Fact Data Modeling

Fact Data Modeling Day 2 Lab
How Meta models Big Volume Event Data

Building a datelist data type

Transcript:

2:03:32
here is we have this if we say like select star from events you'll see we have this really
2:03:39
fancy table here that has um sometimes has a user ID sometimes doesn't have a
2:03:45
user ID so you'll see this is like what
2:03:50
this is is every um see some of these are like a
2:03:56
hacker trying to like get into my website there's also this is like essentially every Network request that
2:04:02
goes on right so I just kind of want to go over this because this is real uh I just want to show you how this works so
2:04:09
see this back robots.txt so this is on the host here is dubdub du. exactly.com
2:04:15
so let's go there real quick so if we go www. exactly.com robobots
2:04:22
.txt right and you'll see this gives us a little file here so uh this is like a
2:04:30
a file that's kind of hidden for most of the people on the Internet it's uh essentially what this is saying is uh uh
2:04:38
you see how I say disallow I'm like I'm saying for every user agent go wherever
2:04:43
you want my my I essentially say Google scrape me scrape everything take all my
2:04:49
data like you could be more aggressive about how you disallow things here I just want to show an example of that
2:04:55
real quick so if you go to Facebook and you go robots.txt you'll notice uh they
2:05:00
are a little bit more um like they don't want Google to just scrape everything you'll see like here Google so see so

2:05:08
here's like Facebook talking to Google it says okay you can scrape everything
2:05:13
except for these web pages so that's essentially how disallow works there's
2:05:18
like also an allow but what I'm trying to say for the purpose of this lab today
2:05:26
is you'll notice we have a bunch of data in here that goes for a bunch of
2:05:31
different websites and one of the things I want to show is let's look at Max event time and Min event time because I
2:05:39
think this will kind of help you illust help you understand like okay so you see we have this essentially has data
2:05:47
from uh January 1st of this year to April 27th I need to update this data
2:05:53
because I actually have more data here but um like this was essentially I pulled this data right at the beginning
2:05:59
of the last boot camp that was um why it's April 27th and uh but I'm logging
2:06:06
this data all the time so anyways what we want to do with this data you'll notice uh we have some other columns in
2:06:12
here so you see how we have this uh user ID column that's another great column
2:06:18
that we want to look at so what we want to do is essentially make it so that we can
2:06:26
see on uh we can we can cumulate this up and find all the days that different
2:06:32
users were active because you'll see there's all sorts of different users in this all sorts of different users right
2:06:38
and um so that's essentially what we want to do so what we want to do first
2:06:44
is we want to create a table we're going to go create table here and we're going to call it users uh users
2:06:52
accumulated and this table is going to have a user ID it's going to be an integer
2:06:59
well uh let's make it a big int right uh why is it big int instead of integer
2:07:05
because um integer actually has a limit uh which is two billion and this number

2:07:12
is much bigger than two billion this is like this is like in I think the
2:07:17
quadrillions I want to say it's a very very large number but a big big int is fine you get like uh I think you get
2:07:24
like 50 or 60 zeros with big int so that's good enough so then we have um
2:07:31
dates active and this is going to be uh we're going to call this we're going to call
2:07:36
this a date array and then we have uh we have call
2:07:41
this current date which is a date and oh does it not like that oh
2:07:47
yeah because you don't want to use current date because it's a keyword right um uh it's another way to do this um we
2:07:55
just call we'll just call it date then date's probably fine and so uh what's the primary key here
2:08:03
primary key here is going to be user ID comma date where date is going to be the
2:08:09
so I'm going to put some comments here and then I'm going to paste this to y'all right so uh dates active here is
2:08:16
the list of dates in the past where the user was
2:08:22
active and this is the current date for um the
2:08:29
user um so this is going to be our table that we're going to be working with
2:08:35
let's go ahead and kind of look at like what this would look like um on a given
2:08:40
day so uh we saw that everything starts on January 1st and essentially what I
2:08:47
want to do is I want to build something that goes from January 1st to February 1st or January 1st to January 31st and
2:08:54
uh we'll be we'll go from there so in this case what we want to do is we want
2:09:00
to create that same thing where we say like with today as and then or with
2:09:06
we'll call this yesterday as and then today as okay so in this case uh
2:09:14

