I can't. I can't speak to that one, but I know that one of the one of the things that has happened, it's true of 19, it may be true of 12 'cause. I haven't got that. When we run automated tests. We expect an exact result set. And we are human and fallible, and so we make mistakes, too. And one of the mistakes that the curriculum team made when making up this test was they assume a particular order. And there are two different valid ways to group things in 19. One of them gives you the right order and one doesn't. And it's a little bit like this. If you it's like the state table where there's a one to one mapping between the state abbreviation and the state. So it feels like it's perfectly fine to use either one. To group by. And in this particular case, just as a you know, spoiler alert or whatever in 19 you need to group by the genre name and not by the genre ID. Right. If you haven't gotten to 19, then that gives you huge clues about how to solve 19. Then good for you because like. I'm OK with that, alright. I'm gonna actually, out of curiosity here, I'm going to pull up 12 and see if I can. Now, I'm not gonna do that. I'll get too far a field on that. But I'll look at it later. 21. He's a is one where we were not wrong. We just were being Super Trixie, alright? We wanted to, you know, like we tend to do on our last question on the exercises. We wanted to really you know push the the envelope a little bit and make sure that it was hard to solve right. And the thing I will just tell you is that hint for those. The two things that you should keep in mind is that what happens is all about actors and which films they're in and how many films are in that sort of thing. Well. One actor might be in two different films with the same name. And there may be two different actors or more who share the same name. So you need to keep those in mind as you're solving 21. Because. So otherwise you'll bang your head against the wall trying to figure out why you're not getting the right number. France. But there is this second question which has nothing to do with exercises, which is right now you're being given these queries. You're you're, you're said, come up with a query will tell you exactly how many rows that should have. Right in the real world, nobody is going to tell you like the query is supposed to tell you that not not, you know, the you are not giving so much information and it can be very tricky to think through the real world implications of exactly how you frame your query. Right. It is one of the reasons why. Even when you're when you're creating new queries or when you're generating queries, you have to get used to exactly what they're asking, and you have to think about things like. Again, it's the whole 111 demeny idea is is really key here, right? I would actually. porsche start grade Course, start quitting, but to think of whether or not a relationship is 1 to one or not, so genre and a genre ID are one to one. A state abbreviation of state name are one to one. You're not going to have children genres with the same name. You're not gonna have two states the same. But anytime you get into things like people names. Or the names of books or movies or anything where where there's some possibility of duplication. I went up to Netflix because I was curious and and now I'm trying to. This was last cord 'cause I was looking at something else but and I found one movie that where there were four different movies from the same title. Alright, people reuse this stuff you know, especially when it's short, you know, pithy thriller names. They they like to, you know, they like to use the same one or would somebody remakes the same movie over yet again and they just you have to go by the the year of release more than you go by the the the title so. And it would just keep in mind in those instances where you have. Different possible entities? With the same natural name with like, you know, movie The title. May have different ID's and then you have to be very conscious of are you trying to get all the titles or are you trying to get? Like if you just want to list the titles, maybe you don't care and you only want one of those, even though they're four movies with the same title. But if you're trying to group it, or you're trying to count it or anything and aggregate it in any other way, you better use the ID because that is your way of making sure that you're unique. Right. Let's go for. We are going to talk today about. Share a screen so. Constraints will tell you that there is a primary key on the city. Now we're going to get into what happens when you try to violate a constraint during this lesson. The big thing about relational databases is that they don't let you violate constraints, and that's a huge benefit because otherwise you have the problem that you have in Java, which is you're responsible for all the the what we call referential integrity of the. You know where all the references point to each other. In Java you can go ahead and say. Let's say that you have a map. And has a unique key. And you have something else that is going to refer to, again taking the vending machine. Let's say that you have a map and it is. Heat off your slot number. And somebody tries to use a slot number that is not in there. Right, it's pointing to nothing, and that's fine when you're actually checking that thing. But if you were depending on it pointing to something and you get back a null and then everything you get, you know pointers. Basically you don't want no part pointers, so that is the constraint is that it is a primary key, which means it is unique. I like foreign keys. Foreign keys are a little different. Foreign keys says there is a. It will tell you the state abbreviation. Foreign key state. These names will kind of clear you in there's a foreign key for the city to the state table and it'll be state abbreviation to state abbreviation. And the. You'll see that sometimes it will say that the table is public dot state. Technically, all our cables are under public. So, well, you almost never need to refer to it in any other way. Public dot state means that's the table, which is just for your purposes, the state table. So you can find out these constraints. And by looking at the the properties of your different things. I just show you. Because because this is a little different. If we look at the park state, we remember it only has the park ID in the state abbreviation, right? They are both part of the primary key because yes. It's not Null, says I am not allowed to have a null value. That is another kind of constraint because think about it. If you do your left join in, you know for something that has doesn't have a value in there, it would normally fill it with a null. It won't allow that. The thing I wanted to show here is so the primary key is both columns together that is the primary key. It's park Ivy and state abbreviation, so together they are unique. Each one individually is not necessarily unique. Another foreign key is going to be your connection between the park ID. The park database in State Park in your state. Foreign key is going to be by state abbreviation. This will tell you that joiner thing when we talked about how when you join that is the value that that you know connects the two. You actually can connect it via other things, but you then don't know that there's a one to one relationship. When you got into other issues. Alright, all that is just preference to say we are now going to deal with our United States database. Unless you have been messing around with it, which you might have because we do have tutorials and you know. All that and it should be in a consistent state. Right now it should be where we expect it to be. If you have messed around