Data Quality Patterns

Day 1 Lecture

Transcript:

0:00
[Music]
0:05
good pipelines start with good documentation at Airbnb they have this thing called the Midas processes which
0:11
is how you build a solid gold pipeline that is going to stick around for a long time that people understand that people
0:17
know how to debug that people know how to improve and it really allows pipelines to have a lot longer long-term
0:23
value and in this 2hour course we're going to be going over everything that Airbnb does to build these pipelines I'm
0:29
really excited for you to check out this course and if you want to build more pipelines in the cloud definitely check out the data expert. Academy you can get
0:35
20% off in the description below how many of you have experienced working with bad data I think that this is
0:41
probably one of the things uh that is a very common uh pain point with data
0:49
workers is getting clean data and getting data like able to um be be um
0:57
good and everything so what are we covering today uh one is how to build High trust in the
1:02
data sets that you build because without that trust without that acknowledgement and that trust like are you really being
1:09
data driven if if decisions aren't being made off of the data sets that you're creating are you even having the impact
1:17
that you're expecting to have like who knows that's a I think that's a big thing that you want to think about as you're kind of going through uh this
1:25
process today is really think about trust and how trust can be built and how
1:30

trust can be broken cuz both sides matter here and uh we're going to be covering kind of both sides as well in
1:37
this presentation today also how to build good data documentation the entire lab today is on how to build a spec uh
1:46
and that's also what your all's y'all's homework is this week is going to be how can you build a good data spec um that
1:53
is very important and a lot of times like if you actually go through the
1:58
entire spec process you will find that you create way better data that answers
2:05
way better questions than if you just immediately start writing SQL like you're you're going to you're going to
2:11
have a way better time if you actually do this upfront thing and besides that
2:18
like just from it will increase the quality of your pipeline but then your pipeline is documented then people
2:24
non-technical people or other technical people like other Engineers or anyone else who needs to like look at your
2:30
pipeline and look at the data they can learn about everything that you're creating way quicker than being like Oh
2:37
yeah here's my airflow pipeline go read a bunch of python code and you know scratch your eyes out like freaking
2:44
because a lot of times that's currently what most data Engineers do they just like immediately go into airflow and SQL
2:49
and they don't even think about the spec because there's a lot of pressure to create pipelines and to to give data
2:55
there's that you know there's that Trope of like you have the data scientist who's is like hey can you just quickly
3:01
pull this data for me and then you know data Engineers like jump out the window or they like throw a chair or whatever
3:09
right because that pressure is just obnoxious and the last thing we're going to talk about is data quality checks so

3:15
you know in the first two weeks we talked a lot about facts we talked a lot about Dimensions those are great uh data
3:20
quality checks are also going to be a big part of this we're going to go into a lot more detail on data quality checks
3:27
and automated data quality checks next class on Wednesday uh but today we're going to go into a lot more detail
3:33
especially on like the more theoretical side about data quality checks so yeah
3:39
let's um let's let's let's get into this lecture okay so what does data quality
3:44
even mean that was one of the things I alluded to in the last couple slides um
3:50
so one of the most important things for data quality and it's underrated but
3:56
also difficult is discoverability so what that means is like if someone's
4:02
trying to make a decision and they want to know like oh does this data exist it's easy to get that data and easy to
4:09
find that data if it exists and so like a part of this is like a cataloging
4:14
problem of like showing all the data that your company has which is like
4:20
that's not as much of a data engineering problem as more of like a data infrastructure problem but they are
4:26
related they are definitely related and we we're going to go a lot more deep into discoverability in a in a couple
4:33
slides um you also have misunderstood and incomplete definitions of data quality so like I just want to give a
4:41
quick example so I don't know if y'all like remember Zillow and they uh they
4:46
thought that they had this like perfect machine learning model that was going to like predict the housing market and
4:51
predict everything that was going to happen and then they bought a bunch of real estate and then they lost like a
4:57

