Intro:
Data is any information that is collected. The data can be about the customers, products or any intermediary. Data collection is only the beginning of the challenge. Data needs to be parsed, maintained and stored as necessary. Additionally, database management has a lot to do with making sure that the data is compliant with all data collection and storage rules and regulations. Database management also includes making sure the data is clean and usable to be able to build models using the data that is being stored. Data can either be stored on-prem or on the cloud. Most modern day companies use cloud platforms such as Amazon Web Services (AWS), or Microsoft Azure to store their data. Data manipulation, storage and analysis have great prospects for companies of all sizes.

For our use case, I expect for us to have data including customer credentials (which we will use to validate customers and let them login), any files, or information they upload, including the text to speech information, or the drag and drop information. Database management will be a keystone for AAC's project, and its importance will only continue to grow in future semesters, where some of the collected data might be used to add additional useful features to the app we are now building.

Sources used:
https://www.informatica.com/services-and-training/glossary-of-terms/database-management-definition.html#:~:text=Database%20management%20refers%20to%20the,throughout%20the%20entire%20data%20lifecycle.

Resources for future research for Milestone 1:
1. https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system
2. https://www.codecademy.com/resources/docs/general/database
3. https://cloud.mongodb.com/v2#/org/652712fa4e8de465bc374416/projects
4. https://www.mygreatlearning.com/curriculum/database-management-system-courses

Advanced:
Database management has to do with keeping track of data, tracking data and maintaining the data. There are different types of databases:

1. **Relationship Databases**: This is tables in tabular form (with rows and columns). This type of data is usually manipulated using SQL. SQL stands for Structured Query Language. I will do a deep dive on SQL next week.
2. **Object Databases**: In these databases, data is stored as objects.
3. **Object-Relationship Databases**: This is a combination of relationship and object databases, wherein, data is stored in both tabular and object form.
4. **NoSQL Databases**: There are a few ways for these types of data to be implemented: Key-Value Store which is basically analogous to a hashmap. Document Store: This is where data is in the form of a file format. An example is a JSON file. Graph Database: This is where data is stored as a graph. Examples are google maps, mapping friend networks on social networking websites.

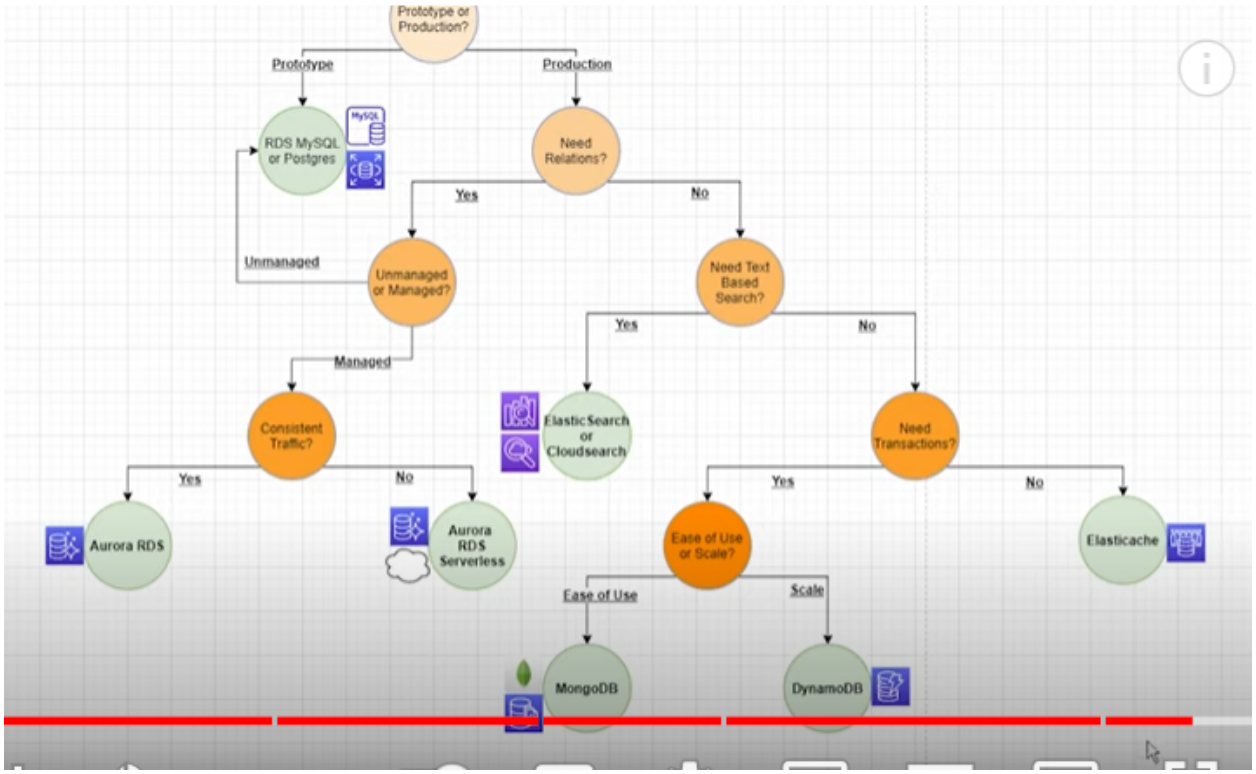
DBMS Course notes: (Link: https://youtube.com/watch?v=T7AxM7Vqvaw&list=PLdo5W4Nhv31b33kF46f9aFjoJPOkdlsRc)

