Dimensional Data Modeling

Dimensional Data Modeling Day 1 Lab
Data Modeling - Cumulative Dimensions, Struct and Array - Day 1 Lab

Transcript:

0:00
[Music] welcome to dimensional data modeling day
0:07
one lab so today we're going to be doing more of a Hands-On exercise to really
0:12
Master the concepts of struct and array inside of postgress uh to be ready for this lab make sure to blone the GitHub
0:19
repo in the description below you'll be able to see the homework there and follow the lab there make sure that you
0:24
have Docker installed and be able to spin up your postgress instance via Docker and make sure you o have a client
0:31
a SQL client installed like we're going to be using data grip in the lab today
0:36
but data grip is paid after a 30-day free trial so you might want to use something else uh the other options are
0:42
not as good you have like DB visualizer or D Beaver those are two other options
0:48
that are free forever that you could possibly pick from to if you don't want to spend any money at all and if you
0:53
want to learn how to do this type of data modeling with Hands-On exercises in the cloud I recommend joining the data
1:00
expert community and there's a link in the description below so the main table we're going to be working with today is
1:07
um we have a table called player Seasons you'll see we have um Let me refresh
1:12
this this is like my connection there is like broken we should have okay there we go so you'll see in player Seasons we
1:19
have uh every row here is one player in the NBA and then they'll have like their

1:24
seasons and their stats um one of the things that is rough about this right
1:30
right is that we have that temporal problem with this table where uh if we ended up joining this table with a
1:36
downstream table it would cause that uh shuffling to happen and then the we
1:41
would lose compression so what we want to do is we want to create a table that
1:47
has one row per player um and and then it has an array
1:54
of all of their Seasons so that we essentially remove the temporal component or we push it into uh its own
2:02
kind of data type inside right so let's let's just go through this data table
2:08
real quick to look at what things are actually part of the season and what
2:15
things are part that are not really changing right so obviously player name is not player name is an attribute of
2:23
the player so that one will stay the same um I think that uh height obviously
2:28
is probably not going to change College not going to change country draft year draft round draft number all these are
2:34
going to be uh like these are the same every time right you see how like they're just duplicated so these are
2:40
things that like you see how like this this data table actually has a lot of duplicate data in it that we can we can
2:46
probably work with to get rid of if we model things more correctly and then
2:51
you'll see we have all these other kind of uh but then once we get to this GP which is games played now we are in the
2:59
actual attri attributes of the season and these are going to be all of these at the end here these are all the
3:05

different attributes of the Season including the season itself the season of um name so what we want to do is we
3:13
need to create um a struct right we're going to call this
3:18
season we'll call it season stats so um and in postest you can do
3:23
create type so the first thing we're going to have here is a season and this is going to be an integer
3:30
is it is it like oh yeah I forget you got to do um oh yeah there's no colon
3:35
there we go so we have season integer right um we can look at some of these other ones that are probably good uh
3:42
like we probably want to get uh I'm not going to pull in all of this because that's just going to make the writing the sequel really obnoxious I think that
3:50
really the only ones we care about are these like first three games played points rebounds assists the rest of this
3:56
is like super nerdy and like we don't really need it so we're going to put in all the rest of these so games games
4:03
played is also an integer right because it's a you you can't play half a game
4:08
and then you have points right which is going to be a real
4:14
you have Reb which is a real and you have assists which is a real and I
4:21
think I think that's it I think that that's all we really have right is um
4:27
for this uh data table is or for this struct so this is going to be its own
4:33
data type right um that we are going to have here is going to we call it's the season stats struct and we're going to
4:40
create this data type awesome now we have our season stats struct now what we want to do is
4:47
we want to consider a new table that what it will be
4:54
is um hold up yeah yeah what it will be is it will be all of The Columns that