hundred million or something like that they lost some Weir weird sum of money because they were like so confident in
5:03
the data that they were creating but the thing is is that they they their data
5:08
wasn't high quality enough because they they didn't think about the gaps of like oh what other things could this uh model
5:15
be missing like what other data points might be very important to also include and sometimes you get uh sometimes even
5:21
if you have complete data you get other things right like if y'all ever heard of like Black Swan events which is just
5:28
like a crazy event that happens that is uh very hard to predict like uh good
5:33
example there is like the the covid-19 pandemic like being able to predict like when that was going to happen that kind
5:39
of Black Swan event is U an example of some things that are like a little bit trickier for machine learning models and
5:45
data to predict sometimes they can get a a decent idea but when they're like rare events like good luck so obviously then
5:53
the third bullet point here is like kind of the standard one of like okay for
5:58
columns that are supposed to not be null they're never null and for uh for Di especially for Dimension tables there's
6:04
no duplicates so that's great um let's talk about the last three here they are
6:12
these are trickier so there is business value being derived from the data this
6:17
is one that can be tricky sometimes because what does that mean what does business value mean um so usually that
6:25
means uh money in some regard uh sometimes it might uh a couple degrees removed from money
6:33
like for example um say you're looking at say you're building a pipeline that measures user growth say like you know
6:40
just we'll just use Facebook since I work there so if someone signs up for Facebook like how much money does

6:46
Facebook get when someone signs up for Facebook they get nothing right and so
6:51
like actually tracking how many people sign up they don't get money directly that's like a um growth like signups is
6:59
um a leading indicator of money because it's like the only way that they actually make money is if people sign up
7:06
and then they stay and they're they stay in the feed and then they start consuming ads because if they start
7:12
consuming ads that's when you're going to get a lot more money right and so and some things can be even further removed
7:19
right so think about it this way like if uh for Facebook say they went and spent
7:24
money on Google and they were like they spent money on Google ads to get get more people to go to Facebook because
7:31
then it's like okay just because they clicked on the Google ad doesn't mean they're going to sign up and just
7:36
because they signed up doesn't mean that they're they're going to watch stuff in the feed so that like the number of
7:43
clicks from Google is even further removed from actually making money but
7:48
my whole point here is generally speaking you can like every single data
7:54
pipeline that you write is either going to generate more money there's going to
7:59
be some way that it generates more money or like has like an obvious way that it is connected to revenue or it could be
8:06
on the other side it could be cost savings where like you might be making a pipeline that measures how much spend is
8:13
happening on AWS so that you quit giving Jeff Bezos billions of dollars that's a
8:18
that's a one of our goals as data Engineers is to make Jeff Bezos less Rich so um that's the idea business
8:25

value usually is revenue there is one exception here I want to talk about with business value is if you're creating
8:32
data sets that are they that provide strategic value so say you're creating
8:38
some big data set that is used for some big executive decision and then it's just to inform that decision and it that
8:46
might not be money in like a clear direct path like the other uh data
8:53
pipelines that you write are but those pipelines are also kind of rare I think I've written like like one or two of
9:00
those in my whole career so generally speaking the pipelines that you write are very closely linked to either
9:06
revenue or cost okay well what about these last two uh data is easy to use so this is an
9:13
interesting one because I I spent the last two weeks teaching y'all how to make data a little bit harder to use uh
9:20
by using arrays and structs and complex data types and showing analysts like hey
9:25
you should you know use cross joint unest or you know kind of blow up the data or we can like analyze it with
9:31
these array functions and like you're going to have to be a little bit more technical to look at this data right and
9:36
so obviously this is something that is a Continuum and it depends on your data
9:41
consumer on like whether the data is easy to use or not but generally speaking what I mean by data is easy
9:48
easy to use is the column names are obvious the column names make sense the
9:53
column names like are like clearly distinguished like a very common thing
9:58
that happens uh at Airbnb is you have Dimension columns and Metric columns and
10:04
the dimension columns start with dim and the metric columns start with M underscore and that is a very common uh
10:12

naming convention I would hope that naming convention gets more adopted like across industry but some people are
10:17
different they like they like to use other things and some people are like oh that's like excessive just put dim in
10:22
the table name and then you're good to go like there's a lot of debate on like the way to actually name your columns
10:28
we're going to cover that in another slide and uh the last thing here is the data arrives in a timely manner like a
10:34
big common thing between analytics and data engineering is when there's a data delay
10:42
analytics is like data engineering like where's the data what's going on and
10:49
so especially uh this back and forth happens a lot when there isn't an agreed
10:55
upon uh refresh interval so for example when I worked at Airbnb I worked on unit
11:02
economics and when I first was working there the unit economics pipeline was like so bogged down and slow and um like
11:09
the the refresh interval was three days and so like the only time what it meant
11:15
was like if analytics uh came to me and they're like hey the data is two days late my for me the the response for that
11:24
is do nothing right just wait and then uh if it's more than 3 days late that's
11:29
when I'll actually troubleshoot so coming up with agreed upon arrival times for data is a very
11:38
powerful thing that can help your analytics teams kind of be more streamlined and it it makes it so data
11:44
engineers get less requests as well because then the analytics team can be like oh it's delayed but it's not
11:49
delayed enough that I need to like ask someone about it it's like not quite at a pro it's not problematic yet so those
11:57
are a lot of the pieces of like what data quality means uh there like obviously there's um there's more to it
12:04

