Introduction to Database & BigQuery Setup

---

1. Problem Statement
2. What is a Database?
3. Database Management System
4. Database vs. Excel
5. DB Schema Design
6. Data Types
7. Concept of Keys
8. Data Warehouses
9. Setting up BigQuery

Please note that any topics that are not covered in today's lecture will be covered in the next lecture.

Company Name: Amazon Fresh Market

Amazon Farmers Market (AFM) is a beloved community market that focuses on providing fresh, local produce directly from farmers to consumers.

Known for its commitment to quality and sustainability, AFM has become a vital hub for the community, offering a wide variety of fruits, vegetables, and artisanal products.

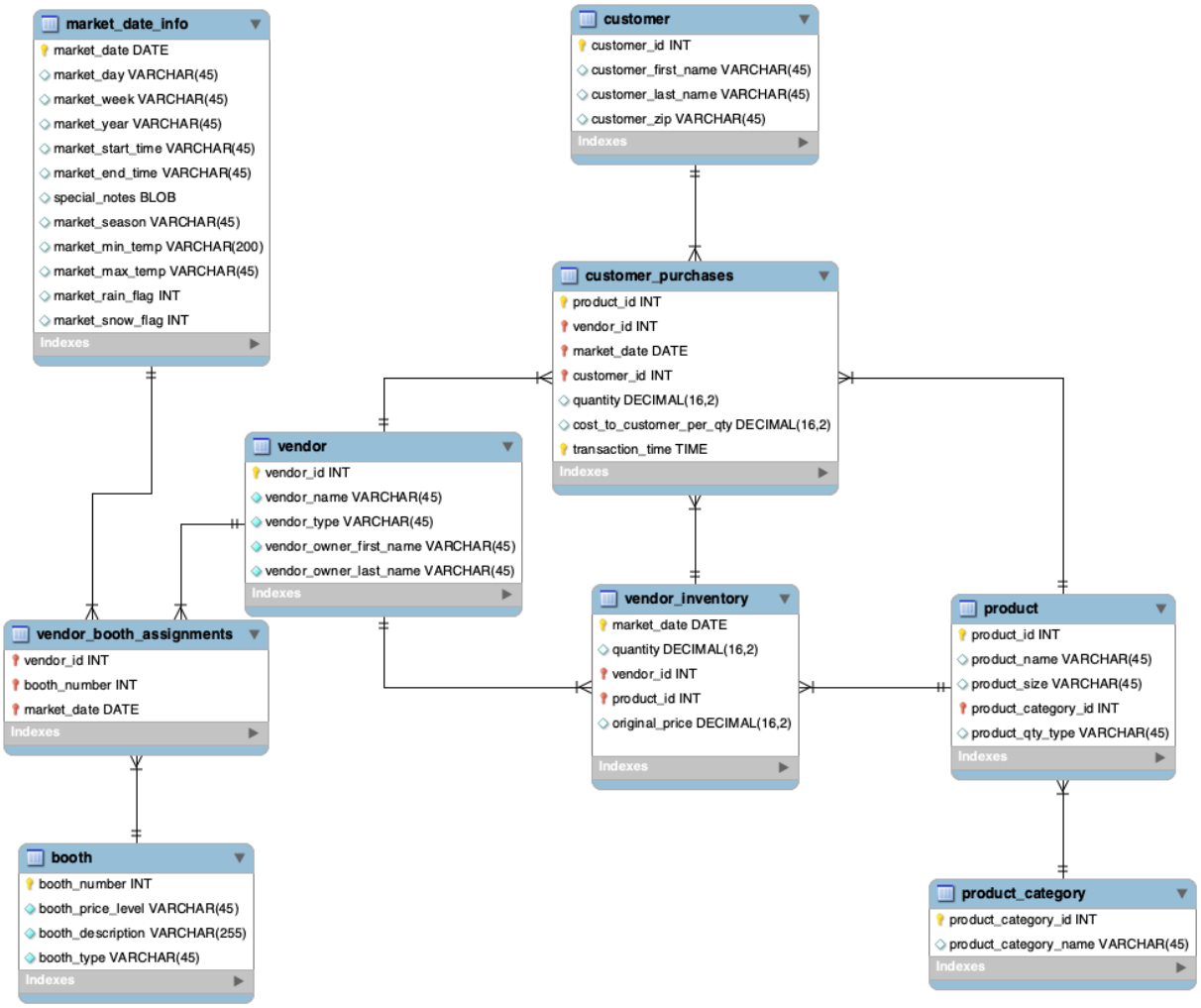
Despite its popularity, AFM faces increasing competition from large grocery chains that are expanding their local produce offerings.