obviously there's better ways to do this where you can like use Python to automate a lot of this stuff but
2:09:20
yesterday is going to be where we we we read from users
2:09:25
cumulative and we can say where date is equal to and keep in mind we're starting
2:09:30
with yesterday so that's going to be 2022 1231 that'll be the
2:09:36
um the day that like we start with because that's yesterday which is
2:09:42
because we're technically starting on the 1st of January and so that's what we're going to start with so what I want
2:09:47
to do is I want to paste this down here I'm we call this from events so in this case we have uh uh we should have event
2:09:56
time okay so this is going to be uh right now if we query this where we
2:10:02
change this to 2023 0101 right I think this query is going
2:10:07
to give us nothing back or it's going to yell right because this event time is actually uh is there not something else
2:10:15
in that table is there I it it honestly is
2:10:20
uh cuz event time should be able to be can we do like a I think we can cast this as a time stamp though right cast
2:10:27
as Tim stamp and then uh and then can you just do a
2:10:34
date I think this works like I think because the actual data type okay that
2:10:39
worked great so that's what we'll be using uh to actually work with things so
2:10:44
this will give us all of the things you just have to cast the the because it's actually a string in the in The Source
2:10:52
data but this is going to give us what the value was today
2:10:58
so we're getting close here so now what we need is we need essentially all the
2:11:04
users who were active today and
2:11:10

honestly this is where you can have a kind of a different whatever kind of
2:11:16
definition of active that you want there's all sorts of pages here right where I mean you could also be like and
2:11:23
um URL equals login right cuz this will
2:11:29
oh yeah there's none I forgot because it's like that doesn't start until L that doesn't start till later but um so
2:11:35
anyways you could think about like what event counts as active for simplicity's sake because we
2:11:43
aren't trying to create uh anything too crazy here because this is a onh hour
2:11:48
lab um obviously this would be different in uh like a business situation but
2:11:56
we're going to essentially start with user ID here and then we're going to say count one and then in this case we we
2:12:03
know that this date so what we want to do here is we want to essentially put
2:12:09
this date in here as well right and we don't actually need to have uh the count
2:12:15
my bad we only need the date and we can say this as date active and then in here
2:12:22
we can say Group by user ID comma this
2:12:28
guy right and so when we run this you'll see that this is a lot less right see
2:12:36
that this is now we're down to like 85 records but there is some more things that we want to work on here like
2:12:43
because there's other problems with this data um you'll see like you see the null here see how that there was like a null
2:12:49
at the bottom there and the reason why null happens there is because like I make a problem in my Dev environment and
2:12:56
like I can't infer the user ID based on Local Host so we just need another wear
2:13:03
Clause just to deal with the data problems here so we can say and user ID
2:13:08
is not null right so that we can get rid of this guy because he's going to be annoying later on uh with a full outer

2:13:15
join full outer join is going to really cause this guy to be super annoying so
2:13:21
we now uh have our today data and we have our users cumulative data so we
2:13:29
what we want to do is we essentially want to make it so that we create the
2:13:35
each day uh each day going forward right so like this is very like and I'm just
2:13:41
letting youall know this is going to be a very similar approach to what we did uh in the very first lab uh with
2:13:48
Dimensions so in this case what we could say like select star from uh today T
2:13:54
full outer join yesterday Y and then we want to say on we can say
2:14:02
t. user ID equals y. user ID and so if
2:14:07
we run this whole query there we go and you'll see that like essentially everything for
2:14:15
uh uh the the Y is null which makes sense because we haven't loaded in any
2:14:21
data yet so what we want to do is we want to get things to match this schema
2:14:26
here this uh kind of user cumulative schema that we are looking for so in
2:14:35
that case what we want is first is the user ID so in this case it's going to be
2:14:41
that same sort of coals thing where we say co. user ID comma y. user ID as user
2:14:48
ID that'll be our first one that one's that's the easy one then uh let's go ahead and get dates
2:14:53
active uh dates active is a little bit trick or the I mean then we have dates
2:14:59
active which I'm going to skip for now I'm just going to put null as dates active we'll work on that
2:15:05
in a second and then the last column we have here is date so in this case we
2:15:11
have uh right we have t. dat active which kind of works but t. dat active
2:15:19
might not um be there right because they might not actually uh exist yet so what we
2:15:27
actually want to do here is it's going to be another KS here we're going to say cols t. dat active
2:15:33