than that but that's kind of the idea so data quality like in a in in a very
12:10
short way to think about it is Data Trust Plus data impact so it's you have
12:17
people that believe in the data and believe that it's correct and then on the other side you have the the data is
12:25
correct and changing the business and if you have those two properties of your
12:32
data sets you're going to have a good time that's like if you can consistently
12:37
create data sets that have those two properties you're going to get promoted very quickly that's like a it's it's
12:43
more difficult to do than you would think uh and that's one of the reasons why we're here in this lab today is to
12:48
talk about how to do this because this is not an easy process to uh
12:54
consistently deliver high quality data every time so yeah let's let's let's dig a little bit deeper into
13:02
this how do you build High trust in data sets so this is coming from the angle
13:10
of before it's in production before you've created any pipeline you really
13:16
want to uh lean into empathy is a big thing I
13:23
think it's one of those things that data Engineers like generally speaking don't do enough of
13:29
uh in this case what I mean by that is you need to get all of your Downstream stakeholders together and be like I like
13:37
to do I love I love these conversations they're like one of my favorite conversations is like you get your Downstream stles together together and
13:43
you like if if uh cost and time wasn't an
13:50
objection what data would you want and like really have them like shoot for the
13:56
sky sky for two reasons one is it gives you a lot more clarity on like all the
14:02
problems that they're trying to solve and being able to get more big bigger picture kind of things and it can also
14:08

give you ideas of like what to do in a year or what to do in six months and as
14:13
you progress in your career and data you're going to be wanting to think about those things especially like once
14:18
if you want to go from senior to staff data engineer your goal is to determine
14:24
what to work on next that is like one of the most important things so this exercise and empathy can be very
14:32
important in determining not just what to work on today but to what to work on in 3 months in six months in a year and
14:38
like really starting to build those like long-term road maps of like what does data Excellence look
14:45
like so yeah those conversations are great um off also a spec and this spec
14:51
should be reviewed uh the spec should be reviewed by a couple people uh one is the spec should be reviewed by another
14:58
data uh expert like a like at Airbnb how it worked was um every spec that was
15:05
created had to be reviewed by at least another staff data engineer so that was
15:12
a requirement that and so if I wrote a spec I couldn't review my own spec I'd have to have another staff engineer
15:18
review my spec and so then that was just like a hard requirement but you also needed the spec to be reviewed by the
15:24
stakeholders so both uh the like like a technical review by a an experienced
15:31
engineer and um a review by the downstream stakeholders to make sure that you're covering most of the
15:37
requirements so obviously that's great another big important piece of this is to clarify the business impact because I
15:46
think that that's one of the things that people can kind of get caught up with a little bit is like how do you have the
15:52
right amount of business uh impact and the right way of kind of describing that
15:58

stuff and one of the things that's really good is if you can clarify the business impact that's also something
16:05
that is great for performance reviews because sometimes like depending on
16:10
where you're at in the development cycle um you're not GNA like say say you're
16:16
doing a spec review and it's December and you're just starting the spec review
16:21
then you're not going to get credit for if you if if you think that you only get
16:26
credit when you finish a pipeline that's not true if you can build a spec that looks very
16:31
promising that has a lot of really good business impact and you can convey all of that within the year and say it's in
16:38
like November de December then that can count towards that Year's performance review and that can be a great way to
16:44
also kind of squeeze in one last thing to get uh more value into your promotion
16:50
packet so that you get most of the credit for that pipeline for the previous year not the the coming year
16:56
and that can be another great way to help you get promoted so like that's a great thing to think about is and why
17:03
clarifying the business impact is powerful because like you essentially can take credit for the impact before
17:10
it's delivered and that is like I I don't know like i' I found that to be so
17:16
tasty and so satisfying in my career that I've just been like thank you I'm I'm I'm glad I I found this and saw this
17:22
and because then it's like then you just have to implement it and then you still get credit on the other side as well so
17:28
um um also another big thing here is like talking to your Downstream stakehold holders about current and
17:35
future needs so a lot of times uh stakeholders are going to have like a burning question like they're like I
17:42