5:00
are the player at the player level so that we don't duplicate them and then we
5:06
will have the array of season stats that will be the other kind of big piece of this puzzle right so let's go ahead and
5:12
we're going to say create table players and so in this case um we can
5:20
I'm going to comment this out comment this out so um the first obvious column
5:25
we have here is player name which is going to be a text right um so age age
5:30
is an interesting one right because age is technically actually uh a property of the season right it's not actually a
5:37
property of the player because it really depends on the um uh the season value
5:44
and that will give us our age but we can also figure that out based on what their first age was so um but like we're just
5:52
going to ignore age for now but we can also keep a height Heights a a text I hate that Heights a text but it is and
5:58
then we have College right which is also a text right uh and we'll just uh we'll just ignore weight as well CU I think
6:04
that's one that that can change um then we have country text draft year so draft year is
6:12
also a text even though it looks like an INT because you have undrafted here I draft here is a text and then you have
6:19
um draft round which is a text and draft number also a text because you have
6:25
undrafted is also a value so that is going to be are essentially these are
6:32
the values that don't really change in this player's data set then we have uh we have one more
6:39
column here which is uh we have season stats right and this is uh this is a
6:45
season stats array right so this is uh where um you
6:51
might be seeing things that are a little bit new to you um there's one more

6:57
column in here that I like to talk about which is going to be uh it'll be season
7:03
but it's actually what I call current season and this is going to be an integer um because we are going to be
7:10
developing this table cumulatively and so the the current season is what like as we like do the
7:17
full outer joins between the tables this current season will just be whatever the the latest value in seasons in the
7:24
seasons table is and uh that will make sense as we kind of build this up uh so
7:30
we have our uh but like what's the primary key of this table the primary key here is going to be player name Comm
7:37
a current season it's going to be the um because uh like it will build up and
7:42
build up and build up and so this will be our unique identifier on this table
7:48
and what we're going to do is we're going to uh go ahead and create this so we have our players table and it is um
7:53
good to go now what we want to do is we want to think about how we do this full
8:00
outer join logic because I think that that's one of the things that is going to be interesting so first off let's
8:06
let's figure out what the first year of in this table is right so let's do select Min season from player
8:14
Seasons so this is going to give us like whatever our first year is okay so we can go back to 1996 awesome um that's
8:22
going to be our first year that we work with so let's go ahead and I'm going to
8:28
show you how this today and yesterday query ends up working right so we can say with um we're going to say
8:36
yesterday as and then we're going to say um select star from players where
8:43
current season equals 1995 I know this is going to be strange
8:49

but just just go with me on it right now you'll see and then in today we're going to say select star from player Seasons
8:58
where season equals 1996 so this is going to give us the
9:04
cumulation between today and yesterday so what we can do here right is we can
9:11
say select star from today full we're going to say today T full outer join
9:18
yesterday Y and then in this case we're going to say on t. player name equals y.
9:24
player name that's going to give us the um the values that we're looking for for
9:30
so let's go ahead and like do this query cuz I think that this is like probably a lot so you'll see one of the things
9:36
you'll notice is like as expected everything from yesterday is just null
9:43
you see that just null like as we would expect because it's just all null um
9:50
which is great that's like where're that means that uh like now what we want to do is the remember the first the first
9:57
thing you want to do is you want to coales the values that are uh not temporal like
10:04
essentially the values that aren't changing so in this case the first thing we want to do here is we want to say
10:09
cols want to say t. player name y. player name and this is as player
10:16
name and then what we want to do is essentially copy this guy a lot of times
10:22
because we're going to do boom boom boom boom and then we're going to do uh we got to do all of these right we have
10:27
like height height height then we have uh what
10:35
college College college and then draft year draft year draft year and then
10:44
draft round draft round draft round and then one more here and then that's a draft
10:52
number and then draft number and then draft number so let's look at this query