Y is that not okay then we want y. date as date um there's a problem here
2:15:42
though because y. date is going to be off because this is actually yesterday's
2:15:48
date and we want everything in here to be the same date so in that case you actually want to put plus interval um
2:15:56
one I think it's like one day it's like that or it's like one day I think that's
2:16:04
what it is postrest syntax this is so weird let me put interval like that okay
2:16:09
so we'll we'll test this out but now we see okay we have a good date and it's
2:16:16
always uh that date there right that so far it's looking pretty good uh we are
2:16:26
um we just need to do that dates active column now to really understand like how
2:16:31
we can get the right ones here so this column is a little bit funky because uh
2:16:38
we need to be collecting the array of values here so we can say like case so
2:16:44
like what's let's consider the first case when uh yesterday's array is null
2:16:49
so we can say case when y. dates active is null in that case we can say then and
2:16:56
this one's easy where we say array and then we say t. dat active done right but then we have uh
2:17:05
when y. dates active is not null or or or or I guess there's just an else here
2:17:10
the else here is going to be uh y. dates active
2:17:15
concat uh pretty sure that's concat and then we have um array of T.D
2:17:22
active so and then there's an end here so this is going to give us
2:17:28
essentially what we're looking for but there is one more wait there's actually
2:17:34
one more uh thing that we got to uh look for which is actually we have another
2:17:40
when here and the when here is when. T.D dat active is null then y. dates active
2:17:48
because we don't want to just keep keep adding a big old array of NES so let's run this query kind of show
2:17:56

you what I mean and then I'll paste this to yall so so far it makes sense like
2:18:02
it's kind of boring it's exactly what you would expect like kind of kind it's kind of a snooze right now and uh that's
2:18:08
totally fine so in this case uh what I want to do is I'm I'm G paste this to y'all because I think that this is we're
2:18:16
getting a lot closer to kind of what we're looking for and I actually made a mistake here and we actually do want to
2:18:22
concat at the beginning so that we have uh it's it's a clearer picture that um
2:18:27
it doesn't matter that much for the date list because the way we're going to be generating the date list it's not going to matter as much but um for uh
2:18:34
generally speaking uh you want the more recent dates to be the the lower indexes
2:18:39
of the array and then like you you you essentially pop it in on the front every day every day that the new data is
2:18:45
active instead of popping it in out the end so this is essentially it then all
2:18:51
we need to do here is we need to put an insert into uh users cumulated at the top here and then what we want to do is
2:18:59
we can uh run this uh we have big int out of range it's saying that
2:19:05
like but user ID here is definitely numeric there's a null no because the
2:19:11
null is not going to be in there because we have a and is not null right here dude so well like let me look at the uh
2:19:18
one second let's just look at the whole query here and see like what was going on here cuz or this is me what is going
2:19:25
on let's stomp some of this okay this query here run this guy oh
2:19:34
wait so is this out of range is this like
2:19:40
because if we it's I well we can see
2:19:46
right is that like too big like okay okay wow this is actually
2:19:52

