Dimensional Data Modeling

Dimensional Data Modeling Day 2 Lab
Data Modeling - Building Slowly Changing Dimensions (SCDs) - Day 2 Lab

Transcript:

0:00
[Music] welcome to dimensional data modeling day
0:07
two lab today is going to be a Hands-On exercises that builds on the day one lab where we built a cumulative table that
0:13
tracks all of the NBA players seasons and their stats throughout the seasons
0:19
this time around we are going to be taking those data sets and converting them into slowly changing Dimension type
0:24
two because remember slowly changing Dimension type two is the gold standard of all the slowly changing Dimension
0:30
types it's the only one that is purely item potent so we're going to be using postgress in today's lab if you don't
0:36
have the stuff installed make sure to clone the repo in the description below you're going to need Docker installed
0:43
and you're going to need to have that set up before the lab to get the most out of this because this is Hands-On
0:48
exercises and we learn data engineering by doing here and yeah if you want to learn how to do this type of modeling
0:54
with the hottest Technologies like Iceberg and trino I highly recommend checking out joining the data expert
1:00
Academy in the description below first we need to like probably drop and recreate the players table I made a
1:06
mistake in the last lab where like I didn't add the columns that I wanted for this lab and I forgot that these Labs like build on top of each other so
1:13

anyways I'll just go over this real quickly to kind of go over what what I needed to add so we needed this is
1:19
active column in the player table which just says whether or not the player is active in the current season we need
1:26
that and then also um uh we want to all this this query does is it like does
1:32
this crazy thing where it generates a series of all the seasons and then it creates that same uh like uh array table
1:42
that we had from the previous session what we're going to do is I already ran all that so what we're
1:50
going to do is we're going to uh we're going to create a table we're
1:56
going to create a table create table we're call this uh we call this players CD so the main things that I I I care
2:03
about two dimensions in that other table right so let's let's go ahead and select some things from this table real quick
2:08
scoring class and is active from players where current season equals
2:14
2022 so you'll see in this table we have uh a scoring class I's put player name
2:20
in here and you'll see with this we have access to the player name the scoring
2:27
class and whether or not they were active that season and so what we want to do is we essentially want to
2:35
create um an SCD table that models like from
2:42
what year to what year they were either like we want to Mo like we want to see
2:47
we want to track changes in two columns because that's one of the things that's cool about SCD tables is that like you
2:52
can actually track multiple columns at once and that's what we're going to be doing uh today so what we want to do
3:00

right is let's create our players SCD table uh what we're going to have in here is we're going to have a player
3:06
name and this is going to be a text and then we're going to have a scoring class
3:11
uh which is a scoring class we have is active which is a Boolean and then we
3:17
have a current season uh which is an integer and then our um primary key here
3:23
is going to be player name uh current season it's going to be our uh oh yeah
3:30
we have a couple more things here though right so this is because we're trying to do a type two so we need to add two more
3:38
um columns here right which is going to be start season which I call an integer
3:44
and End season which is an integer so you're going to have both the start season and the End season as well so
3:52
this is now essentially a properly modeled SCD right we have our two these
3:58
are our columns we're tracking and then this is going to be our current season this is usually going to be the
4:03
last column like in uh like the big data world this current season you could
4:09
think of as like the date partition of the of the data table so you can kind of like think about it that way and then
4:15
you have the start season and End season and then these are the two columns we want to track so one of the things I
4:21
want to show here is like we want to see how Michael Jordan kind of like cuz he like retired and then he came back right
4:27
and we want to like see how his records change over time and we're we're going
4:32
to do this first what we're going to do is we're going to write the query that looks at all of history and creates one
4:39
SCD record from all of history that's what we're going to do first and then
4:44

