Fact Data Modeling

Fact Data Modeling Day 3 Lab

How Meta models Big Volume Event Data

Building reduced facts

Transcript:

so we can say create table uh we're going to call this

3:21:08
uh let's call this uh let's call it uh let's call it array
3:21:14
metrics I think that's a good name um then we have a user ID
3:21:21
right and apparently want to call it numeric cuz numeric will work because I
3:21:29
remember in yesterday's class it was like weird because it let just match it with whatever's in events here even
3:21:34
though I don't like the word numeric I don't even know what that means like but like it it's fine because we used big in
3:21:41
last time and it didn't work but I think numeric will work because I don't want to use text okay so we have us already then we have uh like month start uh we
3:21:49
call this a date um then we have metric name uh metric name is going to be a
3:21:56
text and then uh we need the array right so uh I'm just call it metric array and
3:22:03
then so there there's a big debate here about like okay like what type should
3:22:09
this be right is this like a real array or is this an integer array or is this
3:22:15
uh like you could say this is like a scoring class array right if you want to do like array of struct right but like I
3:22:23
I'm not about that life um I think we're going to do integer array here I guess like you could say integer or real right
3:22:30
because if you use real like real and in like they they work like you can put an integer in a real but you can't put a
3:22:36

real in an integer so uh okay fine you guys convince me or I convince myself we're going to use real um in this case
3:22:45
we have a primary key here primary key is going to be uh user ID month start
3:22:52
metric name right this is going to be our uh
3:22:59
table we're going to be working with today uh that we will be building up slowly but
3:23:05
surely so let's go ahead and uh create this bad boy cool so one of the things
3:23:15
about this that is kind of tricky is that you have to uh think about this in
3:23:23
terms of partitions and like Hive and uh or like partitions and things like that
3:23:30
and that's where this like can be a little bit messy compared to like in postgress whereas it's a lot cleaner
3:23:36
like when you have like that insert overwrite sort of mentality right because in this case we need to have
3:23:43
month start but it'll make more sense like I we'll cross that bridge when we get there but I I'll show you what I
3:23:49
mean so what we want to start with here is we want to create that daily
3:23:55
aggregate um function all right that's actually not too hard so let's go ahead
3:24:01
and say with daily aggregate as so we're going to be
3:24:07
pulling from events here I'm going to comment this out pull from events we're going to say select star from events and
3:24:13
then well what do we want from events well we want a proba want user ID and
3:24:19
then we probably want to count one as num sight hits and then we're going to have a a
3:24:26
wear here and then we should be able to do date of event
3:24:32
time equals date then we'll say like 2023 0101 so we're just going to do
3:24:38
we're going to do like the month of January and that's what how we're going to build up this array so if we do right
3:24:44

now and we say daily aggregate uh we're missing though we're missing we're missing the group
3:24:51
by so we got this guy we run him cool see we have all of our hits
3:25:00
okay that's a problem we need to get rid of that uh this null
3:25:08
so and uh I'm G paste this to y'all like because obviously I just like jumped
3:25:13
immediately into coding all right I was wondering how you are typing so
3:25:20
fast while speaking so good okay so we have our
3:25:27
aggregate for the day right so we like what we want to do is we need like
3:25:35
yesterday's Aggregate and so this is where like like for the purpose of this
3:25:42
lab I would change this because you you don't get this is one of the things I hate about postgress that I wish had
3:25:48
postgress had was like a merge because postgress doesn't have merge right yeah it doesn't have merge I think it has
3:25:55
like I think you can do on conflict on conflict update though right I think you
3:26:01
get that I think we can use on conflict update we'll try I I'll um I I might end up Googling a little bit here but so in
3:26:08
this case we have our daily Aggregate and we need uh last month's aggregate as
3:26:13
well like we need like uh we we need like what was yesterday because if you think about this in production when this
3:26:19
is running um uh like on January 1st the array will have one value then it will
3:26:25
have two values then it will have three values do all the way up right so that's something that we need to consider when
3:26:31
we are building this out so let's go and get like yesterday array as um and then
3:26:39
in this case we're just going to say uh select star from array metrics where uh in this case we're
3:26:46
going to say uh month start equals date 2023
3:26:52
0101 cuz we that's all we care about um so now in this case we can do like a