10:58
now and uh just see like what this does right oh yeah we got to actually do the whole whole thing here so you'll see now
11:04
we have essentially the same query like you'll see like this is now like you
11:11
probably like Zach that was the most that was like the lamest thing you've ever done there like you just essentially queried uh that's just today
11:18
right this like it only manages today's data but I promise that's because this
11:24
is what's called the seed query for cumulation and you know it's the seedery because this first players is null right
11:33
you see how there's nothing in there right so since it's null that's that's what that's what makes it the seed query
11:39
and um that's why the full outer join here is just taking everything from today but that won't be the case as we
11:46
build this up as we cumulate more and more so the next piece that we need to put into this query is we need to add in
11:55
the um the seasons array right so what we want to do here is we want to say uh
12:02
we're going to have an array uh and then in this case what we want to what we want to do is we want to say we want to
12:09
see if the um if yesterday's value is there right so let's do that first we're
12:15
going to say case when y. season stats is null then right and then we're going to
12:23
have a um an else here uh and then I'm just going to put an array and we're
12:29
going to put like we're going to work with this for now so why we need to do this is um we're
12:36
going to be doing an array concat here and what this will do is it's going to
12:42
slowly build up the array of all the values here so in this case what we want to do is we're going to build up an
12:49
array of row and in this case we need this row to be these values season GP

12:56
points rebounds assists right so in this case we can do t. season t. GP t. points
13:04
t. Reb t. assist right so and then in
13:09
here we need to also do a um you got to cast it as like a um uh colon so this
13:17
colon colon here will cast it as the type that you're looking for it will take this row and turn it into the
13:22
actual struck type type that we defined so this is only if it's null right if
13:29
it's not null then what we need to do is we need to say y. season
13:34
stats um and then we need to conat this right that's going to give us
13:42
um the current value right uh or this will give us if uh if it's not null
13:48
right so let's go ahead and run run this query okay so you'll see now everyone
13:56
has this uh this nice little uh struct thing here we're going to call this as season stats right and then um the last
14:06
piece here right is oh wait yeah okay there's actually one more case here
14:12
because of the fact that like we're only managing if uh because we don't want to add to the array if uh if today's value
14:21
is null for like a player that does that has like retired because we want to hold
14:27
on to that player's data but they we don't want to keep adding more NES to the array right so what this is not an
14:33
else here this is actually uh we're going to say when uh this is uh t. season is not
14:43
null then and then we have one last else here and the else here is y. season
14:50
stats so that we don't add a bunch of nulles that we don't want to add to
14:56
right that would be because we want to so this is going to cover all three cases right so it's like if it's null we
15:02

we create the initial array with one value if um if today is not null uh then
15:10
we create the we create the new value right and then otherwise we just we
15:16
carry the history forward and this means this is like a retired player who doesn't have any new
15:22
seasons so the last thing we want right is this current season value so current
15:29
season is an interesting one right so in this case we want to say um um case when
15:36
and then we're going to say t. season is not null then t. season and then we want
15:41
to say else y. current season plus one and so what this will do is this is
15:50
going to give us our current season value uh because it will either take the current season or um oh there's actually
15:58
a way to do this that's that's the that's an ugly way to do it so what we're can say is co t. seon y. current
16:05
season plus one as current season right this will this is much
16:13
nicer so this is now matching right this is this query
16:21
here is now matching what we're expecting let's just run this kind of show you so you see now we have that
16:27
1996 for everybody which makes sense because it's the first uh it's the first record right and so um
16:34
everyone should be in here and they should all have 1996
16:39
so now what we need to do is we need to turn this into like a little pipeline Let's do an insert into here we're going
16:46
to say insert into players right and this this query should just
16:51
run column season stats is of type season stats array but expression is of
16:56
type integer
17:02
you what did I miss a I must have missed something College High College oh it's