it is OK. I'm going to go ahead and do this just to demonstrate it, and so we have it available to us. I am going to right click on my United States and I'm going to use my query tool so that I am no. I'm in my United States database and that's what I'm connecting to. I am going to go into. Just wanna make sure I'm in the right places. As much as they are all the same and I I would rather do this where I I'm doing, so I'm going to be under today. Under lecture under resources under Post SQL and there's the United States, I'm going to open it. So now it's up here. We will go into the state of this file a little bit more later, 'cause we'll understand more, but I'm gonna go ahead and run it. It is gonna drop all the tables that are already there, which means get rid of them entirely. And then it's gonna rebuild them. So this will always get me back to that stable state. Run my thing and you see that commit, and that's really all you see and took 201 milliseconds. OK, so now I have my table. If I want to refresh just to be sure. Fresh table. OK so I have my my tables here. So today we're going to say, well, what happens if we want to add things in? I'm going to start with when in. The term that you use in SQL is insert. You're going to insert data, so these are going to be new rows. Think about what we know about a neuro. It needs to have. All those values for all the columns. Defined when we when we insert it. We're not allowed to. Well, at least anyone where it doesn't allow enough. We need to. We need to have all those accounted for somehow. There are different ways to account for them, but we will. We will get there, so I'm going to leave this here. This United States sequel there so that I can get back to it easily. When I go back in to my United States, right click and do my query tool. So now I am in a scratchpad. Actually, you know what I mean? I figure out I'm gonna use the lecture I was gonna do it just in a scratchpad, but. I still I'm gonna leave my United States there so I'm still doing exactly the same thing. But now I'm gonna open my lecture rather than. So this is the make sure I'm in the right place. I am insert lecture. Alright groovy we're gonna insert. So we want to we have we're going to insert into the park table. So let's take a look at the park table for a minute so that we know what we're talking about. Part table. As a park idea park name a data stablished area and has camp. What they're saying is, let's add Disneyland to the park table. And egregious oversight. Don't have digital, so. Insert. Into that's how we say we're going to add a row is insert into or add. You can add more than one row at a time, but for right now add a row. So into park. So this is where we identify the table that we are dealing with, so we are dealing with the park table. So insert into park. And then we're going to give the columns that we're going to set. Park. Name notice I'm not putting in my spelling right either, but I'm not putting in park. The reason the park ID well it is the unique ID. It is a sequence that auto sequence one. In fact maybe. I think I can even show that from here let me just make sure I go to my properties for a moment and I look at my constraints. I think it is. Hold on, I look at my columns columns is Park ID. It doesn't tell me I I saw it before so I know that it is here somewhere. Um? Go on key check any. Notice me. I don't know where it shows it in here. Somewhere in here it makes the point again, I'm not used to PG admin as much for this kind of thing. I'll play with it later and find out somewhere in here will tell you this is an auto sequence key. And when auto sequence key means is the first one that gets added, gets a one and the second one gets a two and every time you add it will be right after the late last one in the table. If I delete a row, it's not going to replace, but it's going to add one at the end all the time. Right, so I don't need in my. In my insert into, I don't need to specify it because I wouldn't know what it was if I. You know I I wouldn't have any way of knowing. So I'm gonna skip that one 'cause it'll handle it itself alright, the date established we're going to say. When you use their format because I don't know why that's what they use. 1955 are you still defining what the columns are? First, you're right, I messed up. I'm starting to think about the the values. OK so data stablished. Thank you. Hi. Taria has camping. So this says insert into park these columns. You you don't need to include the auto sequence one. You also can choose not to use one that has either a default value or there's allowed to be null. In general, you're going to Add all the columns. I mean this is. This is your place to to set everything. In general, we're going to do it, but it's a little bit like when you're in Java and you have it constructor and you are building that one object, which is like a row. Some things could have default values you could choose not to do. This is like the constructor that includes all the class variables and you put them all in. That's the normal way that you will be adding things OK. So then you, that's your definition, right? And then you say values. And values is going to give you the values that go into each of these things. Right some people. Like to try to line it up I. The problem with that is that not the right length, so it gets really hard, like sometimes it's it's useful thing if you're going to set up a table and you want to be really clear. Beach Bayou is in this case, it's pretty clear what the add values are and they won't line up, so let's just. We're gonna say values and again we're going to put in parentheses. And there's going to be each of the values for these things, so Disneyland. And. It is stab Liszt and this is where I was starting on 'cause I got distracted seventeen 1955. Area is 2.1 square miles. It does not offer camping. So these are going to be then the the values are going to show up based on the types that they're supposed to be. This is a character bearing of whatever size it was. It's shorter than 50, which I think is how often this is assumed to be a date. So we we put it in quotes and it calculates the date out of that. This is a numeric area with only one decimal place, and this is a true false. So this. Would show it. Now let's just so that we don't. Let us say OK select. Star from park. From park. Just to see if we can fit. So if it happens to already be out alphabetical then it'll be easy to see. So if we go down here. Can we look for Disneyland? There's no Disneyland. Hi. Now we're going to do this. We're going to insert into. OK, insert 01 which basically says. I was getting that we added one thing. Right, and if we do our select again. OK, and go down in. We still don't have Disneyland. Why don't we have this? Because it's at the bottom added by the park ID number, right? It's going to be down here. That's where if we had done. This and we said. Order bye. Purk name and we ran. Now we would see Disneyland where we want it. But the the park ID is going to be 64, not 18. So let's let's add a city just to get another feel for how this works. OK, and we're going to say. I'm gonna open up my city. And I'm gonna let somebody else drive this time 'cause. So far so good. So what do we need to do here? I'm gonna make sure that you can see the. OK, So what do we who wants to drive? Somebody start talking me through how to do this and add a city into the city table. We insert into. City in the parentheses. City name., population., area. City named well if we're going to go in the order that we do here. We probably need to have all of them. I would tend to say always do in the order you have here. Just skipping N shoot. Yeah yeah. The city named state abbreviation. Every VA shun. Oops. Population and area. And now we're. Return align and then will enter the values. Single quote Hawkins um. And then, single quote I add. Population is 30,000. And then, 3.1 Hertz shooting 38.1. 