3:26:59
have our daily aggregate full outer join yesterday array one of the things that I
3:27:05
hope y'all like uh at the end of this data modeling stuff is that like you'll
3:27:11
recognize that every data modeling problem is actually the same problem where you just uh where you full outer
3:27:18
join all the time and obviously like if if I said
3:27:24
that on LinkedIn people would be like Zack did you have a stroke but um anyways uh let's just run this now and I
3:27:30
think this will make more sense so we have our site hits and then we have nulles across the board on the other
3:27:35
side like we would expect right so now what we want to do is uh we have all
3:27:44
this then what we want to do is we want to uh essentially create an array or we
3:27:52
want to create a new array for this daily aggregate if it doesn't exist and
3:27:57
then we have to fill backwards like essentially from that date so that's where we actually do need to pull in
3:28:03
this date here we so uh because this is going to be needed we say as current date because we need this uh oh we'll
3:28:11
call this fun we'll just call it as date I hate how postris doesn't like like current dates it's like a keyword so
3:28:17
we're going to need this and we're also going to need to group on it right because we'll need this to do the the
3:28:23
date math later for like there's an edge case where we need that I promise so
3:28:30
initially what we want right is we're going to have a a coals here of
3:28:36
da. user ID and ya. userid this is always the best part when you're just
3:28:42
like yeah I got my cols figured out and then now like okay so then month start
3:28:48
right month start can just be uh like another cols here so the cols here is
3:28:55

going to be between and this cols is really really gnarly actually so in this case we have um you have ya. month start
3:29:03
comma and then you have da. date but as a month start but this is not quite
3:29:11
right because uh this day moves forward so you got to like truncate this right
3:29:17
so you got to do like date trunk month uh da. dat so that like as we kind
3:29:26
of cumulate up this will still stay month start okay so now we have month
3:29:33
start then we have metric name uh good thing a good old metric name here is we call the site hits as metric name this
3:29:41
is something that like is usually hardcoded but it's good now we have the hard part building the
3:29:49
damn array so uh you can think about it in this first case when yesterday array
3:29:55
is completely null and completely empty it's like pretty straightforward because
3:30:01
we know all the users are on the other side so or they're only in the daily aggregate but they're not in the
3:30:06
yesterday array so we're going to essentially fill in the first one and uh
3:30:13
cuz I want to I want to really illustrate to y'all the pitfall here that can happen so in this case uh we
3:30:20
just want to do array and then um so we want to say uh case when ya. metric
3:30:26
array is not null then uh what we want to do is if
3:30:34
the metric array is not null we want to say why. metric array but we want to um so this is the next day though so this
3:30:41
is the opposite like from yesterday's date list one where we put the most
3:30:46
recent data first this is actually the other way around because we want everything to line up right that's one
3:30:53
of the things we want to do here is we want to have everything line up so what we want to do is we want to do a concat
3:30:58
here of and then we want array and then we have da. num sight hits but there is
3:31:07

uh this is uh another one of those edges where this could be um null and that
3:31:16
might be okay y'all might be okay with null like I think for this case I don't
3:31:21
like null I want to do a zero instead of null so that will be um so then we have
3:31:29
else okay so then then we have uh so if the metric array is not null that means the user already exists but then we have
3:31:36
when ya. metric array is null then for now what we're going to do is
3:31:44
we're just going to put in the uh this value here but this is actually
3:31:53
wrong and um I will explain why this is wrong here in just a second but you'll
3:31:58
see with uh this kind of daily aggregate this is getting us pretty close so you
3:32:03
see we have our user ID and we have our case when statement and it looks really nice wow someone someone hit my website
3:32:10
60 times on New Year's wow someone needs to get a life or maybe they just love my
3:32:16
data engineering maybe he's just a Super Fan I'm sorry um okay so then in this case this is metric array right want to
3:32:23
say as metric array this case we need to put like an insert into here insert into array metrics right and then on conflict
3:32:31
so in this case on conflict so our primary key here we got to put all of them here so we have on conflict then we
3:32:39
have all of those right and then we say set and then we say metric array
3:32:46
and then in this case we say equal to what is this even doing okay because we want it to be I
3:32:55
think we want it to be the excluded one because this is going to be the um the
3:33:00
the other record right and so I think this is just dot metric array I think
3:33:06
that's all we do so uh we'll see if this actually works uh but uh there is one
3:33:12
more bug here with this guy which I will uh kind of show here in a second but we had to get this on conflict right I'm
3:33:19

