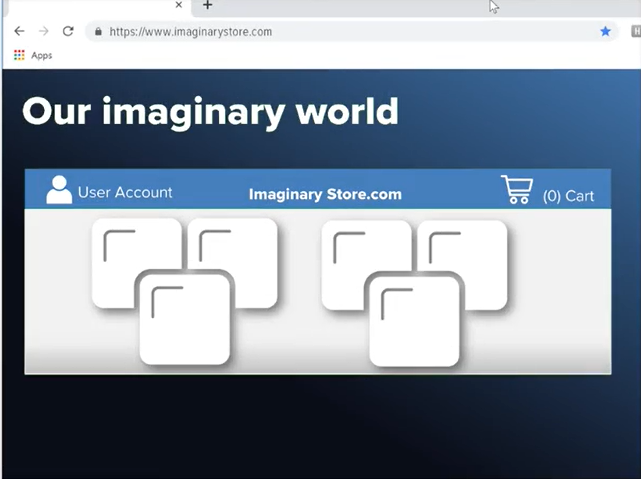
1. learn techniques to

validate and clean the data.

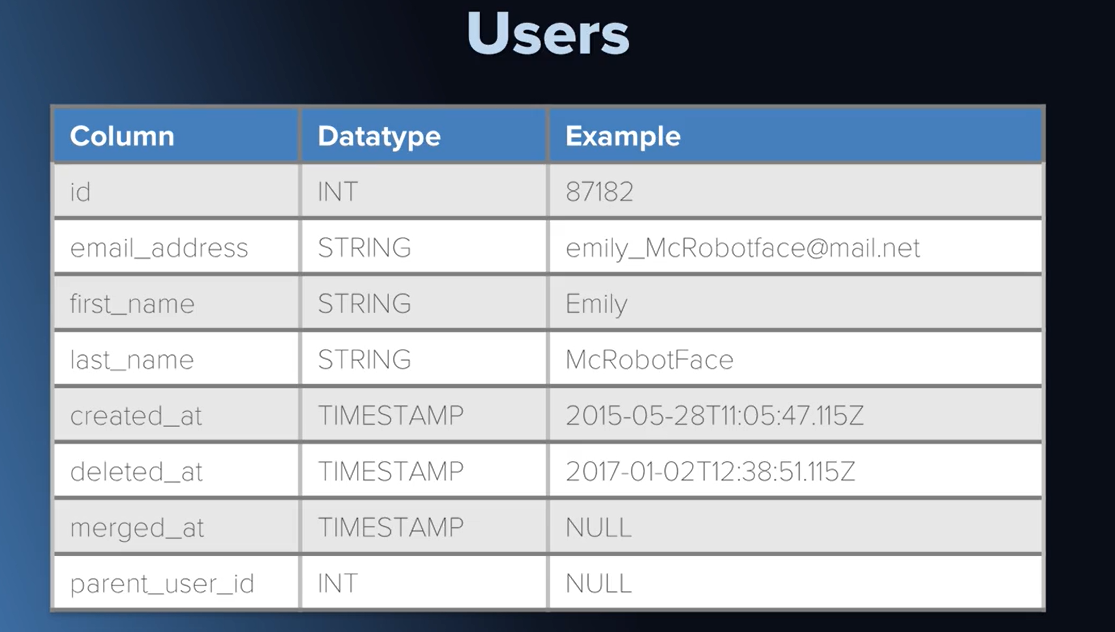
1. you'll discover how the data gets to you?
2. solving problems with SQL.
3. AB testing.
4. Data Quality – Validating and cleaning the data

The project uses a dataset which tries to mimic an online e-commerce site.

Below is a simple wireframe drawing of a website of our [fictitious](https://www.google.com/search?sxsrf=ALeKk00HfY_9PZxSnMnORpeKzTt_YsxRtg:1623159430976&q=fictitious&spell=1&sa=X&ved=2ahUKEwiU1K-KlIjxAhVL8HMBHYkMApkQkeECKAB6BAgBEDE) e-commerce company. We have some items which can be viewed and purchased via the users, in order to purchase an item a visitor must first create an account and be logged in and then add the item to their cart and then check out.

