, this is untrue since the CPU of the computer processing the database can only carry out one instruction at a time. Since only a small number of threads can be run concurrently, even more, recent multi-core CPUs are constrained in how transactions are processed. Transactions are typically interleaved, meaning that the operating system transfers CPU services between activities to complete a portion of each transaction within a predetermined time frame. This task switching is carried out so swiftly that two persons using laptops next to each other to process the same database can believe their two transactions are finished simultaneously, when in fact they are interleaved. Resource locking, which forbids sharing by locking data that is retrieved for updating, is one method of preventing parallel processing issues.

1. What is a Structured Query Language (SQL) and why is SQL important in manipulating data stored in a relational database?

SQL is an important component of a Database System because it is the programming language used to communicate with the Database and the DBMS. Given that all major relational DBMS products can understand SQL and that database systems frequently submit SQL queries to the DBMS for processing, SQL is an internationally recognized standard language. Due to its simplicity of use and the extremely effective way it queries and manipulates data, and completes a huge spectrum of additional functions to convert enormous collections of structured data into actionable insights, SQL has managed to remain a steadily common pick for database admin over the years. It has been included in many commercial database products because of this, including MySQL, Oracle, Sybase, SQL Server, among others. Because they don't support SQL programming, several non-relational databases, like MongoDB and DynamoDB, are referred to as NoSQL products. Although successive SQL versions are using different structure and syntax for important operations, fundamental commands like select, insert, update, and create are generally shared by all SQL versions.