after that we will I'll show how to then take this query uh take a an existing
4:51
SCD and then build on top of it incrementally because you can do it either way so let's go ahead and uh run
5:00
this code here run this create table it's already exists great okay
5:06
so here's a good question like how are we going to do this like how do we create um like so players like if you
5:14
ran that other big nasty query I sent you like players is going to have data all the way back to 1996 as well right
5:20
you'll see we have all sorts of data in this players table so how do we create a
5:26
players table with no filter here so what we want to do right is we have a
5:32
players scoring class is active what we want to do is we want to calculate the
5:40
streak of like how many seasons or how long that they were in this uh like in a
5:48
current Dimension and how we do that is we look
5:53
at like what was the dimension before that's how we can see how this works so
6:00
and how do we do that like well we use window functions right so let's go ahead and look at lag right scoring class
6:08
comma one um we need to uh partition this though right so we're going to say
6:13
we got to put over we're say Partition by uh and then in our partition here we want a partition by player name and then
6:20
we have an order by our order by here is going to be current season so we can see
6:25
who like and this is going to be as previous score ing class right and let's
6:33
put is active up here just to keep things going so then we want to just
6:39
essentially copy this again for is active and then this will be previous is
6:44
active so this will be depending on what the previous season was that will be

6:49
what we get out of this so if we run this you'll see uh oh we need current season in here as well let's put current
6:55
season here so you'll see okay like here is AC Green
7:03
1996 he was bad and um is active is true
7:09
right his previous um you'll see his previous scoring class and previous is active these are both
7:14
null um so like you can see that this like when it's a line like this that means null if it's empty that means
7:21
false so um that makes sense because he didn't play in 1995 or there's no data
7:27
for 1995 so that makes sense but you'll see how you can go forward and forward and forward here and like he essentially
7:35
was just bad the whole time right and so you can see like other players like they're like they're pretty consistent
7:40
right but this might not always be the case right where uh so what we're going to do is we're going to create a CTE
7:48
here we're going to call this um we call this with
7:54
previous all right and then say select star from with previous so then in this
8:00
case what we can do is we can create an indicator of whether or not it
8:06
changed so what we can say is like case when scoring class does not equal
8:14
previous scoring class then one else zero end this is um we can
8:21
say put a then and an else and an end we're going to call this we're going to
8:27
call this as and this is going to be change uh well score and class change
8:36
indicator and we need to have essentially a similar similar bucket here where we have K when is active uh
8:44
does not equal previous is active we can say is

8:50
active change indicator so let's let's go ahead and just run all of this and I
8:56
think this will make a little bit more sense so we have our um players here and
9:04
see how we have these change indicators if we sort here do we have okay so see here's here's someone um here
9:12
uh he was um so Aaron Brooks you see how like he went from bad to average and
9:19
then he went from average to good wow so it looks like he was really ramping up here from 2008 to 2009 right so it looks
9:26
like he changed two times like in like back back to back Seasons right and then oh yeah and then but then then he went
9:32
then he kind of faltered right so you see how he went from good to average so these are the kind of changes that we're
9:38
looking for right we're trying to see like how um um how people like are
9:44
changing and like how they are going about their changes so this is this is
9:50
great so what we want to do here right is we we essentially want
9:56
to create uh um a streak to to identify when like for every change that they
10:04
have so imagine the first the first time they enter they get zero and then the
10:10
the next time they make a change they get one the next time they make a change they get two and so on and so forth so
10:16
this is um we're going to put this like we're going to say like make a new um CTE here with
10:23
indicators so this is now this is great so
10:29
what we want to do is there's two ways to go about doing this right is um like
10:35
really we want to track these things together right and I think that that's probably something we want to do here um
10:42