33.1 right. If we look a little longer here we see it says and Cicely, Alaska, with this information, right? This is where if we want to add a second row at the same time, we would just put our, and we could do it on the same row or we do a separate row, right? In this case, you probably fitted on one row. I tend to do it. Your area is wrong on the values. 28.84 thank you. So now I'm going to do Cicely, Alaska. Let me do the same thing. Let me do seriously. OK. So, being once we've put in their values once, we don't have to do it for the second one. This you only have to specify what the columns are wants. And all I'm in the values when all we have to do is yeah, it's just a common limited set of things in parentheses. OK, so right. This is. Now, so I say Sicily, Alaska, right and I'm gonna say woo I have with a population of 839. Area of 11.4. Type. So. Again, we're gonna.; Go down here, just make sure it slapped star from city. OK, this select start city. And it there are 345 rows affected, so now I'm going to do this. I'm going to run this. Inserted 2 when I select start from whatever. It's now going to say 347 and like before. Even though everything else in alphabetical, they're gonna show up at the very bottom. As Hawkins, Indiana immensely Alaska. Auto generate some ideas for. Then I have a few questions about inputting the values. Sure, so for Disneyland we put in the date. We didn't have to do it like we've been formatting. The date. Is that something like? Does it just know to format it correctly or? Let's try the other way, just because it's easiest way, I'm gonna say. Wavo so we're going to say. So that's the way we've been doing it, right? So let's go ahead and insert it in. OK, this time I'm going to do that was in the parks, right? So now I'm going to say my park. I'm gonna show my properties and I think there's one that says data. It's wanna make sure I'm I'm trying to figure this out 'cause I'm trying to learn PG admin. How you do it? Somewhere in here it will show view, edit data all rows. Sort of like you even tells you the slack statement it uses. But it does it for you. So then when I go down to the bottom, I Disneyland Nuevo and it has the same date. So it will accept different kinds of formats. But as I've said, this can be ambiguous. If you have somebody who happens to be used to, you know European formats instead of American formats. So, but it will do whatever the locale. That you're in that it thinks you're in. It will use that to try to interpret if you use the more explicit version that's unambiguous between the two. OK then I mean. Is there is a really good point? It's sort of one of those. OK, So what do you do? Well, I don't know. Yeah, my main point into that is then what happens if you put in something that. It doesn't accept that it won't format correctly. Does it give an error? Does it still force it in there? Really this is this is our chance to try this stuff. Afraid to change mine, I'll just let you do it. So we're going to say this one and we're going to say it's in. Year 1 and it's in the month 13. OK, so it's not. It doesn't know what to do with this, right? So what is going to happen when we try to do that? It's going to go ahead and do it well. What will it put in there right? I have no idea, just curious. It clearly thought it would do something and now really intrigued to see what it thinks it did. View Edit Data last 100 rows. So yeah it said 2017 oh 113. OK, alright well I know to watch spelling and typing now 'cause it will force something in their excellent well OK let's try something that works. 'cause there should be one that we can. We can say OK we can say 31. 31 31 right. We have a lot of disneylands around. So we're gonna. Alright, it says Datetime field out of range 3131 thirty one. I don't know you need a different date style setting. OK, so if you if you mess it up badly enough. It will give you an error, but notice that that one. 3130 or whatever 13 one didn't give me that kind of error. Having had lots of fun there, let's see if I can get back to to what I was. So. What's going to happen if I try to run this again? Will it make another one? But it'll have a different unique value, so you'll have another OK, so be careful with that, because we may think the city and state name together are of a valid kind of limiting factor. But the only limiting the only constraint we got on here. Is on the way. There's more than one constraint 'cause I get to the point where we show different constraints in a minute, but the only one that comes into impact here is the unique ID, which is the primary key which is the city ID. So when I go ahead and create this, it's going to go ahead and create two more of them. And be perfectly happy about fact that it's just created another Hawkins, Indiana, in Sicily, Alaska. Let's keep going down here and at some point we're probably gonna have to like change some of this stuff. And we'll, we'll play around with what we've got and see if we can undo some of our problems, or whether we should start over and and not let it happen anymore. Not OK, since Disneyland is in California at a record representing that to the. Park state table. Right, So what I'm doing is I'm saying now we know we have our Disney. And we are going to be rendered. Right, we're at it so that we we have that you know that associative table that we needed. Perfectly. Steak. So far so good. OK. Here I'm using essentially a subquery. To find my information. I know that I'm going to use California. For my part ID, I'm going to get from this select statement. See if that works. Part name means an under score. Something that's parking right, yeah? Making sure I have my syntax right, I'm not at all sure I do OK, so it inserted that. Let's find out whether they actually inserted what we thought it insert. So what it should have? We are looking for the park state. We should have the park ID, the park ID. Should be the one we had. We just added for Disneyland. So if we look at our parks and we look at our view edit data and we look at the last 100 rows. So that we get the the ones at the end. OK, we have Disneyland should be 64. So what we should see in our park state table? Park State and we should see. OK, so we have 64 in California. Alright, I'm gonna do. I'm gonna skip ahead and will come back to it, but I just want to say OK. What if we're trying to put a value in there that it isn't a valid? Um, a valid park ID. What's it gonna do? It's just a number. So what if we went back to our lecture and we said what's gonna happen if we try to make a new park state for? We'll do it. We'll do it using values. OK. You say values. 