17:08
because I missed
17:17
country okay so now uh now we have all the columns I think country draft year draft round
17:24
okay okay so now we can just we can just run it
17:29
I think it ran right yeah so yeah it ran because now it's it's yelling because there's a duplicate so if we say select
17:35
star from players now you'll see players now has uh a player and it has uh their
17:44
height college and then it has their season stats and then the current season right
17:50
and then this is all exactly like what we're expecting to um expecting to see
17:56
here and so this is uh but now what we can do is you
18:03
can really you're going to witness the power of accumulation and uh what we do
18:08
now is we just fiddle with these two values and I think you're going to
18:14
really see what we do so you saw how we loaded in 1996 so now what we're going to do is
18:20
we're going to change yesterday to 1996 and today to 1997 so what this is going to do is
18:28
we're going to insert into players but we're going to have the [Music]
18:34
um this is going to create now if we say like select star from players where
18:39
current season equals 1997 right so now in here you'll see we
18:46
have a bunch of players right um and you'll see now you see the you see how
18:52
this now has an array of two values that and we we pull these players forward
18:57
right and see thisy guy here this guy this 1997 guy he joined this is his
19:02
first year right and so this guy just just joined right so he's not even um
19:09
like he doesn't even have like like he's like a rookie probably or something like that right and you'll see some players

19:15
will have two records some players will have one right and that's pretty common
19:20
right and that's because we're holding on to the players there's going to be some people in here who like will have
19:26
just a 1996 record because they retired right because they played last year but
19:32
they haven't played in 1997 so but now we're holding on to more players and we're going to keep holding
19:39
on to more of History so what I want to do real quick is let's go ahead and
19:44
change this 97 to 98 right and we're just going to we're going to do this like two or three more times because I
19:51
want to illustrate uh there's like I don't know if y'all uh follow the NBA at all but there's um okay good that worked
19:59
so like you know Michael Jordan he did this weird thing where like he like retired and then like he came back and
20:05
played for the Wizards right and um and like he wasn't that good on the wizards but like um but he was like one of the
20:12
best basketball players of all time right so um I just finished loading in uh 1999 so let's do uh 1999 to 2000 so
20:20
Michael Jordan came back in 2001 so uh let's just load up like two more data
20:26
sets here so they went Chang us to 2000 2001 so now uh let's go ahead and load
20:32
this up okay so now let's go ahead and select
20:38
star select star from players um and then we're going to say where current season equals
20:45
2001 so you'll see in here how we have all sorts of players in
20:53
here and they're going to have all sorts of columns right and you'll see like this guy this h Houston he he just
20:59

joined in 2001 right some people are going to have many seasons some people are not going to have that many seasons
21:05
so well let me show you how this is really powerful though because we can say if we say m player name equals
21:11
Michael Jordan right because he
21:16
uh okay so you see here's Michael Jordan North Carolina Country draft year all
21:22
that stuff but if you look at his um you'll see how he has that Gap right so
21:27
you see like 1996 1997 and then 2001 so he had a gap right where he um he was
21:33
like gone for a couple years right and um so that can be a big kind of metric
21:40
that you can also add into this um kind of
21:46
like like you can cumulate and add more metrics into this table that like can
21:52
can kind of illustrate like how long it's been since someone retired or like years since retirement stuff like that
21:59
you don't just have to collect this array right that can be another big thing that you can add into this table
22:04
so that's one of the things that I think would be a powerful thing that you can also add
22:12
so I want to slow down here just for a second so that we can really kind of
22:17
take in how powerful this actually is because now if we joined this right so
22:24
say we uh like join this table with another table that gives us more information about Michael Jordan then
22:31
what you can do right is this actually can
22:36
easily become so this table here is now flattened out right and we it can easily
22:43
become player Seasons again this table here I'm just going to show how that
22:49

works real quick so what we're going to do is we're going to say select player name and then what we're going to put in

22:55

here is I don't know if y'all have ever used un but there's a thing called unest then

23:01

we're going to say season stats and we're going to say as season stats all right so

23:09

this okay so you see now how we have three records we have the 1996 1997 and

23:15

2001 records that will all they're all there right so this is one powerful way

23:22

um that you can get this back right I think do does dot can you put a dot star

23:27

here I I mean in post or in Presto you can do that okay maybe can you do it this

23:34

way because then can you do season stats this okay one second I want to see if

23:39

this actually works okay but okay so you have the season stats here and then this gives you the unest right I