like these two like it gets complicated if we have two indicators like this in this pattern so what we actually want to
10:48
do is combine these into um into one single indicator and that's pretty easy
10:56
so what we're going to do is we're going to paste this up here this case when and then we're going to just call this
11:02
instead of scoring class change indicator we're going to call this change indicator which is where if
11:07
there's a difference in scoring class or is active so like if they go from good to bad or bad to good or from active to
11:14
inactive then this will indicate that they they did that right so now what we
11:21
want to do is we want to create um let's let's look at this so we say with
11:26
indicators so what we can do here is we want to
11:34
essentially sum these up so that we can keep track of the streaks for players
11:39
right so in this case what we can say is you can say um you can say row number no
11:46
we want sum my bad sum uh in this case we're going to say uh change
11:51
indicator uh but this is actually going to be uh another window function in this case we're going to say over and then
11:58
again we're going to say uh partition bu and then in this case our partition bu here is still going to be a player
12:06
name and then our order bu here is going to be uh current
12:15
season and then this is going to be as streak identifier which is probably like what
12:21
is Zach doing here right so let's go ahead and see what's going on here with a a certain player right so here's AC
12:28
Green who like seems like he was active and bad for like a long time right so
12:34

let's go look at streak identifier you see how streak identifier okay so he did change here though so you see um he went
12:41
he became inactive at this point so you see how his identifier changed to one
12:47
and then it looks like he was just inactive for like the rest of the data set so um and you see how like that's
12:54
all in the same that's all the same value here right so you can see exactly
13:00
when he became inactive so this is exactly what we're looking for um when
13:06
we are trying to build out these scds is we are looking for like how long they stay the same value and you can kind of
13:14
see how different players kind of progress and change their values over time and uh some players are only active
13:21
for like one year so what we can do is from this uh data right now we have one
13:28
more window function here so we're going to call this um uh with streaks as and
13:34
this is going to give us our streak identifier so now what we want to do is
13:40
we essentially can aggregate on this identifier right because if we do a Min
13:46
and a Max on this identifier for the season that will collapse all of these
13:52
rows right because we uh we know that all those values are going to be the
13:57
same because we just prove that out with uh how we are doing this with the lag and the lead and all that stuff so in
14:05
that case right what we can say is um we can say select uh we're going to say player name and then we're going to say
14:12
um Street or we're going to say from with streaks so in this case we want to do a
14:20
player name and then we can say streak identifier and we want to do a couple we
14:25
probably have a couple more here we have is active and then we have um uh scoring class and then then it

14:33
gets a little bit interesting because then we say Max we can say Max current season as um End season and Min current
14:44
season as start season oh well let's put this you want usually you put the start
14:49
before the um before the end there right and then
14:56
this is now uh this is a regular aggregate ation right so in this case we can say uh Group by player name streak
15:04
identifier is active and scoring class right so now this will collapse all
15:11
those records right so now you'll see we have um these players who are in here
15:19
and they are like some of these are for a long time see this 2003 to
15:26
2022 this guy right and then um some of these people change a lot right like
15:31
this guy changed 10 times wow like um so what this is doing right is you can see
15:38
with these people you can uh like a lot of times like when people do this pattern they actually don't you Group by
15:45
streak identifier but you don't include it in the data set right and then what this does is let's put in order by in
15:52
this to kind of really make this pop so we can say order by player name I think that will really uh make this pop a
15:59
little bit so remember we had that AC green guy right where he had just that
16:04
one change where he went from active to inactive right so you'll see here how he
16:09
was active from 1996 to 2000 and then he changed uh in 2001 to
16:18
inactive right and then from 2001 to 2022 he was inactive
16:24
so what I'm trying to illustrate here is you see how
16:29
this is essentially it like we essentially have what we're looking for now this is our uh our SD sort of table