75 CA. Hey. So if I try to run just that. I'm inserting a new entry in there. What's it gonna do? He's just gonna throw up 75 in there even though it does relate to anything. Gotta get mad. The constraints as well stop it right. The main thing is it's going to get mad, but it's going to get mad and it says it violates a foreign key constraint, which is that your park state. You know this fkey Park State Park. Alright, she whatever is not present in the table, we cannot add something. If we have a foreign key constraint now we could have a table that had a park ID and never make that foreign key constraint. And then it would let us put an integer in there that didn't work. But by putting it as a foreign key by specifying that it a foreign key and what it links to. The database is now going to enforce that constraint. It's going to say you cannot use that. So and so there is some sort of like. Order in which you should add things starting from the most unique identifier is that the big picture is that the point, yes, that's gibberish to everyone else, but the easiest thing to say is you better not create associative tables until you have created the the things which they you know are going to use. You can't create park state table. First, and if you add but if you add a new car. It will yes need to be added before you can add this constraint and so it becomes a sequence of things and we will later on talk about transactions you guys read about him last night, but but transactions and how we grouped these things together so that they all happen more or less at once. But even within a transaction you need to every every line has to actually be valid. When it gets executed, so you have to add them in the right order. You can't just add a a reference to a park that doesn't exist and you literally later on. We also see you can't delete. One if it has. If it's used somewhere else. So it is very it does a great deal of work for you in maintaining referential integrity. Referential integrity, that is one of those I will write down on our own CHEAT SHEET or whatever. That is one of those terms that may well come up even if nothing else does in your. Interviews and things like that. If a fancy word that just means everything is In Sync so that you can't, you know you don't have any. Nothing points to places it shouldn't point. Nothing is inconsistent. So. Get rid of that line. So in order to solve this problem and there are, there are different ways to write the same thing, but in order to solve the problem that we have. Of. Finding the part of the right park ID. We can use this this method to create the using a select statement to create the values. Do you understand why it says select park ID, You know CA, which isn't part of the park at all. If if if anyone doesn't get it, the reason is that this is still mine. That tells us what to print out. And by print out it means return as a value set. So. The selection really does. It goes to the part, it constrains it, it filters it down to just this one, and then it it prints out the park ID and then just a literal string. So the end result of this. Is a a. The end result of this is going to be whatever that party ID is, and that literal string put together. Another way to do this same thing. As you could say, values. Here's OK. You will probably see this more. I think this was in the in the reading as well where you'll say. And unlike other things, when we do subqueries, we tend to to put them in. All on one line. Right? It's this would do that same thing. What it really is doing is it's. Anytime I have a subquery inside parentheses inside parentheses like that, it's gonna return a value. If it had returned more than one value, it's like it does a series of those values comment. It's just a way for us to get the actual unique keys when we don't know them. Like what I should probably do here. In fact, I will. So they don't confuse the heck out of people. I'm gonna say. So when you look at it later and you see that the coding here, these two things are essentially equivalent. I'm just saying you could do this. Either these two ways. So now we get 2 updates. Updates are just what they sound like. OK and insert says I'm going to create a brand new row. An update says I already have a room. I want to find that row and I want to update right and so I want to change. This is where we start getting. We start getting risky. Alright. Doing and insert you're only inserting however many rows you've actually explicitly put information for. You know exactly what you're putting in. You are not going to set up anything, which is unusual and dangerous, right? When you're updating, you have to be very careful that you know what it is your update. So if I go to. This is this is your your your easy case. Easy cases just update state. Set. State. Nickname he goes. Happiest place on earth was. Happiest visitor, of course, within double quotes Justin. Try to trip me up here. I take personally and they think they do. I like, I don't think it's they're just trying to. Anyway, they're doing it just to trick me. Where? State. Help I spell my. State. May abbreviation. You could have said Sidney, but and again. Alright, so update state that tells you which which table we're dealing with. The state nickname. When you use a set, the way you use it is a set value and then it's a common delimited series of things like this. I also wanted to change the area I would just do area equals and a value. So you can put multiple values, but the. For reasons that. Baffle me personally, I don't understand why they use the same logic as they use in in in insert, but the way you do it is a series of. This value becomes this value. We don't want to change the area 'cause I don't know what area we would set it too. So we're going to go ahead and do this. But first because this is what we do, let's say slacked. Start from state. State where? Stay radiation equals. CA. Changed it. Alright, so I'm going to run this. And he's gonna say. California has a state nickname of the Golden State. We're going to change to the Happiest Place center, right? So we're going to run this. Update update one record. Right then we're down here and we look at this. And we're going to run it again, and now it's the happiest place on Earth. All this is great. What if we want to do something where we use the value that's already there and we want to do something with it? So in this case, what did we say? California is growing by leaps and bounds, and it's added a million new people. We do something like update state. Set. Population. Peoples population. Plus OK Plus a million people. OK, so we have our select here. I don't know what this California biases, but population is. 39 million. Hey. We're going to run this. We're gonna add a million people to it. And it's gonna say. It is job, so we're gonna run it and I'm not gonna get it right. Right? And now it is 40 million. If I run it again. Spoiler alert it's gotta be 41 million. 41 million so we can just add those values. Let us. All right, we're going to change the capital. Now what do we find out about the capital capital? We talked about it yesterday. It's 264. So that's another one of those situations where wait a minute. The capital is a city ID and we want to change it to Anaheim. And I don't know what the city ID is for, and I I hope there is a city ID for an eye, but I'm guessing there is OK, so let's. So this much is obvious. So, and we know that we're going to set the capital. OK, anybody have any idea? Spoiler Alert, it's the same stuff we were doing upper earlier. What am I going to do so that I can set the capital to Anna? What would I put after the = Would it be a select statement? Then it would be a select statement. So we'll be doing a subquery of select a city ID from just to be clear to everyone. If you're going to do a select statement in place like this, always put in parentheses. Yeah, in presence is fine, I just want to make sure everyone knows that. So you said select city ID. Be from city. Where City name is equal to Anaheim. Let's see if I can spell in. I'm right in OK. So and then you would do that then and then after the parentheses you do where state abbreviation equals California or CA. Make sure you're only setting the capital of California exactly. Where stated previous. Alright. There's something wrong with this. This is almost completely correct, and it's not quite good enough. What is the thing that is missing here? What if there is an Anaheim, Missouri? This rare statement only impacts. In the update. So I would need to have. This. Also in here. To be sure that what I'm getting is the. Is the city Anaheim that is in California, 'cause otherwise when I'm going to wind up doing is I'm going to say, you know, in High Missouri into the it's the capital of California and that's gonna upset a whole lot of Californians. 41 million of them now from right here. So. Let's see. So it was 264 before. Still feels so Dagger run it probably never ran it. Alright, syntax error new go where what did I do? Where, oh sorry I said where and. How? Alright, I'm done my update at the back. And now my capital is 9, which makes a certain amount of sense, 'cause Anaheim is much lower in the alphabet and we gotta they were added alphabetically. So I'm guessing that's right. We can check. But instead of checking we're going to take a break, we'll be back here at 10:15 and we'll keep going through this, OK?