23:47

think okay I think postgress is strange and like because what I want to do is I want to show how you can actually

23:53

explode it back out and get the get the columns here so I think what you can do is if you put this in a CTE so we're

23:59

going to say this is like

24:05

unnested okay so in this case what I want to do

24:12

is I want to say um season stats. star yeah it does work awesome okay so this

24:18

query does work missing from wait do you get okay

24:28

you okay you have to do it like I think you just have to do it this way you say season stats. assist does it

24:35

give you that it doesn't give you that what that's so weird that actually should

24:42

work one second like I know I'm really close here there's like um uh I want to just check something real quick on the

24:49

um here there's like a I have the unest query does it does it actually show here
24:54
no that this is just saying seasons though cuz one of these has it one of these has the
25:01
um the way to do this one second let's see here cuz this has all the coales
25:07
stuff is it one of these season stats oh
25:14
and then you have to like cast it so I think what you actually have to do here then is if you do this I think that will
25:21
make it work if you do if you cast it like that that will give us the um
25:28
okay there we go so postgress is strange but if you do it that way can you do a DOT star like that though because really
25:35
you should be able to do dot star like I know in um in like spark and Presto and
25:40
stuff like that you can do do star like this and it will ah it does work okay cool that's the query that's the syntax
25:47
right so this is how you get back to um the the old schema right so you can see
25:54
how like you can go back and forth easily going from uh the unest stuffff
26:00
back to the the regular stuff and that is definitely one of the powerful ways
26:06
that this query and these syntaxes can like essentially you get the
26:12
versatility of knowing like all sorts of other facts about that player but then
26:18
you also get to know like okay like um like if they do if they want to do a
26:24
join they can go back to just this players table and it's already at that grain and then now like for example let
26:31
me just remove Michael Jordan here real quick and then you'll see that this is going to always be sorted right you'll see that like every player you see how
26:39
it keeps all of them together so now all of these player names at least for that
26:44

like you know that runlength encoding problem that I was talking about you'll see how there's all these players in
26:50
here and then like all their years and they're all kept together um and so that
26:56
is the powerful thing that you get access to with cumulative table design
27:01
is now if you have this temporal component right in this case is this season
27:07
component that now can be if you have a join or anything like that you do the
27:12
join awesome get the new things that you want for player but then after that you
27:17
can unest this and then everything will stay nice and compressed because it keeps all of the temporal pieces
27:24
together and you don't have to worry about uh messing up sorting because the Sorting will all stay and you'll be able
27:31
to keep all the Sorting together so that can be a very powerful thing that um I
27:36
highly recommend that y'all kind of like look at and look into
27:41
so but like what if you want to like figure out other stuff here right I mean
27:47
there's all sorts of other things here that I think could be really uh powerful
27:53
to look at right and and this is one of the things that I like I also want to illustrate is how you can use these
28:01
analytical queries and this cumulative table design um let's go over this players table again like uh let me just
28:08
I'm going to just stomp this now so if we say select star from players uh where um current season equals 22 2001 right
28:17
what we're going to do is we're going to go ahead and we're going to uh we're going to just drop the drop drop the
28:23
players table real quick I know this is like kind of sad because we just loaded up all this data but we want to add in
28:30

two more columns into this table so let's go ahead and uncomment this and
28:35
then so we want to create um essentially
28:40
um what's called the scoring class column uh which is like essentially like is this a good player a bad player and
28:47
we're going to base it all on the points column but like what we want to do is we're going to say create type um
28:53
scoring class right this going to be as enum and then enum values here I'm going
29:00
to say star good average and we'll say bad go with that so this enumerated type
29:08
is going to be all of and we will come up with a definition here of like what each one of these things mean we have
29:15
scoring class but we want to put scoring class in here as well so let's put scoring class into this table and then
29:22
we also want like years since last season CU I think that's another great
29:27
um a great uh metric that we can also compute um so this is going to be our
29:34
new players table and we're going to add two more columns into that players table now we have our players table but we
29:40
need to add two more columns in it right so we have um and then we have a current
29:46
season is still the last column so it's after the season stats we're going to put two more columns in here so so in
29:52
this case we're going to say case when um what's it going to be t. season is
29:57
not null then we have the case this that means they were active this season then um
30:04
we're going to put then uh we'll put something right we'll put our values here and then oh wait it's going to be a
30:10
cols my bad this is going to cols put a cols here um and
30:16