To maintain its customer base and thrive in this competitive environment, AFM aims to leverage data analytics for a deeper

understanding of customer behaviour, vendor performance, market operations, and inventory management.

Problem Statement

- It's your first day as a Data Analyst Intern at **Amazon Fresh.**
- Amazon Fresh has recently decided to open big stores where farmers can directly sell their produce from their stores.
- **All the data is stored in a Database.**
- Your manager reaches out to you and gives you access to a **DBMS and a DB schema** that looks like this
- As an intern, you feel the need for some initial guidance before you can tackle the actual problem statement.
- Your manager takes the time to explain some foundational concepts related to databases and how data is organized and managed.
- Then he tells you that to maintain its customer base and thrive in this competitive environment, AFM needs to leverage data analytics for a deeper understanding of Customer Behavior, Vendor Performance, Market Operations and Inventory management
    - They aim to optimize inventory management, tailor marketing efforts, improve the customer experience for long-term success and solidify AFM's position as a vital community hub for fresh, local produce.



Formulating questions to be explored based on the data provided:

- What is data?
- Where is it stored?
- What is a Database?
- What is a Data Base Management System (DBMS)?
- What features should DBMS offer that overcome the limitations of Excel and Gsheets?
- What is a DB schema or an Entity-Relationship Diagram?
- What are tables, columns, and rows in a database?

Where is this data stored? - DBMS

Your manager tells you that the data is stored in a Relational DBMS called MySQL.

-> What is a DBMS?

- Any interaction you make with an app, **e.g., Amazon**, searching from the list of products and categories, wish-listing a product, adding to a cart, and placing an order.
- Whatever web pages we visit or applications we use, all of them use some backend system to collect all the data.
- **All of these backend systems are connected to Databases** that store all the interactions we make with their app.

DBMS consists of two phrases:

1. **Database** - a collection of interrelated tables -

2. **Management System**
    a. A set of operations that help in managing & manipulating the Database
    b. Example of the operations -
        i. CRUD - Create, Read, Update, Delete

In simple words, a **DBMS consists of related data and a set of programs to access & manipulate that data**

→ Now, the tables in which we store data are also called **relations.**

→ And thus, this type of DBMS, which stores data in a **tabular** format, is

called **Relational DBMS** or **RDBMS.**

→ Examples of **RDBMS** include:

- BigQuery
- MySQL
- Oracle
- SQLite
- Microsoft SQL Server
- PostgreSQL

Question: You should be wondering, why do we even need a **Database** if we already have **Excel / GSheets**?

1. **Scalability**:
    - Databases can handle large datasets and scale effectively as data size increases.
    - **Whereas Excel has limitations on the number of rows.**
2. **Performance**:
    - Databases are optimized for querying and retrieving data quickly.
    - In contrast, **Excel can become very slow** when performing calculations or filtering **large amounts of data**.
3. **Data Integrity**:
    - Databases enforce certain conditions (also known as constraints) ensuring that the data is accurate.
    - Excel, on the other hand, is more **prone to human errors**.
4. **Concurrent Access**:
    - Databases allow multiple users to access and modify data simultaneously.
    - Excel, on the other hand, can **lead to inconsistency in data due to multiple user access**.
5. **Security**:

    - Databases allow administrators to manage user access & modification rights.
    - Excel files, on the other hand, **can be easily shared and copied**.