glad I got the on conflict stuff to work so we don't we can do it the right way um uh in a big data world like you don't
3:33:26
have to worry about this because you get overwrite right and overwrite just will
3:33:32
just like you don't have to worry about how to set the updates of things like
3:33:37
I'm like I don't know I I maybe I was I've been spoiled and I've just been using overwrite for so long that like I
3:33:43
just expect it out of every technology work with now and every time I'm like why do I have to update I don't like the
3:33:49
update keyword so um anyways uh let's go ahead and we should be able to run this query
3:33:55
now okay so we ran the query for day for the first day all I want to do here is I
3:34:02
want to run the query for the second day and just so I can illustrate the problem and and and check if this conflict thing
3:34:10
works okay well something works so if we say uh select star from array metric
3:34:16
we should have some here that have two values okay there we go perfect yes
3:34:22
everything everything is exactly what I was thinking was going to happen Okay
3:34:29
so remember in uh okay so you see how like for some of these metrics like you
3:34:34
see this first guy here he had six on January 1st and zero on January 2nd so
3:34:41
he didn't show up the second day right um this person was three and three just very consistently going to three pages
3:34:48
right and so um you'll see though like remember one of the things I said was
3:34:55
that every for every iteration of this
3:35:00
for every data set here regardless of when a user shows up everyone should have the same number
3:35:07
of elements in the array in this case they should have uh like these guys should have one more
3:35:14
they should have a zero at the front because this person essentially didn't
3:35:19
exist until January 2nd and that's what is going on here so what we need to do

3:35:27
is there is a so in this case we have the okay if the metric array is null
3:35:34
then we need to have this array but there's also uh it's really awesome so
3:35:39
we have an array fill function right and we need a concat here so the the array
3:35:46
fill function here is actually going to be equal to uh so in this case there's
3:35:51
you see we have month start so then we have uh we have date and minus month
3:35:57
start so this probably looks really funky but uh like this is what this like
3:36:05
so what this does array fill what this is going to do is it's going to so for the second or or or
3:36:13
let's imagine we're on the seventh of the month and a new user shows up then what this will do is uh date will be the
3:36:21
7th of January and month start will be the first so then this will be uh six
3:36:27
right you'll have six values that are there that need to be um kind of uh and
3:36:35
so what this will do is this will create an array of six
3:36:40
zeros right this it'll be 00000000 six times so one of the things I want to do
3:36:45
real quick is I want to like clear out array metrics though uh we're just going to we're just
3:36:52
going to clear them out cuz like and it's going to yell at me because it's saying there's no you see I love I love
3:36:59
that data grip does this so that like because you don't want to delete all the data but we do want to delete all the
3:37:04
data so um uh this that's what this array Phill
3:37:09
is going to do and what we're going to do is we're just going to move this back to January 1st and then we're going to run this two times
3:37:16
so we're going to we're going to run it for array fill integer
3:37:21
integer does not exist okay so this is
3:37:26
that's so weird so you actually give it an array like that that is so weird but

3:37:34
okay right was that like dimensional values cannot be
3:37:40
null oh is it because month's start is null right
3:37:46
interesting because that is null and then this is null because it doesn't
3:37:51
exist yet but oh this is this is an interesting Edge right so in that case
3:37:58
we have uh I think there's like a third condition here actually so when when
3:38:03
it's completely empty this is not going to work right so when it's completely empty though we can just have this first
3:38:11
array because we know that that date hasn't happened yet so what we have here is it's like when ya. Monon start is
3:38:20
null then we have that array okay I think this this should run
3:38:29
now okay it ran so if we look look at it let say like if we search here this
3:38:37
should perfect so the first day ran that worked great but then let's move it to
3:38:42
the second day so that we can uh just see this work working and then I will definitely send this query to
3:38:48
y'all okay so that should now we should get our
3:38:54
filled that did not give us the field zeros oh oh oh oh oh oh oh oh oh oh
3:39:02
because no that actually makes sense because oh because it's not matching
3:39:08
here right so that's still going to yeah this is
3:39:13
wrong actually like so we got to like essentially coales this cuz one of these
3:39:19
values is going to always be there right because essentially what we want is like
3:39:25
if both of these values are the same so that's so weird I didn't even like I
3:39:32
thought I ran into this problem before I I I know I'm like kind of fumbling here in a second but like let me let me go
3:39:37

over what we actually needed to do here and what's going on so the problem here is this array can't accept a null value