need to know availability of ariran bees in Brazil in July 2023 because there's a
17:49
big event going on and right it's like they have a very specific narrow question that they're
17:55
trying to answer right now and don't build your pipelines to answer that
18:00
narrow question if you do you're going to have a bad time um because then they're going to come back to you and ask another narrow question and you're
18:06
going to build another pipeline that's almost the same Pipeline and then you're
18:11
like uh that's called uh like um me and my manager back at Netflix we called that like bespoke hell because like you
18:19
just keep essentially you you build a pipeline for every question that analytics has which is like not the way
18:27
to go because you want to thinking about what their current and future needs are so that you can build a data model that
18:33
answers their questions today and the questions they have in six months so that then you don't have to make another
18:38
pipeline you can just be like yeah the data's already there and when that happens when your analytics Partners
18:44
come back to you with another narrow question and then you can just be like the data is already there they will be like wow like that is one of those
18:51
things that really gets you a lot of Kudos because they're like wow like I don't have to pull the data wow I don't
18:56
have to transform the data I just have to do another query and I'm good to go so that's why we spent you know a lot of
19:04
hours a lot of hours together working on data modeling right as you you listen to me talk for 12 hours because getting
19:11
data getting the data model right is very important for building trust and really wowing your
19:16
stakeholders and uh this last bullet here is called follow the Airbnb Midas
19:21
process I will send you guys links for this as well uh I'm just GNA we're going to go over the Airbnb Midas process in

19:28
this um presentation today um also know that like my perspective on the Airbnb
19:33
Midas process is that it's like most companies they lean too little into
19:40
quality and I feel like the Airbnb Midas process almost goes too far the other way where it's almost like too onerous
19:47
but we're going to go over it and um and even if you don't follow all of like
19:53
it's I think it's like an eight-step process it's a lot like and so even if you don't follow all eight steps if you
20:00
follow a couple of them you will see a dramatic Improvement in your data quality pipelines so or in the data
20:05
quality of your pipelines so don't think that I'm like saying you have to do everything that Airbnb does because like
20:11
most likely like you're at a company that might not have that many data engineers and so they like it's not as
20:16
possible so yeah let's let's let's dive into it oh yeah nine W there's nine steps so
20:24
let's uh I just want to go over each of these steps on like how Airbnb builds a very high quality Pipeline and what it
20:30
does with Midas pipelines so Midas uh I don't know if y'all know he's like that's like a legend for like a golden
20:36
touch kind of person and uh every King Midas whatever he touched turned to gold
20:41
and so that's kind of the idea here the Midas process is how to create golden data sets that are very valuable and
20:47
they keep giving value so the very first step of the Midas process at Airbnb is
20:52
to make a spec can you make a spec uh which is like this is the pipeline I'm
20:58
going to build build and we're going to this is how we're going to build it and then uh step two right is that technical
21:04
review I was talking about and the stakeholder review so spec review is you have to get a technical review from like

21:10
a data architect and then also a stakeholder review to make sure it covers their needs then you uh do a build and a and
21:18
backfill of your pipeline so you actually go and write all the spark code and the airfow code and then you
21:24
actually create the backfill code for your pipeline keeping in mind here uh
21:29
when you backfill you're not backfilling like all of history at first you backfill like one month so then uh the
21:36
analyst who you're working with will then start working on step four which is the SQL validation which is where they
21:42
essentially go through all the data that you created like that one month of data that you created and just like dig super
21:47
deep and try to make sure that there's not any weirdness or Oddities in the data and it's crazy cuz I built like
21:54
eight or nine of these pipelines when I was at Airbnb and like every time I like finished my backfill pipeline I'm like
22:01
this is Rock Solid I will never have to do it again like I I know I got it perfect the first time and like I like
22:08
every single time like without fail there was always bugs and that's why you only backfill a month because if you
22:13
backfill more than a month then it's like and then they find a bug and you have to back fill it all again like that's just painful so then you do SQL
22:20
validation and keep in mind SQL validation needs to happen by someone who isn't you right that needs to be
22:25
usually an analytics partner an analyst or like an analytics engineer someone who isn't you because if you
22:30
build the pipeline your BL your ability to check the quality of the pipeline is impaired because of the fact that you
22:37
built it it's kind of like how you know people are very unwilling to admit that their children are ugly even though we
22:43

know for sure that some people have ugly children like and you can just but like if it's not your kid like it's a lot easier to call the baby ugly so it's

22:50

it's a it's a solid way to kind of go about things right so then after the squl validation then you have the manura

22:57

validation manura validation is where you're looking at the metrics so manura is an open-source metric repository

23:04

container thing that Airbnb uses to hold all their business metrics and they validate all the uh new manura metrics

23:12

and this is done by like again like the same person usually the same person who does the validation they do the manura