We have to communicate a lot and and one of the things which has been talked about some is the idea that. Asking everyone is everyone clear on this? Doesn't necessarily best communicate the question. The real question is OK. Which part of this really confuses right and that is a better answer there always something that's confusing and and and instead of asking, you know, do you have any questions? It's like, which questions do you have? So in total transparency, when I answer those things, it is merely 'cause. You know, I am standing on the shoulders of giants who are smarter than I am. Will say this is what you should do. So at this point, what questions do any of you have? for a similar idea is that as an author one of the things you do when you were an author for kids and you have a rough kids have no patience whatsoever. You simply tell them to hand them the rough draft or whatever, and you say. Circle the page where you got bored. I put it down and if they never get bored and never put it down, that's great. But that's about how much they can do. They will circle the page that they where they where they stop treat and that will then tell you that you've slowed down or don't whatever else. So. So if you look down at and you look at the questions and the things we've done, is there a point? Where you got to this where you said oh, oh, I have no idea what he's talking about anymore. Right. Are there and so sort of you don't have to be very exact about things. What you have to do is you have to come up with values that it can somehow interpret.any of those in today's lesson? One more year. it was Thinking about what is the computer doing? Think about it as a whiteboard, right? And what the? Whatever I do in any of these, it's trying to get a series of things written down on the whiteboard, and then maybe erasing some of those things. So when you do a a selection with aware for instance. It's almost as if you're writing down I I'm doing from park. It's almost breakdown. Everything from the park all on the whiteboard. I think it's creepy. Whiteboards are all the things and they say we're part name equals Disneyland. And I erased all the lines except for that one that says didn't. Right, so when I do something like this line 24 insert into park, I'm going to try. I'm trying to park ID in a state of Revy Asia. At every stage where you have parentheses. Don't worry about weird storing it. Think about it as a whiteboard. What's happening is I'm going up to a white board MEM saying select park ID from Park. We're parking equals Disneyland, so it's from park. I read all the parks down. And then erase all the ones that aren't Disneyland alright. And then I erase all the values in the whole row except for park bench. Now all I have is the park. Sitting on my all by itself on my my white board. And then. I look for the next thing and it's just a string CA. So now I have whatever that park ID is, 64 for instance. And then see a. And and all it's doing is trying. It's doing that writing down the way board. So once you've got your values. And you've written all your whiteboards. If the last thing it's going to do is say, OK, where am I putting those values, and then they would say into park ID and state abbreviation. But so all these nested select statements, whatever all they are doing is coming up with information to write down on the whiteboard. Right? So in this case, if I had replaced this whole thing, if I happen to know the number was 64, I could say OK, I'm just going to say 64. And what would I have here if I wrote this on the whiteboard? I have 64 CA. All I'm doing here is getting at that 64. As as as an answer to a query, I don't hear what it comes out in. I don't care whether it's in a table, I don't care what it stored. Add SQL will handle all the messages, I'm just writing down everything in the park. iRacing everything that doesn't match this and then erasing all the columns that don't match this. And even though that seems like an awfully long way around, but it's gonna come out as is just that one thing. So if I were to be explicit about this, if I went like this. Alright. Simple run when I'm inside parentheses. I'm not sure whether real or not. OK. OK select Park ID from Park where park mean was. Disneyland returns a 64 so it's going to replace whatever it did with 64, but let's let's back it out a little bit further. Let's say OK. Where do we oops, sorry. Let's do a select star. Really, what it did was it did OK. We're going to do this first, really. It did select start from. Perk. And I mean, all of you, this is this is the sequence of operations. This is what happens. OK, like last my. Where was I? OK, so put all this on the whiteboard. Although, however many parks 66 now right parks. And then it erased all of the lines except for. The one that has Disneyland. So that's the second part of what it does. Now that's all I have on my way. Now. Think of it as erasing. It just erases everything that it doesn't need, so it only needs perfectly. And internally, this is really the way it is. Kind of the way it is doing it. I am trying to do way too much there, so let's say that. OK, and there's 64. So that is the process that you're going through. It looks all fancy and whatever all it's doing is trying to write the values onto a whiteboard. Does that help? Any other questions that people have for? We're gonna we're gonna assume that for now you guys are good. But please don't hesitate to ask if you got lost. So we set the happiest place on Earth. Do our. K. Increased by a million. OK. Let's get California back to where it was. OK, we're going to do it by two million, because right now. Let's do our thing. Notice that I can do this even though it's in common in a comment. Right, so we're at 41 million and the happiest place on Earth, and we want to say how do we get back to where we were? And we're going to say, Oh my heavens. This is just putting all the stuff back together, so we're going to say. Right, who wants to? Who wants to walk me through this? Anybody? Patty, you shaking your head like no no, no, not me. Alright. So I have to pick a volunteer so we're gonna have to do update state. Obviously because we're updating California. Stay stay. And then set nickname. Set statement then I don't know. It's just stating stating. Equals. Select I went now. Sorry, I'll leave you plenty to do on the other ones. I'm gonna do a car. What else are we gonna stack? Um? Population minus 1,000,000. Happy lation. You're gonna have to say equals end. Each one equals population minus 1,000,000. Yeah, we need 2,000,000. Just 'cause we did it twice. Let's get us back to where we originally were OK. Now we're going to change the. Capital back to Sacramento. So would that just be capital equals Sacramento? No, what's going to happen if we do capital equal Sacramento? Captain. Let's because we. Just to be clear, let's do the let's force it only to California, so we don't really