**Relational Database Essentials:**

Main goal: To organise huge amount of data that can be quickly retrieved. It’s compact, well-structured and efficient.

**Relational Database VS Spreadsheets:**

**The Advantages of Spread sheets:**

* Spread sheets are low-cost to create. Your company may already have access to Microsoft excel. If not, you can use Google Sheets for free with your Google Account.
* Spread sheets are low-code. Building a spread sheet does not require specific programming languages. You just need to know the language for mathematical functions, which you can easily find online.
* Once created, spread sheets are easy to update and can be shared with different people. This is especially true in Google Sheets, where multiple people can edit a spread sheet simultaneously.

**The Disadvantages of Spread sheets:**

* Large datasets or complex formulas may cause the spread sheet to become slow and unresponsive. Spread sheets are, therefore, best suited for small or medium-sized projects with relatively simple data structures.
* All new spread sheet information must be entered manually. As a result, updating spread sheets can be time-intensive.
* As your business grows, so will your datasets. While spread sheets play a helpful role in your growth journey, they are not sufficient to store data at scale.

**Advantages of Databases:**

* Data can be updated automatically. This can save your team time, as manual updates take hours. Additionally, automation reduces the chance of human error.
* Databases can store information gathered from multiple locations. This can be especially helpful if you’re gathering customer information from more than one platform.
* Databases focus on relational data, which creates the foundation for automation. You can automate much of your data collection and manipulation long-term.
* Databases load faster and take up less storage space. Spread sheets load every cell when the document is in use, taking up more of your computer's processing power. Meanwhile, databases only load information that is actively being accessed. This increases your computer’s efficiency.
* Databases are secure. You can easily identify which of your employees can change the database and which can only view information. Login credentials also protect your information from outsiders.

**Disadvantages of Databases:**

* Databases have higher costs. First, your team needs to purchase a database service. You may also need to pay for additional implementation and maintenance — or hire a new team member to oversee this process.
* Databases require more training to use. You’ll need specialized coding knowledge to set up your database and manipulate the information within. If you're not familiar with what makes up a database (like tables, queries, and indexes), you might find it hard to create and manage one by yourself.
* Databases are resource intensive for small data sets. If you have a small amount of static data, spread sheets provide a more cost-effective option.

**Relational Schemas:**

**Primary Key:**

1. A column (or a set of column) whose values exists and is unique for every record in a table is called primary key.
2. Each table can have one and only one primary key
3. In one table, you cannot have 3 or 4 primary key
4. Primary keys are the unique identifiers of a table
5. Cannot contain null values.
6. Not all tables you’ll work with will have a primary key

**Foreign Key:**

1. Identifies the relationships b/w tables, not the tables themselves
2. Column that references to the primary key of the same table or another table

**Unique Key:**

1. Used whenever you would like to specify that you don’t want to see duplicate data in a given field
2. It can be NULL

**Relationships:**

It tells you how much of the data from a foreign key field can be seen in the primary key column of the table the data is related to and vice versa.

Represent the concept database administrators must implement.

Depict how a database is organised.

Blueprints, or a plan for a database, will help you immensely while writing queries.

1. one-to-many (many-to-one):

One value from the customer\_id column under the “Customers” table can be found many times in the customer\_id column in the “Sales” table

1. one-to-one
2. many-to-many