16:38
um one of the things that I think we should do uh for the later part of this
16:45
lab is we want to filter this down just a little bit and what I'm going to say
16:51
here is I'm going to say where current season is less than or equal to 2021 just so that we can use 2022 in the
16:59
incremental build of this so what we can do is uh let's let's run this code again
17:05
and you'll see uh now now this goes to 2021 exactly what we are expecting and
17:12
um you'll see okay here's AC Green with the ordering and so so this is so zero is their first streak and one is their
17:18
second streak and so you'll see like okay this this makes sense let's find someone who like is a little bit more
17:24
interesting like U this Aaron Brooks guy right um it's actually like put streak
17:29
identifier in there as well like the order by so it's not so freaking ugly so you'll see this now actually explains
17:36
the entire history for people so like this guy started out as bad and active then average and active and then good
17:43
and active like he just like was like one year at a time bang bang bang bang bang right well he changed a lot right
17:49
like this guy okay then he did have he did have a um it looks like he went in active for a year here right and then uh
17:56
then here he finally had two years where he was at least consistent this guy like wouldn't even say he's like slowly
18:02
changing right he's changing almost every season so like but it depends on who the player is right and like so the
18:07
streak identifier is just like the first value and then each one after that is um every time it's changed and so that you
18:14

can get the continuous values that you're looking for for different for different players and like over what
18:19
time frame that it exists so um that's kind of the idea behind how uh these
18:25
this streak identifier works so what we can do here is the last thing we
18:32
have in here is this current season which actually we want to just hard code
18:37
this is going to be 2021 as current season because this is uh this is like imagine if this was in like airf flow or
18:43
like some sort of pipeline that this is a a parameter that you would inject
18:49
right this guy this guy so um this essentially is the query that we're
18:54
looking for um I think we say insert into players
19:01
CD um oh what did I did I did I mess
19:06
something up here there's like a there's like a what are the columns in this okay
19:15
so oh ah you're right streak identifier is not in there and it's like score okay scoring classes
19:22
first and then is active and then
19:29
start season end season current season okay that looks right okay so now this is our kind of
19:37
way of so the main thing I'm trying to show with this is this is a way of
19:42
processing all of history and oh it already
19:48
exists see players CD
20:00
what oh oh because there's a duplicate in the primary key because the primary oh I messed up I actually messed up the
20:07
primary key is not that it's um the primary key is all three it's the start season end season and the player name
20:13
it's all three of those small little mistake here so drop table players
20:20

SCD and then um in this case primary key here is going to be player uh name start
20:27
season I think it's actually just player name and start season cuz yeah you don't have to put End season in there like you
20:33
could potentially put it in there but like they they should only have one value at the start of each season or the
20:39
like for that value so I'm pretty sure that's actually the primary key not current season it's start season so go
20:46
ahead and create that table one more time and now this should actually run okay now it runs pretty great so now we
20:52
can just say like uh select star from players SCD and this should okay great this
20:58
gives us the same data right exact same data that we had before and so um this
21:04
is um awesome so this is essentially how you can create an SCD table uh
21:12
from uh your whatever data you have like your daily data or like because you can
21:17
think of the Season here you could change that to date or year or month or whatever and like you can have the same
21:23
exact values right that could be a big thing that can make a big difference here um so that's a big part of like SCD
21:30
stuff that I highly recommend like checking out is just like this is um this could be very powerful
21:37
uh there are some things about this uh pattern that I don't like uh and let's
21:43
um let's kind of go over those before we go into the incremental build just so that you can kind of see the difference
21:49
between the two so the there's going to be some expensive parts of this query um
21:56
the the big expensive parts of this query are these windows right like look how many times we have to slice this up
22:02
right so we have one two right and then we have three so you essentially have to do two two window functions on top of