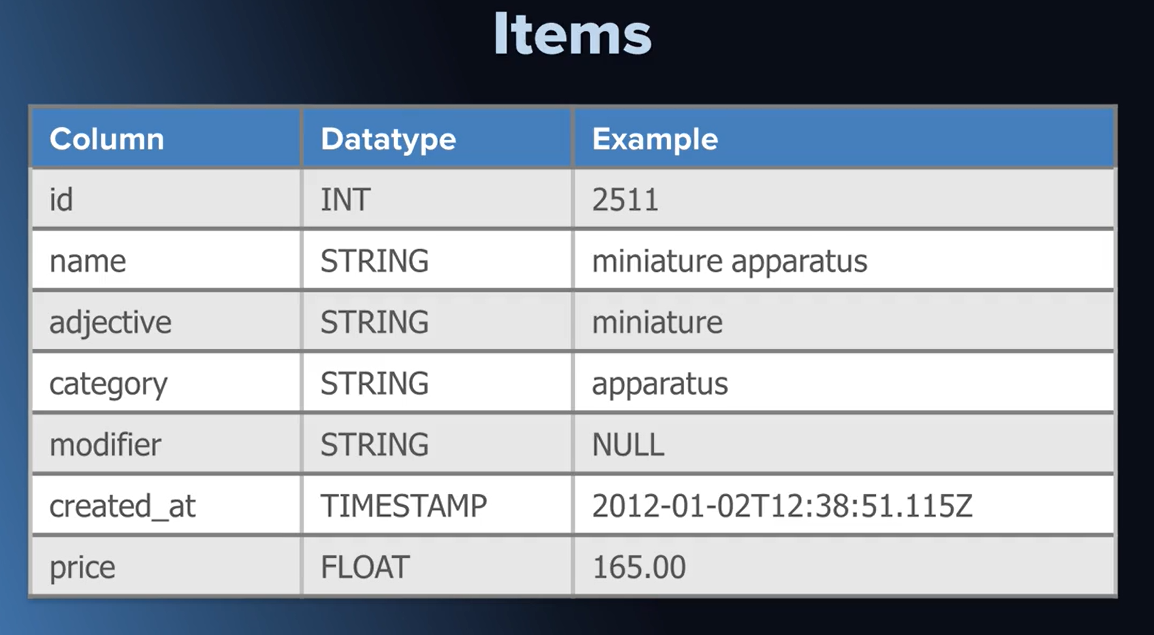


Introduction to the dataset



The first table is the users table, to keep a table of users in our database-

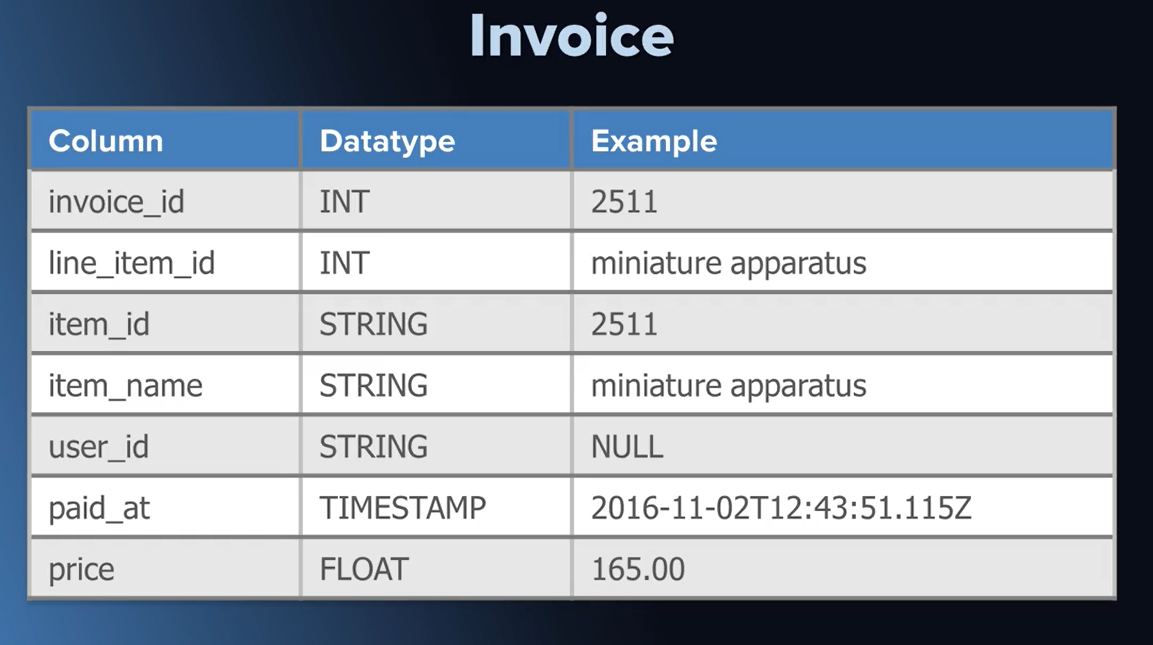
1. ID keeps track of the user ID which is unique to every user
2. Name of the user
3. Email address of the user,
4. When user created the account
5. When/If account was deleted
6. When/If the account has been merged (maybe to group customers with similar purchasing habits)
7. Parent user ID of the user account (ID corresponding to the group in which user has been put in)