outside the range of big int well it looks
2:19:57
like that's really strange okay well I guess like how did this lab work then I
2:20:04
guess uh we're not going to be using an INT or a big int here because it's uh
2:20:09
the data is a little bit different or something uh uh there is not a big int
2:20:14
is actually the biggest integer type so in this case that's fine it doesn't matter that much so what going to do is
2:20:21
apparently we're just going to drop the table and move them to uh we'll move them to a text and text will work fine
2:20:30
and uh I did not uh last time I did this lab this worked fine though so I don't
2:20:35
know maybe I did like a new thing with post press or something but there's your uh if you just drop the table recreate
2:20:40
that will be it I'm going to um comment all this out so I don't have to keep highlighting everything so what we want to do here is we're going to cast this
2:20:46
as a text or actually we don't need to cast it as a text because it's already a text um we need to cast it as a text
2:20:53
here right because it's not it's not a text in the event data right it's
2:20:59
numeric which is super weird that it's numeric that's like that seems off that
2:21:05
like okay so now okay now this is working this is looking better oh you
2:21:10
see how it's off to the left side now because it's a string but um so now we can take this and we can say insert into
2:21:18
users cumulated and this is going to give us good data so let's just like make
2:21:25
sure that like that data is like what we're looking for always like to do that before we like start the cumulative
2:21:30
process so you see this is looking this is looking right where we have our user ID our dates and our date but
2:21:38
um that is interesting uh so now what we want to do is we want to essentially
2:21:44

build this up a little bit so we're going to change this to 2023 011 we make
2:21:49
yesterday that right and then now we run it again so now if we say like select star
2:21:56
from users cumulative where date equals 202312 just to show you how like some
2:22:03
users here are going to have yeah see now this guy's got two values two values
2:22:09
and then most people are going to have one value but some people are coming back right and so this is going to be uh
2:22:16
how we can kind of model the growth of each of our users here so this is
2:22:21
getting pretty close so now we're just going to do a little bit of a manual exercise here of just like we need uh we
2:22:27
need 30 days here so this is going to take a little bit of time but it's not going to be too crazy so just going to
2:22:33
uh run this query again for uh three and
2:22:39
then four and then five
2:22:52
six so this is like the good thing about it doing it this way there's probably going to be a better way to like run
2:22:58
this where you can run it all in one query but like one of the things that I'm trying to emphasize with y'all is
2:23:04
like how I do this in a one hour lab versus how you do this in production as a data engineer are going to be
2:23:10
different right and then you have like kind of more of this incremental way that you want to build things up versus
2:23:15
how like like those kind of large backfill queries a lot of the times are not going to be as performant as you
2:23:21
would expect so that's kind of the idea here where we can kind of build this up
2:23:28
and this will give us kind of access to our kind of cumulative query that we're
2:23:33
looking for this is going to be uh we're almost
2:23:39

we're almost halfway there yeah and uh this will essentially be like how all
2:23:46
this will build up and then we'll be able to then turn this query into the date list and then that will and then
2:23:53
youall will be like wow that is very efficient and then um from there uh
2:24:00
that's pretty much what we have for the lab today but uh we're going to do a lot more like kind of kind of showing the
2:24:06
the different bit maths so you'll see like in the date list query that we are going to be working with today uh we
2:24:13
actually use a bit 32 for our
2:24:18
um uh like L data type and that datelist data type is going to be uh it it's
2:24:26
interesting CU like you do uh it's the same thing like you know where y'all were like is there anything bigger than
2:24:31
long so long is 64 bits right and which is like I thought I had everything
2:24:38
covered there but apparently I didn't and uh so you can technically do this
2:24:44
for 64 most of the most of the um uh the ones at Facebook do 32 because
2:24:51
like you only really need uh 32 cuz it luckily there's 32 bits in an in an
2:24:57
integer and then there's 30 days in a month right so you have like just enough bits in an integer to like fit it so you
2:25:05
have like one or two extra to like do do your kind of stuff with right and so um
2:25:12
okay almost there last one so 30 and
2:25:17
31 okay so now let's look at that real quick so I can say select star from users cumulative where date equals date
2:25:26
2023 0131 so now this should have like a lot of data okay so then you'll see here's
2:25:33
people who like come back on some days right and then you'll see that like we have a lot more users in here now
2:25:39
because this is essentially any user that was active at least one time and a