23:17

review or the manura validation then you have six which is where the data architect comes back so I would do these

23:24

reviews as well where after they did all of this stuff I would then go over their code and their data so in this case for

23:32

the data review I would just make sure that there's not any weird like modeling problems because a lot of times in the

23:38

spec review you're not going to catch everything as an architect and so I just review in the data review a final thing

23:44

of like this column's weird or like why is this missing or whatever and so

23:49

that's the data review and then the code review is where I go through the spark code and a lot of times with spark code

23:55

the big thing is looking to see if they have tests so they need to write unit tests and integration tests for their

24:01

spark code and this is something if you're in the combined track uh which I think about half of you are in the

24:07

combined track we're going to literally be going over how to do that tomorrow on how to do uh unit tests and integration

24:13

tests on your spark code and like end to end tests on your spark code that will be a nice fun little project to work on

24:20

um tomorrow for you in the combin track then after that you migrate the metrics that's what manua migration is then you
24:27
have one one more review by like a this is by like a data scientist like a staff data scientist and he's like okay these
24:33
new metrics look good and then you finally say we created the data set and that's when you launch the PSA and
24:39
that's the ninth step so um I'm sure some of y'all are like
24:45
damn that's like a lot of steps like Cu uh a lot of times like like most
24:51
companies like when you're working I know like especially like when I worked at Facebook how it worked was there was
24:57
step three and step four and like that was it for the most part why all this un
25:03
upfront work though like why are we doing this why are we writing all this documentation like why can't we just write our data and give them a data and
25:09
then they can go on their merry way uh a couple reasons and some of them are actually uh tricky for some Engineers to
25:17
understand sometimes so if you get buyin from a stakeholder before you build
25:25
something they're going to feel like they built it too because they gave you ideas right like it's kind of like how
25:33
if you are like yo like I want to get your opinion about this art piece but
25:38
it's before you build the art piece and you're like what should I put in this art piece and then someone's like you should put a dragon in that art piece
25:45
and then you put a dragon in the art piece and then you show them and then they're like that's my dragon that that that dragon's freaking sick right and
25:51
then they like they love it more right they because they feel like it's theirs like they were a part of the process
25:57
right that's a thing that analytics people really like because and it also
26:02
it's also good too because if you can make them feel like they're a part of the process and then uh it's missing

26:09
stuff like they're going to blame you less and blame themselves more which uh
26:14
is good for you I think I mean it depends like I the blame game is also kind of toxic but uh generally speaking
26:20
that can take a little bit of pressure off of you because you're like yo like you were I I I I looped you in for every
26:26
step of this process like why do we have to backfill again and what it can do as
26:31
well is it can highlight to management like where communication problems are
26:36
happening because sometimes as a data engineer it's not even on you sometimes like analytics partners are just absolutely ridiculous in what they're
26:43
requesting and like how they're requesting it and so that's where like if you can Loop them in throughout the
26:48
whole process and really it's kind of like going on a bunch of dates with your analytics partners because like and you
26:54
know if you just go on one date with them and you're like okay we're building this Pipeline and then you don't even know if you're going to communicate with
27:00
them well and then like you it feels like after you like buy it's like and then it's like when you give them the pipeline it feels like it's like buying
27:07
your second date a car it's like and like you know I don't know that a lot of people would say that that's probably a
27:12
little bit excessive to buy to to give a girl a car on the second date that's probably like a little bit a little bit
27:18
too much right and so that's where uh you can kind of think about how building
27:23
up these connections having these conversations it what it does is it also surfaces like hey we're having
27:28
communication problems hey like this isn't working out well right and you can catch these communication problems
27:34
before people are so damn invested where it's like you you won't be like oh we
27:39

had these communication problems but then I still wrote 1,200 lines of SQL and then I just freaking ate all the
27:45
pain and that is like I've seen that happen I've seen that happen many times in my career as a data engineer and
27:51
that's stuff stuff like that is actually how a lot of data engineers get like meets most on their performance reviews
27:57
because they didn't communicate well enough and then they didn't clarify things and then they just jumped right in the coding and
28:04
then it made it harder for them to undo things because they already had a lot of it built out and so that is be it's a
28:11
beautiful thing when this works it feels very beautiful and it feels like wow we
28:16
are like in this process of solving all these beautiful business problems and like it can feel fun too I I that's the
28:24
part that I've found very interesting about this is like especially for me when when I worked in pricing and availability at Airbnb and I was doing
28:31
these like stakeholder sessions of just like oh what what what do people want like how do we want to change things how
28:37
do we want to implement things and like actually really talking to stakeholders like good stuff that's that good
28:44
dude mhm okay and so anyways obviously I said they feel like they have the skin in the game they're
28:50
they're a partial owner it's like they have stock in your pipeline so they they want they want you to win as well as
28:56
opposed to like building something in a corner and then obviously there's a thing in all caps here like you want to do this upfront work because back
29:03
filling is painful like I feel like back filling is like the b word for freaking data engineers and so like don't uh like
29:11
don't like worry about that too much like if you are doing all this up front work a lot of times even if you do all this up front work you might still have
29:17