To keep track of the items that company is selling on the imaginary website, we have

1. item ID,
2. name for the item,
3. the adjective
4. category
5. modifier
6. when the item was created
7. the current price for the item.

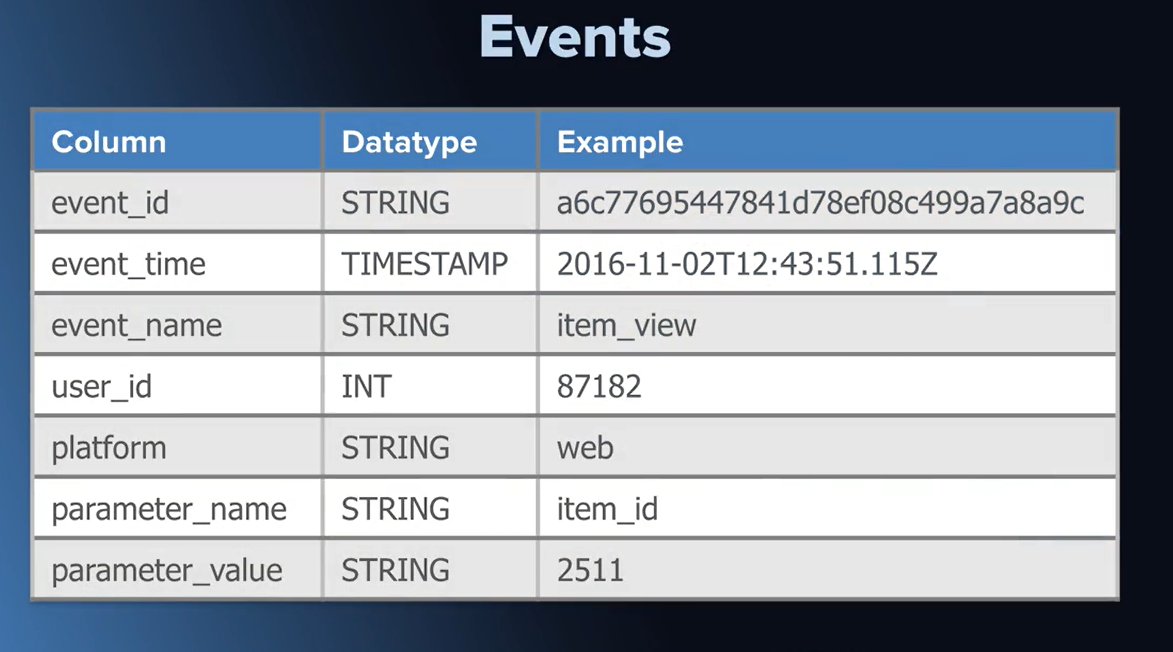
Data Exploration of items table



.