2:25:45
lot of users here are going to be like kind of oneandone users like you'll see we have over 500 right because I only get five 00 in here at any any one go so
2:25:55
um now we have our data right we have our data here that's going to work
2:26:00
pretty well to kind of create our values here right so now we have the tricky
2:26:08
part of uh how do we turn this into uh a
2:26:14
date list like into that integer value that that this part is going to be really funky but I think yall are going
2:26:21
to like it so if you think about it we want to think about this like going backwards again right so uh where the
2:26:29
the most recent data is first and then the the the oldest data is the last um
2:26:35
bit so that actually means in terms of bits though that uh it's actually the
2:26:41
the first bit right is so it it depends on like how you're going to be adding these numbers is it the the the bit that
2:26:47
represents like one or two or or four or8 or 16 da d d d d da or you can think
2:26:53
of like all the different powers of two that you can do there um one second I want to see like there like I have that
2:27:01
in here right I have the this should be a math.pow in here somewhere I have the
2:27:08
this is users cumulative we have dates dates active you have bits okay yeah because
2:27:15
there's the uh you essentially do um so it's the most uh the the bit that's the
2:27:21
leftmost bit like is going to be your uh most recent bit that's yeah that's what
2:27:27
I thought that's like glad I can got the example here so let's go ahead and look
2:27:34
at how to do that so in this case we're going to be using uh we're going to
2:27:39
generate a date list uh for 30 days that's going to be the idea here so
2:27:45
let's think about how to do that so one of the things we're going to be using today is we're going to say um I don't

2:27:51
know if y'all have ever seen generate series but let's kind of go over how
2:27:56
this works so we're going to generate a series from uh the 2 to um the 31st
2:28:06
right so you'll see this is actually a valid squl right here we can actually query this date to date does you might
2:28:15
need explicit types generate series does not work with dates
2:28:21
does it need to be like a string can you do with does it does it work with string then if not like uh I have an idea of
2:28:29
like how to do this okay and then if you do what if you like one to 32 or to 31
2:28:36
does that give you does this give you what you want perfect that is so um so silly that that
2:28:44
works but okay so we can we can essentially do it this way where we can uh like we can gener a series that way
2:28:50
but like I swear there's a way you can do it with dates ah ah okay so it's yeah this I'm
2:28:58
just getting the syntax all weird so let's go ahead and um do it with dates because dates are going to be better so
2:29:04
2023 011 to um date and we have 2023 0131 we
2:29:11
can so this is a great example of like do you use 31 bits or 30 bits or what's a month right so at Facebook a lot of
2:29:17
the times like we actually considered month 28 days because of the fact that
2:29:23
then you have the same if you have a month as 28 days you have the same number of Mondays Tuesdays Wednesdays
2:29:29
and Thursdays in that month and you actually kind of get rid of the seasonality that way so this generate
2:29:35
series works right great so this is the generate series that we're going to be working with today to kind of work with
2:29:43
our data so now um I'm going to uh just uh what I'm what I'm going to do is I'm
2:29:49
going actually just open open a new tab so let's get this so I can keep all this

2:29:54
stuff as well I want to grab these three lines okay so we have our users
2:30:03
cumulative table right so I'm going to say that we're going to say today's we're going to say we're say let's call
2:30:09
this users we'll call it users right this will just be our filtered table here that we're going to start
2:30:16
with then we want our um uh we'll call this maybe series I'm going to do this
2:30:21
like with a lot of CTE and there's probably um another way to do this that might be cleaner right and then in here
2:30:27
right this is going to be so if we say like select star from series what is this called is I think we want like a
2:30:34
okay it's called can we put like a as um date does that actually work if you put
2:30:41
it there it does work okay so this is going to be our um uh we'll call this
2:30:49
series date I think that's a better name okay so now we have our series date
2:30:55
and we have our users cumulative so these two tables together are going to
2:31:01
essentially we're going to need to join these tables um one of the things about this is that
2:31:08
this join is uh kind of an interesting one because you have uh so we know that
2:31:17
date here is going to be fixed so we also need to figure out like what the date diff between uh two days is so that
2:31:23
we can get days sense so first off let's do uh select star from from users and
2:31:30
then what we want to do is we're going to say um uh we can just say uh cross join
2:31:38
Series so let's just look at like what this looks like so we should have uh a lot of
2:31:46
these Series dates right you're going to have like so let's let's down to one user here so it's user ID equals that
2:31:54
user ID so you'll see what we get here is for this user ID we have we have all
2:32:01
the dates that we're looking for this is exactly what we're looking for so now
2:32:08