to back F twice I mean that's generally how it works for me but like if you don't do this upfront work you might end
29:22
up back filling like multiple times like three or four times or something like that and like that is getting a little
29:28
bit excessive that's also giving Jeff Bezos a lot of money that's also just like because data ioe is one of the most
29:33
expensive costs when generating data and so like if you're back feeling a lot of history like you're going to have a
29:39
you're going to have a hard time so uh yeah let's let's let's go to the next slide here okay so obviously like not
29:46
every single pipeline is worthy of this nine-step process right that's that should be
29:52
clear so there's going to be two big things here I think one of them is okay
29:57
are there any important business decisions made by this Pipeline and like
30:02
important is obviously uh like uh relative but generally speaking if it's
30:09
not and it's just more of like maybe exploratory or maybe there might be a trend here or like because sometimes you
30:15
do have that space to like kind of look at things in a more exploratory way in analytics and those pipelines a lot of
30:22
those like exploratory like prototype pipelines they do they have value but they don't have in terms of like they
30:29
need to stick around for years and years and years and give people value every day and so those kind of pipelines you
30:35
probably don't need to do this nine-step process and that would probably be a massive and ridiculous waste of time but
30:41
uh another big thing is is like if the data is going to change a lot like if you are going through a lot of uh churn
30:47
or process stuff like then it might not be worth it to do it now but like wait
30:52

until that churn has like kind of happened I know that that was a big thing that was happening for me like in
30:58
economics like payments was going through this big like refactor and I was like y'all need to like give me a Midas
31:03
Pipeline and uh they were like no we're not going to give you a pipeline because of the fact that like if we did we'd
31:09
have to just build it and then deprecate it in three months because everything's going to be different soon so like
31:15
that's another thing to think about is like some of this stuff like the like this heavy process is really for
31:22
multi-year master data that is going to keep giving value over and over and over
31:27
again and it won't be subject to very disruptive change or if it is subject to
31:33
disruptive change it happens on like a couple year basis right it's not going to happen soon it might happen in a year
31:39
or two so that's a big thing to think about when you're going through these processes because just because you're doing a heavy quality process doesn't
31:46
mean you're delivering value and then sometimes not doing the heavy process will deliver more value that's like a
31:53
big thing to remember here is that there is kind of a a gradient here like it it go is like like you can go as far into
32:00
quality as you want or as little into quality as you want and like they they have they're pros and cons right it's
32:05
not like always do the process or never do the process uh there's a couple big things that make a good spec uh one is
32:12
the description of the pipeline like why are we building this like what is the purpose of this pipeline you also have
32:18
things like flow diagrams uh like how things are uh how things kind of position and like how data goes from
32:25
like the raw data to the fact Dimension data to the metrics then you have schema

32:31
schema is going to be like your ddl statements and your create table statements and then column comments to
32:37
maybe talk about any Oddities in any column comments and you also want the quality checks and like because you want
32:44
the quality guarantees in the pipeline because that stuff's that can be very good to add as well um metric
32:49
definitions also important like uh what metrics are you trying to even measure and uh the last one is example
32:56
queries which is like especially if you have more of that like struct and array
33:02
format stuff from week one and two having some example queries to show people can be very powerful because then
33:09
they might come and ask you questions of like how do I query this and then you have a good you already have a good example for them so that's kind of the
33:17
idea behind what makes a good spec um okay let's just kind of go over this
33:23
like what's in a good flow diagram so here's a good examp example of like one
33:30
that might be uh in a Midas spec when you're creating your data so you see how
33:36
we have our input data sets here we have dim countries dim users and raw activity
33:41
and so these three come together and then they create this fact table called fact activity and that is going to be
33:49
our um this is like the master data I generally speak generally speaking I
33:54
would say that this orange table here would be What's called the master data that is going to be what most people
33:59
query and then you have uh you can also create like an aggregate table which you can think of is just grouping this table
34:05
and moving it down and then you can aggregate the aggregate table and create the the fine grain metrics and this will
34:12
give you like some ratio metrics these are per user obviously you can have