When a user completes the purchase, we would create an invoice indicating

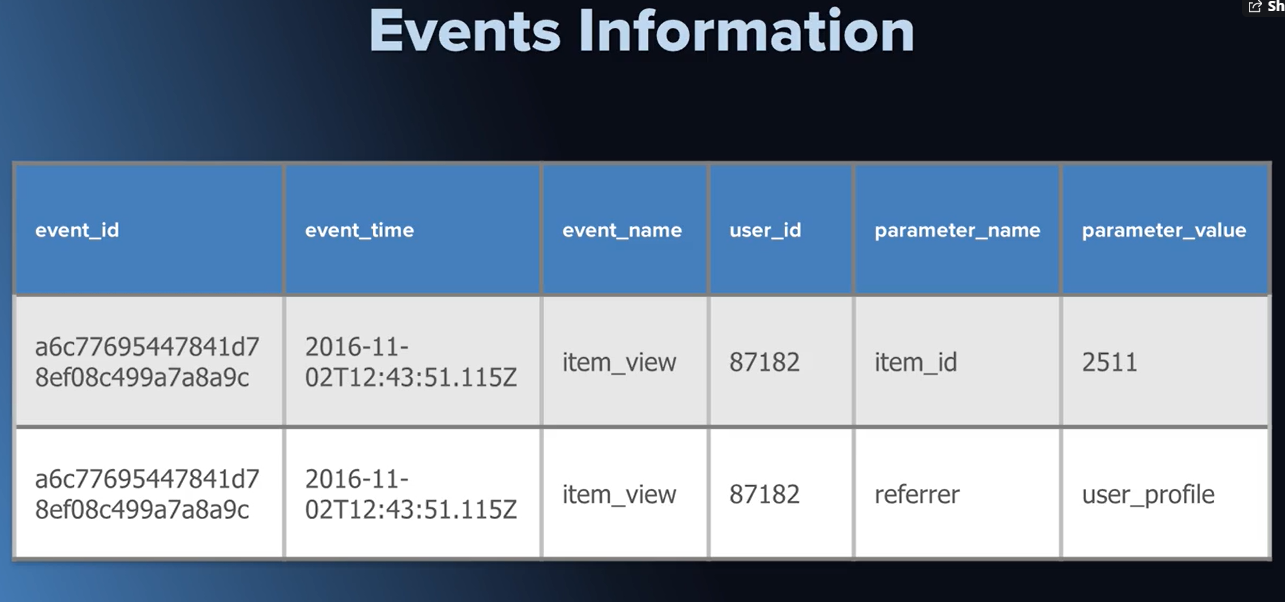
1. The invoice\_id (unique to every purchase made)
2. Line\_item\_id
3. Item\_id (Id corresponding to item purchased)
4. Name of item
5. ID of user who purchased
6. Time of payment
7. Price paid for the item