22:09
each other and then an aggregation right and this is the only time we aggregate so you actually do
22:15
window functions on the entire data set right like you never crunch the data down anywhere right it's only at the
22:22
very end when you do this big group buy that you end up actually shrinking the data volume at all and that is it's good
22:29
because like you have all the data and like you can you you have the full history which is very powerful now you
22:35
have this beautiful little SCD table that like really explains and really like really makes things pop right
22:42
really shows how people like play and do their thing but like it um this this
22:49
only works like this query really only works a lot like given the scale but uh
22:55
this query like also is powerful so like when I work on unit economics at Airbnb
23:00
this essentially was the query we worked with and uh we did this for all the different line items and all the different uh uh things in that table and
23:08
it worked great spark was able to Crunch this pretty well because it's um like it
23:14
it like these window functions are actually not that expensive like you if you just partition things up and like
23:19
slice and dice it so like and then you can essentially just regenerate all of history every day right so then you can
23:25
change this like uh like the next day right so like you would just change it to like 2022 2023 2024 right and then
23:32
then you don't even have to think about the incremental right even though like the data engineer and me like who cares
23:39
a lot about efficiency is like I hate this why am I scanning all of history every day I don't like this at all and
23:46

that's a big thing that I think y'all should definitely be looking into is like how that is um impacting stuff is

23:54

uh like and mo but most of the time like this can be enough and this this this is going to work for you um but on the flip

24:02

side I also love teaching about the other ways that things can be done because this query here versus the query

24:08

we're about to do this query here is going to be a lot more prone to out of memory exceptions and skew and other

24:15

interesting problems like that that like um like for example if you have like a

24:21

like most of this is a slowly changing Dimension but you saw how we had that one guy who was like changing every

24:27

single year so imagine if you have some people or some dimensions in your data

24:33

set that are not um that are not as

24:38

slowly changing as other ones then the problem here is that you get a lot of streaks for that specific user and then

24:46

that blows up the cardinality of this right and that kind of like like slows it down and that's where like when

24:52

you're doing this like they should all have roughly the same amount of slowly changing but obviously that never be the

24:58

case but that's what you're kind of hoping for and but like that's where I like these queries I like this query I

25:05

think this query works great I've ran this query in production at scale so like don't worry too much about that cuz

25:12

that's one of the other things I want to like really uh advise y'all with is when you're working with Dimensions you can

25:19

do like crazy stuff like this like you can do like I'm just going to scan all the history like you can do that because

25:25

dimensional data really isn't that big like dimensional data does even even at Facebook when it's like billions it
25:31
still isn't that big compared to like where the fact data goes right the fact data is just like can be billions and
25:37
trillions and just really really really large So like um you know that was one
25:42
big thing that was different between Airbnb and Facebook though is that this query probably wouldn't fly at Facebook because they have billions of users
25:49
whereas at Airbnb there's like millions of users and so you have like a kind of a there is a line there of scale when
25:56
you add like one or two more zeros where you can just like throw everything into a window function like this and call it
26:01
a day so that is definitely something that you want to be considering as you kind of like are going through your
26:06
career and going through all sorts of stuff so um yeah let's go ahead and get into this next uh kind of way of doing
26:14
things right so what we want to do is so we have players SCD so what we're going
26:19
to say is we're going to say with um yesterday data as in this case we're going to say
26:27
we're going say I call this last season last season SCD uh select star from players SCD
26:35
where current season equals 2021 and then we have um this season
26:44
data as and then in this case we have select star from players where current season goal
26:52
2022 so what we're trying to do here is see if things have changed
26:58
right so the thing is is like some of the records we don't have to about and we know that they're never going to
27:05
change because they like like they're already completed they don't have that current record to them so the thing is
27:12
is like so this last season SCD what we want to do is we have one more thing in

27:17
here and that's going to be n Season equals 2021 because then we want um then we have
27:23
historical SCD as and then historical SCD is going to be this
27:32
guy but this is where End season is less than 2021 so because in this case like
27:38
if someone has already like for example Michael Jordan he like retired in 1997 and then he came back in 2001 those
27:46
records are never going to change even if something happens in 2023 there's nothing that can change those records
27:51
because they have already started and ended so that's what this historical SCD is doing and then we have last season
27:57
SCD so let's go ahead and just like look at that select star from last season SD first just to kind of like get a grapple
28:04
of what's going on here and you'll see that like we're going to only have one record for everyone here right you have
28:09
this 2021 record and like uh it's going to be uh where all of these are going to
28:15
be like these are the current records like assuming that we have all the data up to 2021 and then uh like 20122 could
28:23
um for for the records that don't change they could just increase one and then
28:28
for the records that do change we need to add a new record that's essentially how this is going to work right so how
28:36
are we going to like look at this I think that there's going to be a couple different ways that you could think about doing this right so one is let's
28:45
look at uh like so if we join here right so let's say from last season STD let's
28:51
call this LS um join this season data TS
28:58
all right um this is actually going to be a left join uh because uh there might
29:03
be uh actually yeah yeah it'll be a left join right because oh wait it's the