34:18
other metrics that are like totals and stuff like that as well but um generally
34:23
speaking like these uh ah these uh diagrams I like to use Lucid chart for
34:29
this that's what I use this for um you can get like a free trial for Lucid chart and you can build this for free uh
34:37
Google drawing is okay but I don't know I feel like every time I use Google drawing I always feel like like I'm like
34:42
a kindergartener who's like writing my freaking flow diagrams with crayons and like I don't know it doesn't feel good
34:49
to me doesn't feel good to me so I'm more about I'm more of a lucid chart kind of guy uh but that's you know at
34:55
Lucid chart you should sponsor me uh okay let's let's go to the next slide um we talked about
35:02
schema uh so your table Nam should look good generally speaking your table name
35:08
should have either fact dim SCD or AG in the name to illustrate what type of
35:16
table it is so it's either a fact table a daily Dimension table an SCD Dimension
35:22
table or an aggregate table and like like pretty much every single table at
35:28
Airbnb where you can had one of these in the name like that was kind of like a requirement and it made it so it was
35:35
like a lot more obvious like what kind of table this is going to be and then if you have people who just have even a
35:40
basic understanding of data modeling they can understand like okay this table's for events this table is for
35:46
Dimensions right you can get all sorts you like just from that in the name you get a lot of
35:51
information so obviously for your schemas you also are going to need a lot of comments for your columns like make
35:58
sure that every column has a comment uh and these comments should follow the the
36:04
naming convention that your company chooses um and if that doesn't exist if

36:10
you're at a company that doesn't have naming conventions then that's a great place to freaking get a lot of impact
36:16
and that's something that you should be yelling at people about where it's like why do we not have consistent naming conventions why are we doing things this
36:23
way like obviously if everyone's going to do their thing their own way like people are going to have a harder time
36:29
understanding this data and so like you just got to be very clear with people and this can be a massive way to get a
36:35
lot of impact like if and there's a lot of companies that aren't doing this because I mean a lot of companies aren't big Tech and they're not like so far
36:41
advanced in their data practices that they even have this yet so that can be another great place to go but the key
36:47
thing here is follow the naming conventions at your company or build them if they don't exist comment every
36:52
column every table name should have fact dim SD or a in the name those four if
36:57
you do those four for your schemas you're going to be very far ahead of a lot of different people so that's a good
37:04
way to build good tables that have good names and good columns quality checks we're almost here
37:11
and then we're going to take a break here in like 5 10 minutes been talking a lot so quality checks come in three
37:17
flavors you have the basic checks basic checks are really easy um basic checks
37:23
generally speaking uh like every pipeline should have these basic checks
37:29
all the time because if they didn't have these basic checks like what like what
37:35
are we doing they're so cheap they're like so they're like free essentially so like and they also help a lot oh I
37:41
forgot to add one there's one more basic check which is like uh is there data like there's actually data right there's
37:47

the the the not empty check then you have not null for any column that's not supposed to be null like it's never null
37:54
no duplicates make sure that's the case and then right we tal we talked a lot about enums and make sure all the enums
38:00
are a valid value because in the data Lake environment like it's different from postgress postgress is going to
38:06
enforce it for you and you don't have to worry about enum values but if you're in the data Lake environment like in spark
38:12
or in Presto like it's a lot more flexible so you need to test these values yourself in your data quality
38:18
checks then you have like intermediate checks most of this is like row over or or week over week row counts and
38:25
everything like that so um um in this case like when you look at like week over week row counts you uh you want to
38:33
do week over week and not day over day because of the fact that like Saturday
38:38
versus Friday is going to be weird because a lot of times companies and things have different behaviors on
38:43
Saturday than they do on Friday but if you compare compare Saturday to last Saturday uh that's going to be a better
38:50
consistent Trend there still could be other seasonality things that happen on like a longer term Cadence like for
38:56
example holidays right where like Christmas these intermediate checks like they fail on Christmas all the time like
39:02
it's it's it's annoying how often they fail on Christmas because of like the seasonality but companies are getting
39:07
more mature and they're actually getting more like machine learning and stuff like that involved as well so that then
39:14
you can do seasonality adjusted row counts so that then you can look at both week over week but you can also look at
39:20