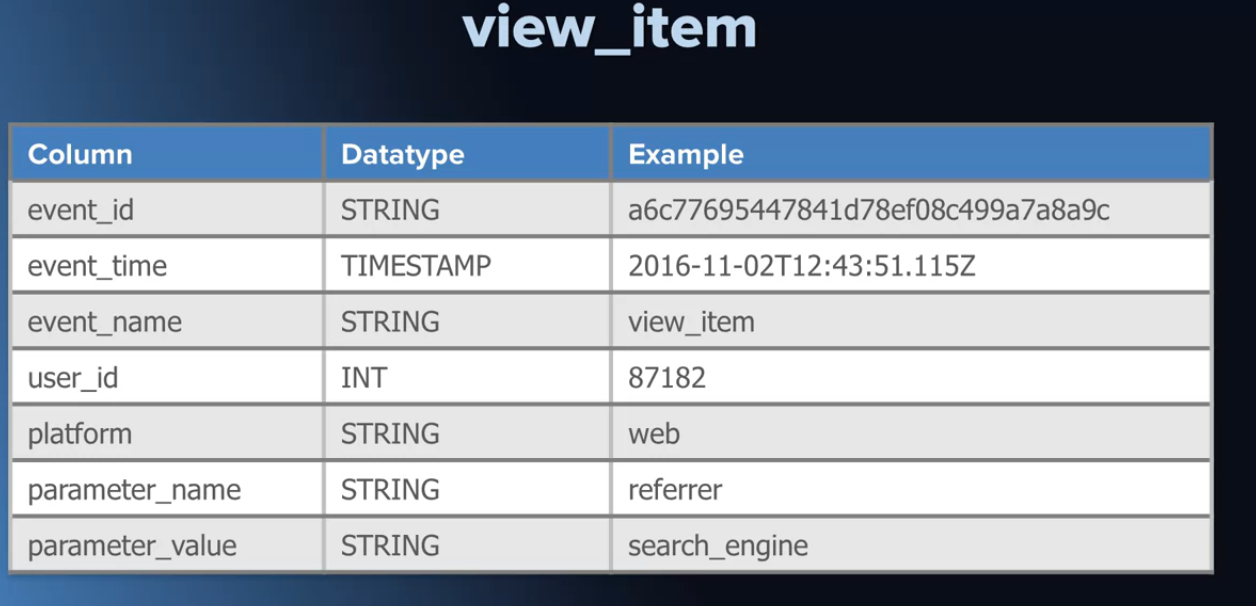


Events table stimulates a copy of a live production table. It is more like a stream of receipts for activity information and we'll be using it to keep track of more ephemeral things like page views.

1. Event\_id shows the unique Id corresponding to the event
2. Event time shows the time at which the event happened
3. Event name indicates the type of event which could be to record a page view or an item view event where a user has gone to specific item page.
4. Platform through which event took place (mobile/computer etc.)
5. Parameter name and value giving further details about the event

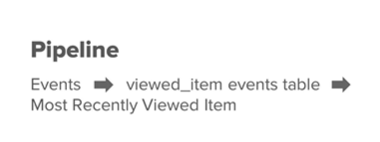
Data Exploration of events table



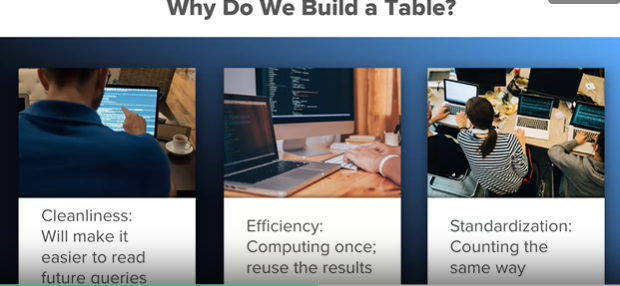


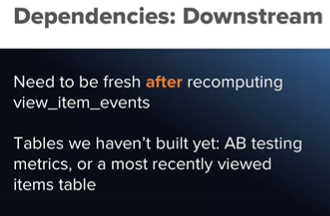
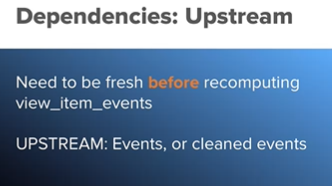
View item table will be giving information about the viewing details of the item like –

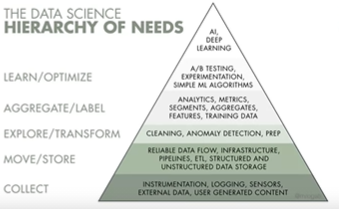
1. Event ID
2. Event time
3. Event name
4. User Id who viewed
5. Platform used for viewing
6. Parameter name and value indicating the page user was coming from before they viewed the item.



This view-item table we want to build would depend on the events table being up to date. The moment new events come in, the view events table will become stale, meaning it will not be up to date. But also if these update happens say nightly then it only makes sense to kick off the job or  the task of updating the view-items table, after the events table has been refreshed and then at last updating the most recently viewed item table in that order. So here's our pipeline. First, the events need to be updated, then the view-item events table, then the most recently viewed-item table. If I build out all these tables and schedule them to run in the proper order, I could say that I've built a data pipeline with multiple dependencies. Sometimes the abbreviation ETL which stands for Extract Transform Load, is used to describe the steps happening under the hood during table creation.  Sometimes the software that manages all of the scheduling is called an ETL System.







What is AB testing?

AB testing is an application of a kind of statistics called hypothesis testing. The idea is that you start with a hypothesis like, this new improvement to a website will cause some metric to increase. Typically, this metric is tied directly to some business metrics like revenue, but it might also make sense to look at metrics that are more explanatory. So, you can prove or disprove a hypothesis about how user behavior changed. By looking at multiple metrics, you can interpret the results holistically to gain insight about the effects of the change.