29:09
other way around because this season can have new data because you can have the new players who didn't come in right
29:15
it's actually this way last season SCD and this is actually just uh yeah and then this is a left join because you can
29:21
have those new players who don't have any records yet and then this is going to be on ls. player name equals ts.
29:29
player name so this is going to give us okay so in this case what we care
29:36
about here is we want ls. player name and then we want um um LS do let's change this this is TS
29:45
and this is LS there we go this is so this is ts. player name and then we have
29:50
um LS uh do um scoring class ls.
29:56
isactive and and then we have ts. scoring class ts. isactive right so
30:02
these are going to be uh the main things we're going to be looking at so you'll see in this case like these players are
30:09
mostly going to be the same but some of these are not right so like this Aaron Henry guy you see how he is not the same
30:17
his record changed um so in this case we have um like we can have the changed
30:24
records and then we have the new records right so that's like that's going to be
30:29
essentially the way to think about this right is we have the changed records and the new records where so in this case
30:35
let's go ahead and add in here we have this is going to be let's call this um
30:41
uh unchanged records call that
30:47
first okay so in this case uh what we want here is there's a
30:52
nice wear condition here and the so there's going to be two things here like first off this is this this is not a
30:58

left join now this is just a regular join because we'll have the new we'll have the new and changed records we'll
31:03
have new and changed and then unchanged records and then those will be our kind of all of our things going together so
31:09
in this case uh we just have a wear Clause here where it's like we ts. scoring class equals um ls. scoring
31:16
class and ts. isactive equals ls. isactive right and then in this case
31:23
what we have is we have um we want to we want to essentially keep all of the the
31:29
records that we need here so we we have ts. player name and then we have uh ts.
31:35
scoring class ts. is active and then we want to have uh then we have the ls.
31:42
start season and then in this case we have ts. current season and this is
31:47
going to be asend season so what this is going to do is for the records that
31:52
don't change we're just going to increase it by one
31:58
so uh like let's go ahead and just look at that we're going to say select star from unchanged records so this is going
32:04
to essentially expand to out one so you'll see uh we have like uh AC AC
32:10
Green right and see now he's going to go from 2021 to 2022 and so that's great
32:16
like he's going to be uh like going to be in there and that will be awesome so then we have uh we have one more in here
32:25
where we have the we have the new and change records that's going to be the other kind of uh the other pieces of
32:32
this puzzle that are going to be kind of an interesting piece right where so um we have unchanged records and then
32:40
we have uh new records right or new this is new and changed records right because
32:47
those can go together so in the changed records category the problem is is like

32:52
you actually end up having two records right you have the record for um uh the change and then you also have the record
32:59
for the um the like the the closed record the one that like just finished and then you have the record for the
33:05
change so in this case what we want to do is we want this in here right and
33:12
then this becomes a left join here and then
33:17
uh this this wear Clause is interesting right so it's going to be um or so I'll
33:25
show you how this works this one's so you have have this as an uh it's like
33:30
like they either one of them doesn't equal either the scoring classes don't equal or the is actives don't equal but
33:38
then there's another one that's like or um ls. isactive is null right or we can
33:45
just say ls. player name is null and that will give us the um the new record
33:51
the new and changed records that we can then work with so in this case right um
33:58
one of the things that we can look at right is uh there's this one is a little
34:03
bit trickier so the you have the you you you you really want to use like an array
34:09
of struct here kind of thing right because you have to essentially take one record and make it two records because
34:17
there there was a change that happened and so you have to have both records in in in there uh I I hope that makes sense
34:24
so um in that case like what we want to do is we want to create a row right in
34:30
this case which is going to be um wait is this going to be I think I made a
34:36
mistake here I think this is just changed because then new new is like the last one you have like one more that gives you new which is just going to be
34:43