it in context with like last year or the year before so that you can get like a longer term seasonality context so that
39:27
like your data quality checks don't false positive more than they should and that's how you get into those Advanced
39:33
checks where you like bring in the heavy guns you bring in the machine learning you bring in all that stuff and that can
39:39
oftentimes make your quality checks higher signal and less noise and that can be very powerful but like bringing
39:46
these in can also be kind of expensive depending on the company that you're working for so uh as your data
39:52
infrastructure improves generally speaking these Advanced checks become uh very easy to add I I noticed that like
40:00
like at Airbnb it was like seasonality adjusted um uh row count was like it was
40:07
like a yaml configuration one line of code done so how do you pick quality checks um So Def Dimension and fact
40:14
tables generally require different quality checks uh we'll go over that real quick here so Dimension tables
40:21
generally speaking they are they either grow every day or they're flat
40:26
especially if you're doing like cumulative table design then like they should be growing or flat every day so a
40:32
very common check for Dimension tables is table is growing which is like is there more rows today than yesterday and
40:39
like it should almost always be like an up and to the right Trend that's like a but like at the same time you don't want
40:44
them to grow too sharply which is similar to that threads example that Stephanie just brought up is like
40:50
because it like if you have like week over week say there was like 100% more users and it's like Facebook's like we
40:57

didn't get two billion signups this week like no way right and so uh there's going to be a percent that is uh
41:04
reasonable for your users depending on like uh the the volatility of your data
41:10
set and then um then you also like you want to make sure your complex relationships are also checked so in
41:18
that case like for example on at on Facebook there is like say you're working in the friending environment uh
41:26
one of the hard limits friending is you can only have 5,000 friends on Facebook and so like you want to check that make
41:31
sure that there's no more than 5,000 friends like make sure that that's like uh enforced in the data right and uh
41:37
other things too like uh say I you know I have Progressive car insurance and it's like if I have an active policy
41:43
there should be a car in my policy as well not just me right um because
41:49
Progressive doesn't do like health insurance right they only do like car insurance so you could think of like all
41:54
the different like uh connections of your entities and sometimes they have to exist and like if they have to exist
42:01
that should be a check in your data in your data pipeline so that's the dimensional data quality checks
42:06
Dimension data quality checks are usually simpler and they don't false positive as much um then you have uh
42:14
fact quality checks so facts usually have like the seasonality problem that's
42:19
where like if you have row counts you don't want to do day over day because the whole Saturday Friday problem always
42:26
do week over week again this can have a problem where uh it's affected by
42:32
holidays um facts can grow and Shrink based on user Behavior so in that case uh the table is
42:40
growing check from uh the previous slide is not a good check uh facts are more prone to

42:46
duplicates because duplicates are a lot more H more likely to happen in logs than they are in production databases so
42:54
the duplicate check is very important in uh fact data um more prone to NES and
43:00
other uh row level quality issues obviously that stuff like because it's logs and logs a lot of the time aren't
43:07
critical for the app to run and so that can be where uh software Engineers can get more sloppy when they're working on
43:14
logs than when they're working on like production database snapshots so that's also true and then obviously they can
43:20
have a foreign key uh it's like a for you can have references like it's like I
43:25
I received this notific from Facebook you got to make sure that that user ID is real and that like that like and and
43:32
this is cool as well you can also do other enforcement like what if you have deactivated users right so it's like we
43:37
want to make sure that deactivate and there's no deactivated users on Facebook who are receiving notifications because
43:43
that is also like a bug it's like it's a different bug right it's not really a data quality bug but it can catch
43:49
infrastructure bugs that are happening uh like in other places in the app so you can also catch things like that
43:56
where these data quality things actually um caught um some like cyber security problems at Facebook a couple times and
44:02
they were the first place to alert that like hey there's an issue here something some people are getting hacked something
44:08
is happening here and like and so know that like these data quality things aren't just to build trust in you but
44:15
they can also help the business in kind of ways that you wouldn't expect especially if you do a good job
44:21
here okay last thing I want to talk about is like in the dupes when I was talking about dupes in um

44:27
uh for fact data don't use count distinct like when you in Presto just don't use it like it's terrible like
44:33
it's so terrible and like a lot of times especially if you have high volume fact data Presto will just die because count
44:38
distinct in Presto is terrible if you use a count a prox count distinct that's
44:44
going to be better and that's going to um capture 99.9% of the data there could
44:49
still be a chance that there's like one duplicate like in some like for far away land but this is going to capture 99.9%
44:55
of the time and it will make your pipeline run faster so if you're doing your checks in Presto use a proc count
45:02
distinct and not count distinct it's just like a random little tip congrats on getting to the end of the SPEC
45:08
Building lecture there's a lot of steps in the Midas process right if you're taking this class for credit make sure to switch over to the other tab so that