then uh wait no no that that was right because we need the case one there okay so we have an end well we'll come back
30:24
to this in just a second then we're just going to put like a that and then we have uh the last one let's do the years
30:30
since last season first right so in this case we're going to say um case when t.
30:37
season is not null um then zero else
30:42
this is going to be um coals y do um
30:47
this is y. years since last season comma 0er plus one and this is end as years
30:55
since last season so I'll go over this real quick so and then
31:01
we'll go we'll do the scoring class here in just a second so how this works right is if the season
31:08
is not null then it's zero easy right um
31:14
otherwise uh they um the years since last season this might be null uh
31:21
actually wait it it should never be null right because it should because the
31:26
first year that they're in it should be there and then it will always be populated after that so yeah actually
31:32
this Co this is not necessary you can just put the plus one and that will work so how this works right is like okay if
31:39
the current season is not null that means they're they they played this season and then otherwise it's been um
31:47
one more year since they've played so this might and as as they've retired
31:52
this number will keep incrementing and incrementing and then um if they come back though it will go back to zero and
31:58
so that's how this like cumulation works right
32:03
so um in this case uh for the um scoring class we essentially want to uh we have
32:11

three values here we have um star good average and bad I like to use uh points
32:17
for that right so then we can say then case when uh T do points is greater than
32:23
20 then um star um when T do points is greater than
32:30
15 and good when T do points is greater than 10
32:36
then average uh else bad end okay there we go so now this is
32:45
going to give us and then we want to cast this as um a scoring class and
32:52
uh that's going to be the first case is like if it's uh um if the season is not
32:59
null but then there's going to be an else here right and in this case what we're going to do is we're just going to
33:04
pull in last year's scoring class so like
33:10
whatever their most recent season if they're retired the scoring class will just be pulled in from their most recent
33:15
season and that's going to be how this kind of scoring class works we going to
33:21
put this as scoring class so this is an enumeration right that's what's up here you can only have these values that's
33:28
the whole idea behind how enumerations work right so let's go ahead and uh got
33:34
to change these back right so yesterday this is 1995 and then this is 1996 so we
33:40
have our now we can run this right let me uh I need to comment this out so we can just hit the hit the Run button cool
33:48
so now this will run okay so that ran and then we want to
33:54
change this to 96 to 97 run
33:59
right then 97 98
34:05
run 98 99 just going to do a couple here just to kind of like show how this like
34:12

99 2000 and we have like one more after this we have 2000 and
34:21
2001 okay so now let's go ahead and say select star from players where current
34:28
season equals 2001 so this will give us so you'll see now we get all these
34:36
columns we have the draft year we have season stats and then we have like scoring class and then years since last
34:42
season but you'll see that like most of these people are going to be okay so see
34:47
here we go see there are people here like so this guy he was in 97 but like he hasn't played in four years right so
34:55
there's going to be a lot of people in here who like have like one or two years in the data set and then that's all they
35:01
have right so anyways uh you'll see here we have our um current season and then if we say like
35:07
and um we's say player name equals Michael Jordan right so you'll see in this case let's just filter down to this
35:14
guy right but you'll see years since last season makes sense right and you'll see how he's a star because in um 2001
35:22
you see he got 22 he averaged 22.9 points so makes sense that he's a star
35:27
right so but if we go back one right so we say well instead of let's go to 2000
35:33
you see then it has years since last season 3 so like because like he hasn't played for three years and he took three
35:38
years off and so like you'll see that like you can build these things up incrementally and you can really have
35:44
some really powerful ways of doing cool queries so one of the things I wanted to
35:50
show here though is let's um change this back to 2001 what I want to show
35:56
is let's do some analytics to see which player
36:03
had the has had had the biggest improvement from their first season to