where uh where player is null and then you just pull it in yeah okay so actually this is just changed records so
34:50
and all these go together I know this is like kind of a lot but it'll it'll make sense so in this case we're going to put
34:56
an array here where we need to unest this array essentially where what we have is we
35:03
have a row here which is going to be these like well because you have the
35:09
um the old record and then the new record they kind of like go together um
35:14
and I think we actually have to create a type here cuz uh postgress is weird in
35:19
in spark and other places you don't have to worry about this cuz spark can just handle it but we need to have one that
35:24
has uh these values it has like the SD type so I think we'll go ahead and do
35:30
that up here so create type SD type and then this type has uh we have scoring
35:37
class scoring class and then we have is active
35:43
Boolean and we have start season integer and End
35:48
season integer I think that's all we need is that and then because then yeah
35:53
the then the player name is going to still be here because we join on player name so that will be fine so we don't
36:00
need player name and then so we'll create this type here okay so now in the row here right
36:09
we need to put these in here so we have the the old record which is just going
36:15
to be all of the old record stuff so we have ls. scoring class ls. is active ls.
36:20
start season ls. n season and then then we have uh and we
36:26
need to cast this as an SCD type then we have another row here same same idea but
36:33
these are now all we have TS scoring class TS so um these don't exist yet
36:41

right because they are um uh this is going to just be current season and this is TS Uh current season
36:49
right because the new record is just one value it's just like the current season
36:54
2022 and then so this is an SCD type as well and so this is going to give us our kind
37:01
of values that we're looking for and the good news is here we can get rid of all this right because we uh this is not
37:08
what we need and then what you can do in this is like we can do like an
37:15
unest so this is going to give us an unest which then we have as uh this will
37:21
be let's like look at like what this this looks like because this is going to be a lot of changes here so let's go a
37:27
ahead and look at like what this query actually gives us so I'll comment this out and then we can run
37:33
this okay so you'll see we have our changed records and uh it now has um is
37:42
that right because we have changeed records where yeah then this is not there so
37:48
this is going to be our okay there we go so now we have our
37:54
changed records and it has like um
37:59
there should be two what oh yeah because it's not like oh yeah there we go so you see how now we have two records we have
38:06
the old record and then the new record for things that changed out so what we
38:11
can do is we can actually uh um we probably want to just flatten like or we want to like get
38:19
get it out of the type out of the struct so we can say like um unnested changed records
38:28
and then in here we can just say select star from change records let me change it this as um records let's call this as
38:36
records we have a player name and then we have a records and then this is an
38:41

SCD type though so now what we can do is we should be able to do in this case we
38:48
can say start season so we can just essentially flatten this guy out and we can say um scoring class
38:59
there we go scoring class and then this is uh have is
39:07
active then we have start season start season and then end
39:14
season why is it isn't that what it yeah start season
39:20
end season that is like okay there we go it's just
39:26
like okay so now we have this is our unnested change records and I know that this has like been a long query in the
39:32
making right now but you'll see now we have this broken out to where these are
39:38
all the players who had um any sort of uh any sort of change here right that's
39:45
going to be our last like so you see how now we have two records for all these these are all the players who changed so
39:51
we have one last one here which is our new records right new records is easy though like so I'm just going to uh what
39:58
we want to do here is we're going to say select star from uh this season data TS
40:05
we're going to say left join uh last season SCD on ts. player name equals
40:12
I'll put this LS ls. player name and this is going to be where ls. player name is
40:21
null and then what we want here is for TS we want to say ts. player name TS do
40:28
scoring class ts. is active then ts.
40:33
current season as uh it's going to be start season ts. current season as End season
40:42
okay so now the now we have everything ready to go we just have to now just essentially Union all of these together
40:48