3:39:47

so what we want to do is we just need to coales this to zero so like if either of

3:39:55

these is null then we just don't fill because we don't need to fill because that means it's the first day of the

3:40:01

month right and that will fix our problem but now uh we have bad data

3:40:07

again so we have to delete from the array metrics but that will um we will

3:40:14

be uh will be good to go here just a second okay so that will fix our problem that's why

3:40:20

you have to you have to put a cols there because you can't put array bracket null because postgress apparently doesn't

3:40:26

like that which is again like one of those like today I learned sort of moments so I think this query should run

3:40:35

now okay but then if we change this to two this should run

3:40:43

now okay now now we should be good how is that still not like okay now

3:40:51

I'm wondering if like the update isn't working if if it's something with the update actually that is cuz here we are

3:40:58

getting our the array fill because because if you have the new date

3:41:05

oh oh oh oh oh oh oh I know what it is I know what it is it's

3:41:11

because month's start is still null because what we need to do is this

3:41:18

month start is not actually in this yesterday array this is a hardcoded

3:41:24

value this is actually not here because what's happening right now let's just

3:41:29

kind I'm going kind of go over what's going on right here right so we have the date here and uh we pull it in from the

3:41:36

array but we have a full outer join here right and so when I have this month

3:41:41

start value here this is not the right one because this is this is going to be null so on the second if someone shows

3:41:49
up and they don't exist yet this is going to be null but really in the
3:41:55
pipeline this is not like this value is fixed right so this is actually date 2023 0101 and it never changes that date
3:42:04
will never change that date's always the same so like that's why we're still getting buggy data wow that's a that's a
3:42:11
very interesting uh uh a very interesting change okay there we go now
3:42:16
now we'll be good with just like one more delete and I think we I think we got it here that's even better I love
3:42:21
that I love that yep but we're we're going to move it to that right because they so you're saying date trunk month
3:42:28
of date right like that yeah I like that better I like that better because then it's not hardcoded right because then
3:42:34
like you don't have to like if I want to change it to a new month I only I still only have to edit it up here right so
3:42:41
okay now keeping in mind that like this kind of array fill stuff it this should
3:42:46
work I'm okay we got to change this back to one though so you you'll see like this has the same uh pitfalls that uh
3:42:54
cumulation does ooh types interval and integer cannot
3:43:00
[Music] match what
3:43:06
coales date trunk because this is is it because this needs to be cast
3:43:13
as a date as well that's post chis is so
3:43:19
weird well because it worked before okay no yeah there we go it's because date trunk returns a time stamp that's
3:43:26
why so you got to wrap that in another date right like because that's like so
3:43:31
dumb okay some of this stuff like like all this silly little data engineering
3:43:37
stuff is okay now like my whole point is I just wanted to get it to be where

3:43:42
everything in the metrix array then there we go there we go I know that was
3:43:48
painful y'all but there we go we got it so one of the things to prove it out right is we can say cardinality of
3:43:54
metric array and then we can say count one and then we can see like how like everyone should have this should be two
3:44:02
right everyone should be two yep there we go 138 users everyone has two values
3:44:08
right and then this just keeps working too though like you'll see if we uh if
3:44:13
we go to three right and then everyone will have three values now right if we
3:44:19
kind of where did that query go there we go put that back and then change that to three but you'll see now
3:44:26
if we run this query now everyone has three so that's kind of the idea here
3:44:32
here let me paste this to y'all because that was there was that weird date cast that I think we missed that this query
3:44:38
essentially does it where we can build these things up and uh run all of these queries at once and we can get all of
3:44:45
the data obviously like uh this this line this line is this
3:44:52
line is absolutely nuts though I don't know if y'all like if you look at this line of code you're like what is this
3:44:57
guy doing here like this is so crazy but that gives us our our code for for that
3:45:04
right so one of the things that I wanted to show though that I think y'all will really
3:45:10
appreciate is um how to do the aggregation of this so that you can see
3:45:16
how we can go we can Aggregate and I'm just going to show how to do it with uh
3:45:22
with metric name and we can group on Metric name but then it will be obvious how like because you can join on user ID
3:45:28
and bring in other dimensions if you want but you can group on Metric name and that's going to make more sense for