mess things up. Where state abbreviation equals CI. OK. What's going to happen when we run this? It's gonna say invalid input syntax for type integer Sacramento. So we can't use Sacramento as a string because we need to have the city ID. Wait for that capital. So we're going to have to. Hit enter just so we have space, so we're going to have to build this out a little bit. What we need to do is we need to get the value that is the city ID from the city that is Sacramento in California. From how to do that? So wouldn't that be kind of just what we did up top set capitals? Extremely similar to what we did up top, so similar that I will copy it. And we will change it, but. Your lives will be spent copying things and changing them. OK, all we have to do now is change this to Sacramento. See if I can spell Sacramento proof correctly. Jack Ryan to not round. My name is well, see what would happen. Let's let's do our. Sorry. Wait? So we have now again we have the capitalist 9 populations, 41 million and the happiest place on Earth. So what is gonna happen if? I mess up and misspelled Sacramento here. When I run it. It's kind of fail, but the point being that because it failed. It didn't update anything. So it's not like he's gonna update the parts of it that worked and not the other parts because SQL doesn't know whether that make will make logical consistent sense. If you put it all into one statement. Assumes that you want that entire statement to work. So when I did my wherever it is. Lost my. When I do this, it's not going to update anything because it couldn't update everything. So I'm going to spell my Sacramento correctly. And we'll try this again. Right update one. It will always tell you how many it updated. Now. Right now you don't really care. You're going to look at that and you're not going to pay much attention to it or whatever else that is going to be important when we get into Java calling SQL. Which tells you how many things got updated. Right, so that return value is what you will see later when we're calling it, and there is a return value or results in. It will tell you whether we're successful and whether and how many things got updated. Alright, so we do our select star from state abbreviation and we run it one more time. Now we have 39 million the way we had before we had the Golden State move 264, which I vaguely remember it being Sacramento, so that it looks like we did the right thing. Alright, so remember we created Hawkins in Vienna. OK, let's go look at our. Select star from. City. Where? City name we'll do it like this like. That's just going to give us the cities that are around. So I got. Yeah, not alphabetical because Hawkins got added at the end and they actually added two of them. Right, this will be important in the moment. All right there are two of them out there, so when I. Go to delete. What I would recommend this is actually not what I would do. Whatever would recommend is that you say I think I'm deleting Hawkins, Indiana. Let me just find out whether or not. Sorry. If, when you're going to delete stuff. It is gone. You're doing it. It's it's gone. Right and what that means is that. You can't recover it easily, it's just gone. I cannot emphasize what that is like. When you it turns out to be a production database and all of a sudden you have dropped. You know an entire state. From your production database because you weren't thinking quite about the query, right? I would always recommend that before you delete something. You start with selecting the thing you think you're deleting and go from there. So this is. Play making my life so. I could say I want this. This is not really enough. Happens. There are two of them. I really want this is actually. This is not going to change anything, but I really want I'm gonna say and state abbreviation. Equals I am. This will still do two because we added it twice, but the the point that I'm really making is if I did something like Columbus. And I just said Columbus instead of saying Columbus and state equals 08, it would get rid of Columbus, Missouri as well as Columbus, OH. Junior Unique case will be 2. And then when you are comfortable with that, then you can change it right? Easiest way honestly is just. OK. Easiest way is to take what I had right there. And just change the select star when you are perfectly comfortable. Change it. A change in meat. OK. I hope. To rent it out properly and from city. Where sitting equals options and state abbreviations equals I am. So delete is all by itself, don't need to have any special stuff. It's going to say what table you're really dealing with and the where is critically important. If I said delete from city and did nothing else, I just highlighted that I would delete every city. And it would not ask me. It would not say. Are you sure you want to delete? You know, 347 cities. It would not do that. It would just go OK. It was shrugged. SQL shrugs alot. OK so you are in charge so now I can do this though and I can say I'm going to delete that right? And it's gonna say delete and it's gonna tell you how many it deleted. I deleted two because I happen to have two cities that were often Indiana. Usually this would be one if it says 347 and you messed up. I will especially point out that this is one of the easier ways to really screw up when you're using PG admin is to not is to start selecting from the top. One of the things you may notice I always select from the bottom. It's insert selecting from the top and not select enough. OK, or just like like there and and not realize you've only selected some of it, it's gonna run whatever the heck you tell it to run and it's not gonna warn you right so? Having done that, let's let's try doing something with this. We're going to say how can can I ask a question about that? Absolutely? What if you added it twice and you only wanted to take off so it was only on there once? But if you only wanted to take it off once so it was still on their excellent question. So let's. Let's actually look at that situation for this. Easy to do so. We're going to go back up into our. We are going to insert. Insert Hawkins twice. Notice I'm only inserting. Just 'cause I know that was all there and it was easy and whatnot. I'm going to say Matt and now I'm going to say my. So we're doing two separate things. Outlook say. Alright, so the select and delete up there is what we really want to do, but I'm just going to do this so we can see what we're talking about. I accidentally created this stupid? I'm gonna figure out which one is the duplicate. This is likely to be a manual kind of process, unless there's some way for me to tell them apart. I'm gonna say you know what I really wanna delete. 