- Data is a key aspect in today's day and age and there is data everywhere. Data can be in any form, including numbers, strings, videos, boolean values, etc.
- Database is a collection of related data: Related data is important here because a database is only when the data points are related to each other in some way. Databases should be able to be used to tell a story about a certain scenario. The more thorough and comprehensive data is, the more you will be able to derive from that data.
- One big problem with data is data **redundancy** where multiple files have the same exact data in them, making it repetitive and redundant. This can lead to **data inconsistency** which occurs when each data file has different data in them. For instance, when you need to update data, and there is data redundancy, if you do not update the data in all the places, this can cause data inconsistency.
- This is a big problem for huge databases. Issues like this can be addressed using Data Management System (DBMS). DBMS is a tool that can be used to manage data. The
- Database system = Database + DBMS
- DBMS and Database systems are used the most when there are huge datasets.
- Such databases and DBMS's are used in fields such as airlines, hospitals, schools, etc.
- Application 1: A university like Georgia Tech has so much data about students, including their personal information, their grades, their degree, the number of credits they have, the classes they took, etc. Similarly, a university system has many such universities in them. We can only imagine how much data this is and how much management this would require. A DBMS would be used in a case like this where there is so much information
- Application 2: An airline would have a passenger's personal details, flight number, number of booking, credit card information, etc. And it is expected by the airline to be saving/protecting the user information, especially sensitive information such as credit card information and social security numbers to be kept safely. A DMS would be necessary for cases like this as well.
- **Relational Database Management System:** RDBMS is a database management system that deals with relational data, which is stored in the form of tables, with rows and columns.

Lecture 2 (Link: https://www.youtube.com/watch?v=eK_umMYxZfM):
AWS databases:

- To pick which database to go with, do the following:
- Understand what the purpose of the database is:
- If the database is for prototyping, you can use: RDS MySQL or Postgres
- If the database is for production:
    - Do you need relations?
        - If yes: Managed or unmanaged solution? If unmanaged, use RDS MySQL or Postgres. If managed, is it consistent traffic? If yes, use Aurora RDS, if no use Aurora DS Serverless.

- If no: Do you need text based search: If yes, then ElasticSearch or CloudSearch. If no, then do you need transactions (update multiple tables all at once)? If yes, if you want ease of use, use MongoDB, if you want scale, use DynamoDB (a no SQL solution). If there are no transactions, use Elasticache.



SQL (links: https://aws.amazon.com/what-is/sql/ and https://www.youtube.com/watch?v=HXV3zeQKqGY)

- SQL is used to store and process information (also known as data) in a relational database, which is data stored as a table.
- SQL was invented in the 1970s, and was earlier called SEQUEL.
- SQL queries are basically SQL commands that do certain things such as render, update, or add data to the database.
- SQL has a parser. The parser updates certain words in the code with special symbols and then runs checks to see that the code is correct, the person running it is authorized to access the data they are trying to access.
- A Relational Engine then processes the query.
- What is a database?
- Massive units of data that can be used to tell stories. Examples: phone book, shopping list, to do list, Facebook's User base.
- Databases can be stored in these ways:
    - On paper
    - In your mind
    - On a computer
    - This powerpoint
    - Comments section of a youtube video.
- A key way to store data is in graphs where nodes represent states and lines (and arrows) represent actions that can be performed.
- A CSV file or a JSON file with data is viewed as a NoSQL Database.
- SQL: Structured Query Language.
- SQL is not a programming language but is usually viewed as one. You can use SQL to manage, and manipulate data (change it, delete it, retrieve it, etc.). You can create tables using SQL:
- INT: number
- DECIMAL(10, 4): Decimal value
- VARCHAR(100): String
- BLOB: Binary object
- DATE: Date in the form of YYYY-MM-DD

To **make a table**, you can use:

```
create table name_of_table ();
```

Here, create and table are keywords. All commands in SQL end with a ;
Ex:
Coding tutorial:

```
create table student (
        student_id INT,
        name VARCHAR(length of the name string),
```

```
    major VARCHAR(length of major string),
    PRIMARY KEY(student_id)
);
```

Inserting data into a table:

Coding tutorial:
insert into student VALUES(1, 'Jack', 'Biology');

This creates this table:

student_id	name	major
1	Jack	Biology

You can add constraints to say that certain variables cannot be certain values. For example: you can say:

name VARCHAR(20) NOT NULL

This basically means that name here cannot be equal to null.

Updating and deleting information:

UPDATE student
SET major = 'Bio'
WHERE major = 'Biology';

This sets the value of something when a certain constraint is true. This entire code block is one line since the semi colon is at the end of the three lines.

Querying Data:

SELECT * FROM student;

SELECT means that we are getting information
* means every

SELECT name FROM student;

would get back all the names of the students.