Now that the purpose and meaning of DBMS are clear, the next question is **how to figure out what tables, columns, and rows are there in a relational database.**

That's where DB schema comes into the picture:

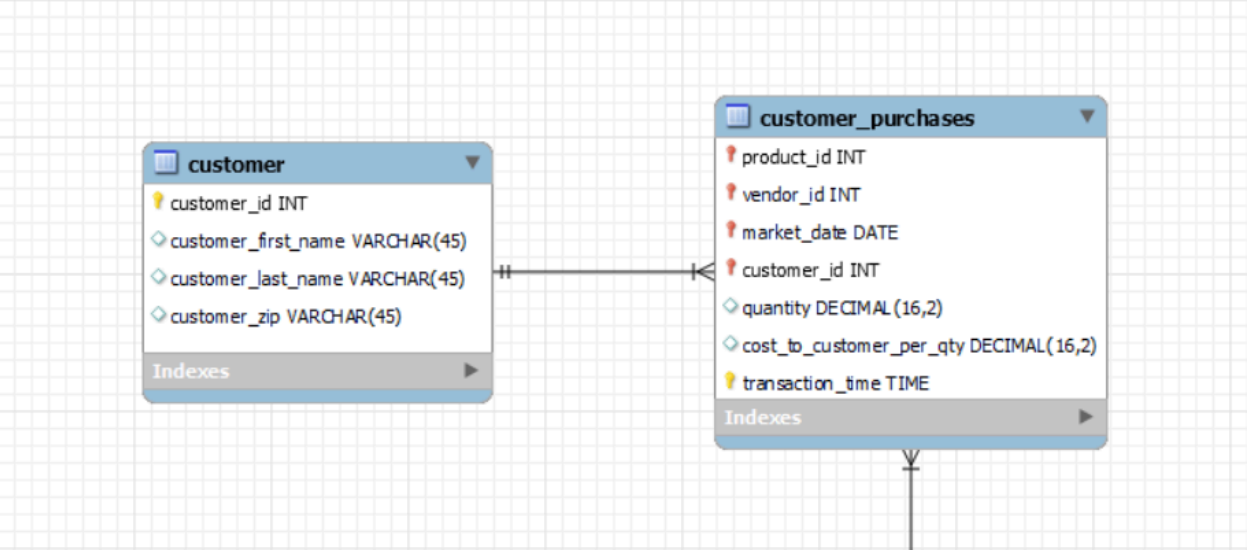
What is a DB schema or an Entity-Relationship Diagram?

- A database schema represents how the data is organized and provides information about the relationships between the tables in a given database.

- In our DB schema, those boxes that you see represent individual **tables/relations**.

- The lines connecting those tables are called **relationships**. We'll talk about different kinds of relationships later at some point in this module.

- Notice that along with the **column/attribute** names, a bit of extra detail is also provided. This is called the **data type** of a column.



- It depicts that for each column, and what type of data is stored in that column.

There are three main data types that we need to know about:

1. **string,**
2. **numeric, and**
3. **date and time.**

String → Char(), Varchar(), etc.

- Char is **Fixed** Length Character String
  - Eg. for Char(3) data type, a valid data entry could be IND, AUS, USA
  - Whereas inputs like INDIA, and IN will throw an error.
  - Because it'll require exactly '3' characters.
- Varchar is **a variable-length** Character String
  - Eg. for Varchar(5) data type, a valid entry could be IN, IND, INDIA
  - Whereas inputs like AMERICA, and AUSTRALIA will throw an error.
  - Because it'll accept only up to '5' characters.

Numeric → Int64, Float64, etc.

- Int64 is for storing an **integer value**
- Float64 is for storing a **floating point value**