351 because it's the second one of the two. And so I'm gonna say From city. Square city ID equals 351 like I would tend to do that very explicitly. Because and look at it and then do it. And I say my delete and I do my delete and then I'm going to do my. This could validate and verify what I've just done. It did say delete a one, so now I only have one. So now when I run this thing so the original one that we had, it's going to go ahead and delete what I expect, which is the one thing it's gonna say it deleted one. So. So another question. If you screw up and you accidentally do wind up deleting all your cities is what do you do? Do you just like accept or lovely countries or don't have extradition agreements? You know, no, the answer is that you you, as quickly as humanly possible suck it up until somebody. You know who has more control than you that you screwed up? People do it. People drop you can. There's a there's one that will allow you to drop an entire cake, OK? Very dangerous, you know. 10 million records. Proof stop right. It can happen, but if it happens you tell somebody right away because they may have backups. They may be able to have it like there are different ways to do that in the real world they they recover from that. But you will. You will bear the scars and mistakes that you make. And. Most developers have horror stories of the time that they did, but whatever thing that really screwed everybody up, you know it. It's going to happen so. But really, I'm serious. The number one thing is, don't mess around. Don't try to fix it, don't whatever else. Tell somebody who has more authority than you until you are the person who somebody is telling you that they did it. Tell them that you did it and after they finish yelling at you for a minute or two, they will go off and do the best they can to fix. They maybe they can't fix it, maybe they can, but. Yeah, we will show a little bit later here. How you feel, at least insulate yourself somewhat from this whole situation. not sweet as ordered by doesn't work either so. You're gonna delete only five, but it it's at the right five. I don't know. It's some 5. We know at all what it's how it's operating behind the covers. If you if you run around me and delete it, I don't know. Having a little trouble understanding. Near me. But try again Donnie. Can you hear me? Yes, it's I don't know what happens when you guys get the reverberation but it makes it a little harder. Probably have to do something or do something else. Go ahead. OK, if you if you delete the wrong thing when you thought right, right, right, right? And you only need delete late one line, but you don't know one line line is. Is there a way to see? For you. In in real production databases, there is probably a transaction log that will tell you all the things. There may even be something in PG admin that I don't know that will tell you the last transaction or that you know what it impact. There are. This is. It becomes important that there are ways to then return the thing that you're you're impacting, or select it first and store that so that you know there are ways to do all of that. Right? Won't get to that little bit. I'm on that slightly. I wasn't paying attention, but I'm actually curious about it. Is the city ID still incrementing or is it filling in the spots that we're deleting cities at? So like if we made another Hawkins, would it be 352 or would it start filling in from 350 again? When in doubt, just do the damn thing you know, because that might be a way to track if it continually increments. Yeah, it's gonna be a lot harder in a real life situation because other people are are impacting your thing at the same time. So if I go into my. Select it's 352. So it does keep incrementing even though we deleted the previous ones. That's cool. Alright. From city. Where? So do you select first to make sure you know what it is you're doing? Where population. Is less than 1000 right? I should give us. Both my Sicily Alaska's that I add 'cause I added them earlier too. OK so if all of them to delete this those two. Alright I'm I'm I'm good with that. So now I've changed that from a select into a delete. I do mind delete it impacted 2 because we had two silver Alaska's and we're done. Notice so here notification messages explain data output. Maybe I put us from the last time so it's not from this. There's nothing that tells me what it is that I I just did. K. Now, referential integrity. This is when we get into let's, let's try messing around and again we can mess around. Part of the reason make your mistakes on the United States database that you can recreate and you know on the fly in a moments notice. Don't make them on a production database somewhere where someone is going to be rough, so let's try adding a city. OK, so how do we add a city? And we remember how we can add a city. It's gonna be insert into. Yeah, so let's do insert. Into city. And then you're going to call the parameters for city. That is a bunch of stuff. City name, state abbreviation, population and area. Wait? You're gonna insert the values of. Ruthville XX11. OK. Values. I'm sorry. What did you use? Call it Lucybell Ruth evil, of course you did. Ruthie Ville. \*\*. What's population 1? Right plus the area 1. Alright. So we're going to try inserting. This. Insert or update on city violates foreign key constraint. FK City State in that The \*\* is not in. So. Nick can't do it. Let's get rid of California. Let's get rid of California, 'cause you know. It seems worthy. I guess. In theory I would in this case I I know what I'm doing, but I'm gonna say from right slapped star from city. Where no wait? What is it where California stated previous and right? I'm I'm doing the wrong thing. Select start from state. Where state. Previous \*\*\*\*. Right, so I'm first going to select this case. I know exactly what foolish thing I am doing, but I I know I'm going to slack. There it is. So let's try deleting it. This works right? They say it violates. Save this foreign key on the citystate on Table City because the abbreviation is still referenced now this is different the last time it was not referenced. This time it is still referenced in the. State table in the city table. Alright, from Table City it is still referenced. So in other words that have references to California, if I wanted to get rid of California and like trust me, we are really wanna get rid of California. But if I wanna get rid of California, I'm gonna have to delete those cities first. Right cities where the state abbreviation. It has that and you know what I mean. Probably have to delete the states. I'm in the Parks dead reference it and and then I have to re remove those from the park state table thing. So I mean like it gets complicated. Get cleaning up. We'll take several steps, but let's go ahead and try deleting Disneyland from the park table. You didn't really like Disneyland all that much, so we're gonna delete it from the park table. So we're going to say do delete. Right? From from Parker yes part table. Where? Park named Peoples Disney. Try doing that and you know what it's going to say. OK, I can't do it because in the park state. Somebody is still referencing this. So let us try at then it says try to leave it from after deleting its record from the park state. So somebody want to tackle this one? This is slightly more complicated. How am I going to find the entry in a query? I don't want to do 2 queries, just wants to do one thing, one delete. How am I going to delete it from the park state table? Could you do? I'm out. Having trouble hearing still. Somebody else is gonna have to do with who's not reverberation. Hi, who haven't we heard from today? How are you Danilo? You wanna tackle this one? Again, delete from So what we wanna do is we wanna delete the. Record from the park skate table. For Disney for. Pretty land. So what do we do? Delete from Park state? Where park name equals Disneyland. Park ID equals 64. Well, we don't know what the the the park ID is. Can we added it and deleted in a couple of times so it may not be the same anymore from Park. Let's assume we don't know what the name is. The the ID is. Let's put in our where 'cause we know there's going to be aware, and we know that we're looking at the park ID, right? Because that's what's in there. You gotta park ID and the state abbreviation. Like we we can start with the state abbreviation because that one is easy. End alright stay perfectly. Alright, so how would we get? Just generally, how would how would you get the perk ID if you knew the name of the list? But not not in here. Just if you just wanted to do a query just to say what is the park ID for Disney. The select star. From org ID or Park state. Well, no, but we're gonna try to. We need to find it. Where do we