3:45:34
now so I'm going to I'm just going to open a new uh query console here so if we say select star from array metrics
3:45:40
right this has all of our data and we have three three three days of data
3:45:46
right now right but we want to aggregate this and what we want back here is
3:45:51
dimensional analysis on 3 days and I'm just going to illustrate how this works
3:45:56
kind of for and then it will make more sense how this works like for like a month so we don't have to do the whole
3:46:03
accumulation thing but uh so what we want to do is we want to say metric name and in this case we want to say uh sum
3:46:11
and then we want to say metric array at one right right or and then we can say
3:46:18
and then we we put this back into an array this is it's so weird like this is
3:46:23
another thing this is another problem that I noticed that uh a lot of SQL
3:46:29
uh syntax stuff doesn't work the right way for this so now you'll see and then we can
3:46:37
say Group by Metric array so this query works right there we go so you see now
3:46:43
how we have and then we have month start here right so oh we got to put it in the group by
3:46:50
too month start so you see how now we have like
3:46:55
this is all added up though like how this uh this is like we have one record
3:47:00
here right so one of the things that you can do right is this part can be like
3:47:06
you can do like a like there should be like a unest like function here that
3:47:12
gives this should okay that worked so what you do is okay I know how to do it
3:47:19
though so you have a the array not unnested
3:47:26
right and we're going to call this as summed array and we say with um say a a
3:47:34
call this an aggregate for now and then what we can do is we can
3:47:40

say um select star from from a we could say cross join
3:47:47
unest and then in this case we have a do summed array with
3:47:55
ordinality right okay I think it works this way with ordinality this is like
3:48:02
this is getting absurdly fancy but I I I assure you that this is important why we
3:48:08
have this like index here we're gonna call this index though so if we query if we run this query
3:48:15
you'll see okay so now we have this index here right and uh postgress is
3:48:21
dumb and we have to do minus one because it does one based indexing but we need
3:48:26
to add one day to our month start so if
3:48:32
we say um so we have metric name then we say month start plus there
3:48:39
should be like plus interval can you do like one day or like is it interval in
3:48:46
yeah we can say day index minus one that is it like
3:48:53
that what there should be a way to do that like add one day so the idea here
3:48:59
is you can take this month and add one day to it like is what is the there's
3:49:04
does anyone know what the add one day function is like like like from an INT it's like you got like date ad or like I
3:49:11
always freaking get this stuff like date in date part date out date
3:49:18
trunk okay so problem here is uh remember it's zero based right so this
3:49:23
is actually the wrong day so we need to do index minus one because postgress
3:49:29
and okay perfect so now this is now we have our um and then we have um LM as
3:49:37
value right and then this is uh okay there we go so now we have our
3:49:44
date and we have you see how now this is back to um uh daily aggregate right um
3:49:52
but the thing is is like I want to show how this works really well though because of like how uh how like if we
3:49:59

just load in one more day into the metric right then this pops in and then
3:50:05
over here all we got to do is add it to the array here and then uh this part was
3:50:11
like I actually ended up writing python that generated this there needs to be UDF that essentially just does this sum
3:50:16
for you but uh so you see now we have the fourth and that just added it uh
3:50:23
very efficiently cuz the thing is is like this explode here is like we we
3:50:29
aren't like every user only has one record right and so we can sum everyone
3:50:35
up and it's like very fast because we just sum them up in the array and we don't ever explode the array we only
3:50:42
explode the array after everything is aggregated and so that's why like we have like one record for each metric
3:50:49
name but that's where like in here when you do when you have this um this sum here this is where you could join in
3:50:56
users to like get some other value right you could get like I don't know there's
3:51:02
other values that you could get here to like explode out the dimensions and so that you would have more here but then
3:51:09
you would have daily data with that dimensional value as well and so that's how you can go from monthly uh kind of
3:51:16
array metrics back to daily Aggregates but it's very fast because you have a
3:51:22
it's the minimal set of data that you need and um that's uh like this idea
3:51:29
like saved Facebook so much time and so much effort and energy congrats on finishing this 5-hour course doing all
3:51:36
the Hands-On exercises and getting to the end and sticking it out I'm really proud of you congratulations not very
3:51:43
many people get this far you really like this content make sure to like comment and subscribe and good job I'm excited
3:51:49
for you to check out week three [Music]