Date & Time → Date, Time, Datetime, Timestamp

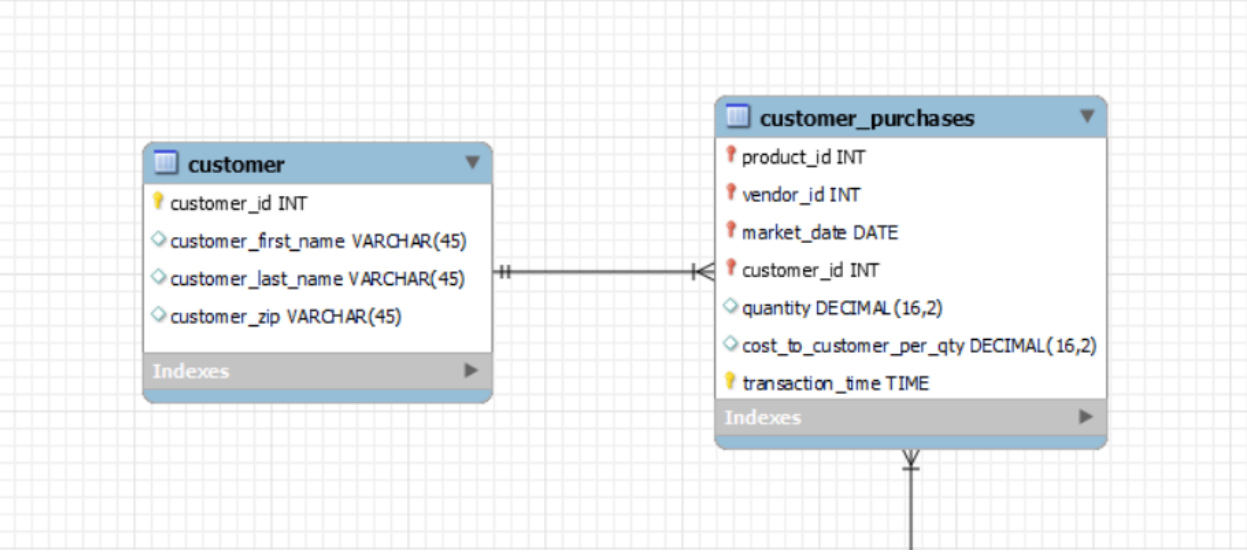
- Date data type is used for storing a date in the format YYYY-MM-DD.
    - Eg. 2023-05-17
- Time data type is used for storing time in the format hh:mm: ss.
    - Eg. 03:07:00
- Datetime is used for storing a combination of date & time in the format YYYY-MM-DD hh:mm: ss.
    - Eg. 2023-05-17 03:07:00
- Timestamp values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC).

NOTE:

- **In BigQuery,** the following documentation page can help you with these: Data types
- **In MySQL**, you can refer to this link to showcase different data types. https://www.w3schools.com/sql/sql_datatypes.asp

---

Concept of Keys



Now if you notice here, there is a key symbol present in both the tables next to the attribute name.

In an Entity-Relationship (ER) diagram, a key symbol is usually depicted as a unique identifier for an entity

It indicates that the attribute is a key, meaning it uniquely identifies each record within an entity.

What is a key?

A key in a database context is an attribute or a set of attributes that uniquely identifies a record within a table. There are a few types of keys, let's discuss them.

But first,

Why do we need a key?

Keys in relational databases help to specify constraints on the data in a table.

Let's understand the essence of different keys using the `**customers**` table given below.

customer_id	name	phone	age
1	Akon	9876723452	17
2	Akon	9991165674	19
3	Bkon	7898756543	18
4	Ckon	8987867898	19
5	Dkon	9990080080	

Question: Which of these columns would you prefer to uniquely identify each record in the table?

Let's rule out which ones we can't use -

1. Name?
    a. No, because customers 1 & 2 have the same names.
2. Age?
    a. No, because customers 2 & 4 are of the same age.
    b. And we don't have the age for customer 5.

That means the column **must always be unique** and **can never be NULL**.

What do we call such columns?

Primary Key

- A primary key is a column or a set of columns that uniquely identifies each row in the table.
- Every table in a database should have a primary key because it ensures the uniqueness and integrity of the data.
- The primary key's value must be assigned when inserting a record and can never be updated.
- Note that a relation is allowed to have only one primary key.

Question - In our example of `customer` table, among the 2 columns - customer_id and phone, which one should be the primary key and why?
Answer - The most appropriate column that should be the primary key is customer_id.

Question - But, why not a phone, even if it satisfies all the criteria? i.e., it's both unique and non-null.

Answer - It can be, but what if a customer wants to update his/her phone number?
In that case, it would require updating all related records in other tables that reference it.

Unique Key

A unique key is:

- Unique for all the records of the table.
- Non-updatable, meaning its value can not be changed once assigned.
- But, may have a NULL value.