what we need is a way to see
2:32:14
if the series date is in the active array and if it is then we create a bit
2:32:24
value that like probably isn't super um
2:32:29
so let's let's just look at it this way so I'm pretty sure it's this is going to be so if we do a comma star I think it's
2:32:39
this oh is it the other way I think it is a dates active it's like
2:32:46
that oh this is time stamp oh we got to put this as got got to put this as a
2:32:51
freaking date operator does not exist date data
2:32:57
Ray okay it is the other way around I got the data types
2:33:04
wrong right okay what there what is it I have
2:33:13
it oh it needs you have to compare arrays like this it's like a weird like
2:33:18
arrays thing right where you have like you have to wrap the other one in Array I remember like when I was doing this in
2:33:24
the last time I did this presentation like this is uh so so instead of valid
2:33:30
date here we have series date which I think is a better name and uh apparently I got the uh I got this wrong the other
2:33:39
like so now you'll see with this user you'll see how um okay so you see how he
2:33:44
has like the the January 30th as a date right you see this like this column here
2:33:50
is now checked great that is exactly what we are looking for um so this is going to
2:33:57
be um this is great we now like we we essentially want to put a case here so
2:34:03
we want to say case when we have this right then then uh like what we want to
2:34:10
do is we want to know the number of days between two dates and I think that works
2:34:16
where we can just say um date minus date Series dates series date I want to make
2:34:23
sure that this actually works because this was I I'm getting back to

2:34:28
like okay cool this does this this works exactly what we want so this is the number of days between uh the current
2:34:35
date which is filtered down to the 31st and uh the series date which is
2:34:41
generated from this generated query here so we're going to use this and math to
2:34:47
essentially create uh an integer here so in this case what we're going to say is
2:34:53
so we have that uh case when statement here uh so if they're active on that
2:34:59
date then what we have is uh pow right
2:35:05
and then you have two to that power but then it's uh 32
2:35:12
minus that right so that or yeah we'll put this in like paren like that uh L z
2:35:20
n as um in we'll call this like placeholder int
2:35:28
value and then a comma so let's just look at like what this
2:35:34
does so you'll see like okay for uh the 30th right they get the 30th value but
2:35:42
they don't get the 31st so this is going to be how we can work with uh creating
2:35:48
our date list in for our user here so we're really close
2:35:55
here um this is going to be we're going to call this uh we let's call this um
2:36:01
place um holder int ins as so because one of the
2:36:08
things that like I want to uh kind of show here is we're going to say select star from placeholder ins and I will
2:36:14
definitely get you guys this query in just one second so if we cast this as
2:36:20
bit uh 32 I think uh this will make more sense
2:36:26
cannot cast type double as well can you just cast this as
2:36:32
int you have to cast this as integer as well integer out of range oh is this big
2:36:39
int okay well there we go so one of the things I want to show