so we have uh so we're going to say in this case we're going to say select star from historical SCD Union all and then
40:56
we have select star from um unnested change record Union all select star
41:03
from uh new records I think Union all must have the same
41:10
number of queries okay what what same number of columns okay what did we miss here 1 two 3 4
41:16
five and this has 1 two 3 4 five and then oh there's yeah we have
41:23
unnested change records new records and then historical SCD oh because historical SCD has um that current
41:30
season in it that's the problem it has that like last value right so what we can do is um but we don't actually ever
41:37
use current season in that table so what we can do here is we can just pop in
41:43
player name um scoring class is active
41:48
start season end season so now this should
41:54
work there we go okay so now we have uh uh this is the way to do it
42:00
incrementally so now like the good thing I know this query probably looks really insane but it's actually um it it's
42:07
processing a lot less data because we're only processing um the compacted uh 2021
42:13
records and then the 2022 records so this is probably processing like about 20 times less data so even though it's
42:21
like the query is a little bit more complicated because you have to think about all the different ways that things can change it's also a very powerful
42:28
query where you can um where everything kind of comes together and it like is pretty pretty pretty freaking cool how
42:36
this stuff like ends up working so um I think let me let me run this one more
42:41

time okay so we have this and then it has all the all the players and what they're doing and so this is the idea
42:48
behind how you can kind of incrementally add more and more uh players in here and
42:55
that's kind of the idea so this is like I think I did it in a slightly different way in the um in the last lab but like I
43:03
like this way of doing it cuz it's very clear of like the um the new the way
43:09
that this works oh actually there's one more Union here actually right there's the uh um select star from there's the
43:15
unchanged records right then Union all yeah that's that's actually what it
43:21
is because you get you get all of this there we go then that like is the
43:27
records that we're looking for yeah and so um this is like a lot
43:34
right and this is one of the reasons why like when I was when I was working at Airbnb and looking at unit economics and
43:39
everything and I'm like dude I don't know about that I don't know about writing this query right this query is freaking gnarly and like there's so many
43:45
like edges that can happen here and different things that like you want to be aware of right so like some some like
43:52
like for example there's going to be a couple other assumptions here that you want to really um think about right for
43:58
these queries to make them work the right way and uh like where these
44:03
problems can happen a lot where one of my big assumptions in this query that like we didn't even really talk about is
44:09
the fact that I assume that is active and scoring class are essentially never null right because if they're if they
44:16
ever get null or nullified or whatever then it kind of breaks this pattern it breaks a lot of this because like then
44:22

like uh for example null right here right uh here's one null you know null
44:29
doesn't equal null right and like this it would it would get filtered out here right whenever you do a comparison like
44:35
that where it's like if something doesn't equal null or null doesn't equal null that's where there's like that other query there's like another one
44:41
there's like that is distinct from and that's where like you can actually handle null in these cases but like I
44:48
never use those I always end up using like the hard equal signs and does not equal signs so but like that's a big
44:54
thing to remember about your query patterns is that they can be like that right you can have um you can have all
45:01
sorts of different things that can happen that like can mess things up and that's why it's important to have your
45:08
um assumptions checked and your quality assumptions checked when you're building out these queries and so in this case
45:14
this query is going to work better in in a lot of cases because it's it's quering a lot less data and so this query will
45:21
work way faster but it also has a sequential problem right where because now we are depending on yesterday's data
45:28
because you see how we have like historical and then current and that it makes it a little bit harder to like backfill and that part of it's a little
45:35
bit annoying [Music]