Foreign Key

Let us look at the `**orders**` table given below.

order_id	item	quantity	cust_id
1	Fruits	2.5	1
2	Veggies	3	1
3	Meat	1	3
4	Veggies	5	5
5	Dairy	2	2

Remember our `**customers**` table?

customer_id	name	phone	age
1	Akon	9876723452	17
2	Akon	9991165674	19
3	Bkon	7898756543	18
4	Ckon	8987867898	19
5	Dkon	9990080080	

Here,

- The customer_id / cust_id column is a common link between the `**orders**` and `**customers**` tables.
- This means that the cust_id column in the `orders` table is used to establish a relationship with the corresponding customer_id in the `customers` table.
- So, if we want to fetch the order details of a customer with customer_id (say 3), we can do that by using the cust_id as a foreign key, referencing the primary key (customer_id) of the `customers` table.

Therefore,

- A foreign key is a column or a set of columns in a table that refers to the primary key of another table.
- It is used to establish a relationship between two tables by enforcing referential integrity.
- The values in the foreign key column(s) must match the values in the primary key column(s) of the referenced table.
- A foreign key may have a name other than that of a primary key.

Candidate Key

Suppose elections are being conducted in your city for the mayor's position.

You have two different candidates out of which you have to choose anyone whom you'll give your vote to.

What you have to notice here is that although you have two available candidates, only one of them can be the city mayor at a time.

The same thing is true with candidate keys.

- Any column or set of columns that can act as a primary key is referred to as a candidate key.
- Every table must have at least a single candidate key (obviously can have multiple candidate keys) which may have single or multiple attributes
- In our example, customer_id and phone both are candidate keys for the `customers` table.
- **Note** that candidate keys are just a theoretical concept. They're not used in real-world applications.

A typical day of a Data Scientist/Analyst working in the industry.

- In any company, data engineers set up a data warehouse. They collect the data from different sources and dump it all into this warehouse in a structured manner.
- Data analysts/scientists / ML engineer needs the historical data from the warehouse to build their model and perform the analytics on top of it.
- A data analyst/scientist would get the data from the warehouse using SQL queries. It is preprocessed and sanitized. Finally using it for their purpose.

What is the difference between a Database and vs Data Warehouse?

- A Database is designed to store and manage day-to-day operational data. It supports routine tasks like inserting, updating, and retrieving data quickly. These tasks are known as **Online Transaction Processing (OLTP**).
- A Data Warehouse serves as a large repository optimized for storing historical data and performing complex queries and analysis. The process of aggregating and analyzing large amounts of data to support decision-making is known as **Online Analytical Processing (OLAP)**
- There are several cloud service providers, such as Amazon Web Services (AWS) by Amazon, Google Cloud Platform (GCP) by Google, and Azure by Microsoft, which offer platforms for various digital services. Many companies prefer these cloud providers for data warehousing due to their scalability and convenience.

For our work, we will use BigQuery on Google Cloud Platform (GCP), a tool designed for managing and analyzing large datasets.

BigQuery is a powerful data warehouse provided by the Google Cloud Platform. It allows you to quickly run SQL-like queries on large datasets using Google's infrastructure without needing to manage servers

Introduction to BigQuery -

- How to create a new project?
- How to create a new dataset?
- How to upload tables in a dataset?

The following document would help you in setting up the **`farmers_market`** database on BigQuery. How to set up BigQuery.pdf

1. **BigQuery setup loom video:** link
2. **BigQuery setup YT video:** link

Overall Summary

In this session, we delved into the fundamentals of databases, focusing on key areas:

1. **Data Management Essentials:**
    a. Introduction to Database Management Systems (DBMS) and their features that overcome the limitations of traditional tools like Excel/ GSheets.
    b. Understanding the structure of databases through schema and Entity-Relationship Diagrams.

2. **Data Types and Keys:**
    a. Exploring various data types such as string, numeric, and date/time.
    b. Introduction to keys in databases, including primary, unique, and foreign keys.

3. **Practical Applications:**
    a. Highlighting real-world applications of databases for data analysts and scientists.
    b. Discussing the significance of Relational Database Management Systems (RDBMS).

4. **Introduction to BigQuery:**
    a. Overview of BigQuery as a powerful data warehouse offered by Google Cloud Platform.
    b. Guidance on setting up a project, dataset, and tables in BigQuery for practical SQL exercises.