2:36:47
here is you you'll see when you have all of these values when you cast this to a
2:36:55
bit int you'll see how like most of these
2:37:00
are going to be like just a bunch of zeros except for one of the days where they are active you see there's like
2:37:06
that one one and so that means they were active what that's uh 1 two 3 four five
2:37:12
6 7 8 n nine days ago so this will be nine days ago that they were active and
2:37:18
then you can kind of work with these bit masks to uh kind of figure out but like
2:37:23
we want the whole history because you can see like they were active like many days in a row right here but we need to
2:37:28
like essentially sum these integers up and that's how we can get back to this so we're not going to uh cast this as a
2:37:36
bit 32 quite yet so what we want to do is we're going to say uh we're going to
2:37:42
put user ID here and then we're going to put some placeholder int value we're
2:37:48
going to say group Group by user ID so
2:37:54
this some record does not exist oh it's because it's place oh my bad it's
2:37:59
placeholder in value like that and then uh let me just show you like what's going on at this point okay so we have a
2:38:08
user and they have this funky sum value right and it's like what does this sum mean but if we take this and we say
2:38:16
we're going to cast as bit 32
2:38:21
oh yeah is it is it cast as big int like why does it like it really wants me to
2:38:27
do this twice okay so you'll see for this user uh now we have uh kind of
2:38:35
our our history here right so you'll see if I like remove uh this uh filter here
2:38:43
now we can we can get the history of all of the users of like what days they were active like see this guy was active two

2:38:50
days and then you can kind of see like oh this person like came back a lot and
2:38:56
so let me explain like what's going on here because obviously this code here
2:39:03
is probably the like like what is going on so okay so let's let's literally step
2:39:11
through this like line by line by line right so in in our user cumulative we
2:39:17
have a list of all of the dates where a user is active right so if we query this
2:39:23
we have this big list of dates that they're active so this person is active on three dates right so what we can do
2:39:31
is if we use if so what this is doing is we're using uh powers of two right so if
2:39:40
you think about it where like if they're active today then we get to add um two
2:39:46
to the so if they're active today that means that date minus date series is
2:39:51
going to be zero so what that means is this is going to resolve to PO
2:39:59
232 which is going to be 2 to the 302 power and so um what this does is this
2:40:05
is a hack right and what it does is it actually converts all of those dates
2:40:10
into integer values that are all powers of two and that's why like before like
2:40:17
if you cast this right and like you saw how like um we also have uh okay so
2:40:24
let's just look at both of these together so you'll see like for some of these users like especially the ones
2:40:29
that have like okay so you see this guy has got 16 here right so it's because he was active like a really long time ago
2:40:36
and you see that like uh that 16 or or you see how he was active so he was active like like very very long time ago
2:40:43
right it was active like 27 days ago 28 days ago and so uh you have the whole
2:40:49
history but you'll see that like the actual integer value here like especially for users who were active on
2:40:56

one day is just a Power of Two And the reason why this is useful is if you cast
2:41:03
a power of two as bits and you turn it into binary and you actually go into
2:41:08
like the binary code right then the power of two actually pops out at you
2:41:14
right and you see how like oh wow this power of two is now um like a history so
2:41:19
if you sum these up that's why if you sum them up like you actually can get like the whole history of the ones and
2:41:24
the zeros and that can be a really great way to understand like what is going on
2:41:30
with this user so this is a very um a powerful way of describing each user
2:41:37
because now you see how like each user has just an integer value and this can give you how many days they were active
2:41:43
in the last 30 days cool all right everyone let's uh kind of go through uh how we can look at this and see um if
2:41:51
someone is U monthly active right so what we're going to do is uh postgress
2:41:58
has a cool function here called bit count it's like and for some reason it doesn't like this right so this this
2:42:07
query does run them okay cool so one of the things you'll see is in this case we
2:42:15
have um all these users and they're all running their data and you'll see like
2:42:21
this bit count here is really powerful because essentially what this is showing is how many day and you'll see some of
2:42:29
these users actually do have zero right you saw how there are users here who because they were maybe active on the
2:42:35
first and then they like never showed up again and so uh you'll see how like some
2:42:40
of these users have one value or then they like and then you have people who are like really high up on the list
2:42:46
right who have like uh see this person is active for 21 of the the last 30 days
2:42:53
so one of the things you can do is in this case like like you can say bit