36:08
the the most re to their most recent season so what we can do right is we can
36:14
say season stats you know at one is going to be this is as first season and
36:21
then we have um season stats at
36:26
cardinality of season seon stats and this is as latest
36:32
season so and then we can put player name in here so
36:38
this you'll see now this gives everyone and you have both their first season and
36:43
their latest season so if we say this case oh yeah we got to do the season stats we do do or then you have to put
36:50
it in pen because it's obnoxious then you see doop points and put this in pens
36:56
and then cast it season stats dot points so now this is going to give us
37:02
uh okay there we go so now we have our first season points and our latest season points so what we can do right is
37:10
we can actually just divide here right or or we want latest season on top
37:15
though right um and then we want to divide so that we can see like how much
37:21
people improved I think this might give us divide by zero because of oh yeah we
37:26
get division by Z so what we want to do is um if
37:33
um equals 0 um one else oh no just then you just
37:41
put like put this so now that will like we can just
37:48
have that if block there right does it not like do I did I mess
37:54
something up that oh is no that that looks right okay fine we'll just let's do
37:59
it we'll do it the case when
38:07
way there now that will work okay awesome so now we have uh you'll see we

38:14
have we we we have a um an order here right this is going to
38:21
give us our um our latest our current SE our first season and our latest season
38:26
so if we say order by two descending this will give us who was the most improved player right so see a lot of
38:34
wow a lot of people have exactly the same so if we go all the way up okay so
38:39
it looks like we have Don mle he like he did a 13x so like between his first
38:46
season and his latest season he did 13 times better right and so that can be a
38:53
very powerful but you see one of the things that I I want to really emphasize with y'all here
38:59
is this query doesn't this query doesn't have a group by so there's no Group by
39:07
but it seems like you like normally you would have a group by here because you would have to have both of those and you need to subtract them or do an
39:13
aggregation of some kind where you do like a Min and a Max and then you Group by right that's how you would do this
39:19
right but this query has nothing right no Group by so this query is actually insanely fast right it's a very very
39:26
very fast query order by part is the slowest part of this query but without the order bu this is like you can do
39:31
this query with no Shuffle very very fast so like that's one of the things
39:36
I'm trying to really emphasize to y'all is this query is like or this cumulative
39:44
pattern that you can use is very powerful it incrementally builds up
39:49
history it gives you access to historical analysis right like and for
39:54
example like if we say we just put in one thing in here let's put n scoring class equals star so that we only look at like cuz there might be players who
40:00
like uh like okay there we go so these are okay

40:06
these names are looking a little bit more common so like so Tracy McGrady Dirk nitzky right oh there's Kobe Bryant so he did like um 60% better and then
40:14
like you see all sorts of players there where you can kind of filter it down to like the current players and you can these names all are like looking very
40:21
familiar now but like that's the idea here right is you can easily do do
40:27
historical analyses on cumulative tables and you don't have to um do any Shuffle
40:34
any group buys and this this query like spark or Presto will destroy this query in half a second right that can and
40:42
that's like it it will just be so fast because the thing is is like if you don't have a group buy and everything
40:47
can happen just in the map step and there's no reduce step then uh this can
40:53
this can be as paralyzable as you want it to be like this can be like what I
40:58
call infinitely paralyzable we're like so this can this query can be as fast as you want it to be essentially right and
41:04
so that is one of the things that can be really really powerful in creating these queries um I I hope y'all um had um a
41:12
good lecture here today

English (auto-generated)

All

From the series

From Data with Zach

Computer programming

Presentations

Related

For you

Recently uploaded