2:42:59
count and then you say greater than zero as dim is monthly
2:43:05
active so that is a great way to see if a user is monthly active or not is you
2:43:13
can just do that that is uh and then that will give you the number of monthly
2:43:18
active users because you just know like at least one bit is on like and so it's that's pretty
2:43:25
powerful I I feel like this this function is really powerful but what if you want to do other ones right like say
2:43:31
in uh like you have other ones like where you don't want to look at the whole string of bits but you want to
2:43:38
look at a a segment of them like uh another very common use case that comes up at Facebook is weekly active right so
2:43:46
I'm going to show you how to do that and this one is a little bit more um dicey and not as um uh elegant as bit count
2:43:54
bit count is so beautiful but uh let's let's go with this one so what I want to do is I'm going to do cast here and
2:44:02
we're going to bit 32 so um this what we want to do here is we
2:44:11
want to put the first well that's one two three four five six seven so we want
2:44:16
to put the first seven bits here as flipped right cuz that will mean that they like so then what you can do is
2:44:25
there it's it's really cool so what what we want to do is we want to take this cast here the one that has the um uh the
2:44:33
other bit count in it right so if we have this and we cast this as um bit 32
2:44:40
there's actually just a single um you'll see there's like a single Amper sand and
2:44:46
let's go ahead and look at like what this does because I think that that's probably uh kind of cool to see uh
2:44:53
you'll see that so this is that uh so what this does is this is called bitwise
2:45:00
and so essentially what this does is for the first seven so how and works right

2:45:07
if you have an and gate in uh like electrical engineering if you have a one and a one then you get a one on the
2:45:15
output but if you have a one and a zero you get a zero or if you have a zero and a one you get a zero if they're both
2:45:21
zero you get zero so what this is doing is we're we're throwing these bits into an and gate and then everything that's
2:45:28
after the first seven days is zero so because we don't care if they were active eight days ago for weekly active
2:45:35
we don't care we only care if they were active in the last seven days so what we can do here is we can use this thing
2:45:41
called bitwise and and so this is going to give us uh so what you can do with this is you can do bit count on this
2:45:49
whole thing and then you can say greater than zero as dim is weekly active and
2:45:54
that will give you uh you'll see now you can see that like obviously the people
2:46:00
who are monthly active are going to be or the people who are weekly active are going to be monthly active but not the
2:46:05
other way around someone could have been active like two weeks ago and then like this guy here he's active a while ago
2:46:10
but not active this week so you can do these cool like bitwise operating
2:46:16
operating functions that can uh kind of take this dateless structure
2:46:21
and then you can see like oh is this going to be uh the right uh values for
2:46:28
us right and obviously the you can do daily active here as well like where you just essentially copy this guy again and
2:46:35
you like you just instead of putting seven here
2:46:42
right you put you put just the one and then this is dim is uh daily active
2:46:50
and this will give you uh all three where you can see like okay are they daily active right and you can see all

2:46:57
the people who were daily active weekly active and monthly active on that given day and that is like essentially how you
2:47:04
can very quickly use this structure because again this and here and this
2:47:12
bitwise and is also an extremely efficient operation like computers are
2:47:19
to do this like when they say like computers work with ones and zeros like
2:47:24
that's literally what we're doing here is like we're working with binary and but what we can use is like we can work
2:47:29
with binary and then we can uh get our uh kind of business questions answered
2:47:37
from the binary and so that is kind of what this does and you can build really
2:47:43
powerful things obviously you have to teach your analysts how to do this stuff
2:47:49
but like is this that much harder than teaching them how to work with arrays like and do stuff like this right it's
2:47:56
about the same and like but if you can get people on the same page on how to generate these but or like don't right
2:48:03
and what you do is you just put that behind the scenes as a data engineer and then you give them the monthly weekly
2:48:09
daily active flags for that day and then they can work with that right and they don't have to work with the history they
2:48:16
don't have to work with the bits because like honestly data analysts aren't going to want to do that anyways you know
2:48:22
congrats on getting to the end of the day two lab those bit functions are freaking nuts right I'm excited for you
2:48:28
to check out the day three lecture in lab where we do reduced facts if you're watching this on the platform make sure
2:48:33
to skip over to the next link so you get the credit that you deserve uh yeah thanks so much okay why should Shuffle