have the name? So the parks? Justin Park Park. So we need you in park. Where what's our where clause gonna look like? Park name equals Disneyland. Right, and again, when we're doing this, we say, OK, this works, we've got select star. From there it's going to give us all the information and we don't need all the information. We just need the park ID. So how do we change our select to make sure that we. Josh had the perfect. So instead of saying star, what could we say here? Say park ID. OK, so now we're going to try it again. We're going to say alright, there's our select statement. Alright, so it gives us the result we want which is 64. All we really are doing here is saying we want this query. To be right here. So we're gonna create we're just gonna put it. Right, you put it online or not, or or however you wanna do it, but essentially what you're saying is I'm gonna make my hark ID come from. I'm gonna do a subquery to get the value and then put it in here. Right? So if we go, where to go do this? Right State abbreviation does not exist. Alright, so I'm deleting from park. Where am I getting my state abbreviation from? I have too many perks here. What do what do I what I think I'm sorry yeah, am I deleting from park? Isn't it park state? I believe from Park State. OK. Now I do have a state abbreviation and I do have a car fake. So once I do this OK. So now if I go for yeah OK so anyway deleted from the park state table. And now I want to delete the park. From the park table. Right? Because I have deleted the reference that that prevented me from doing this before, now I can I should be able to do this successfully because there isn't a foreign key that references that that part. Is everyone good? This is a little. This is a little tricky to understand. There's a perk state table which is association between the city between the parks and the the states. So if I mention a park in that park state table, I can't delete the park because there's a reference to it. It has to get rid of the reference. What nothing is pointing at that park ID from any other table. Then I can delete it. So I got an error before when I ran this. There's delete from Park where Park named equals Disneyland, but when I run it now? It will delete that busy. That means I can no longer have an entry in my my state per. I mean my park state table. How can you figure out the order of deletions that you should do by looking at the erds? If you looked at the picture, how do you figure that out? Take a look at the at the Rd. It is from. Just trying remove stays, had it. In the lecture. OK, so yes, it's here. You probably have various places so. I mean. The last thing you should delete. Is whatever wherever it has a primary key. And the first thing you should delete is usually where there are associative tables. Anything where there's an associative table. It's only point in life. Is to provide that connection between the two? That's not necessarily enough, but that is a good first step is to do is to get rid of it from any associated tables. Once you have cleaned up associate tables then you have to think of places where it is a foreign key for other purposes. So for instance the the city. Doesn't need to have an associated cable for the state. It just has a direct foreign key, so if I am going to delete this thing. I looked at any foreign keys I will need to make sure that that is cleared up first. If I'm trying to get rid of state. And it has a primary key foreign key connection to city. I really need to get rid of any cities that use that primary key. So the very first thing I would always do is the associated peoples and the second thing I would do is any other tables that have a foreign key pointing back to it. So it state is very, very difficult to delete. State is very difficult to live because it's used a lot and. Yeah. But I mean that makes sense the bigger it is a country is harder to lead them in state because it's got even more relationships. Relationships are are because relationships are so important to everything. It should be the hardest. Delete the things which which in a sense are the top of the chain. The lower down you get, the less likely it is you're going to have problems, that's. Any other questions on that sort of? On on what we're doing here and. How you know how I how I? I know that I need to delete things before I delete state so that it will work or referential integrity. Anything on that? You understand someone said to you in a question right now, somebody said. All right, you know. What is referential integrity? would you? Would you have some ability to answer nothing you would have a clean answer. I don't even have that clean answer. What would you have some ability to describe for that? What it is that preferential integrity needs? Go for it center. Oh no, I was just saying uhm, yeah. I'd. It doesn't make sense. It it makes sense, but it's it's it's it's. It's still fuzzy. It's not as clear as I would you know, as I want it to be, but it you know, bits and pieces. I mean, again, the thing I tend to think about is OK. You've got a mail person and the mail person has a list of of addresses that they deliver to. They have a row or whatever. In the real world, they might have an address that doesn't exist. And it's just going to. They're not going to know what to do with their piece of mail. In a in a relational database, it's as if there's you cannot have a piece of mail that does not have a real physical address that goes to a real physical place. And it's as if someone could not right onto the envelope and address that didn't unless that address exists. So you have to build the house first. Once you build the house, you can send mail to it. Right if you try to delete the house you try to, you know, bulldoze over the house. The right relational database will not let you until all the letters have been delivered, because there are still letters that point to that House. Basically tough to meet a certain criteria through, go through the appropriate channels for it to get to where it needs to be right? You need it, or like anything that. Links to anything else needs to be have integrity. Nothing you can't. You're not going to run into the null pointers that you do in Java because the data database will will maintain a lot of it for you. Referential integrity is the thing they're trying to maintain that OK, I'm never gonna point at something that doesn't exist. I'm never gonna create something in a state that doesn't exist. I'm never going to, you know, delete a city. I mean a state when there's a city that's in that state. I'm never doing any of those things because they all are joined together. Relationships are super important. The reason we call it relational databases because the relations are the really key things the data is there, but the relations are the really key thing that makes the relational database what it is. Donnie put a question in Chad I think is mikes not working today through in in there. See that chat question, which might never see these things OK? Integrity of property of all relation, all relational databases and by definition will have. Must have integrity they must have. Referential integrity that is the the definition of a relational database. When you deal with databases that are not relational, they may or may not do various things to try to maintain some referential integrity. But it is not required. So if you have a database, if you have Mongo DB or something like that which is a Jason oriented database where every object may have different things, it is very hard for them to maintain that kind of integrity because there's you don't know that an object even has a state ID. So you don't have any way of enforcing it in the same way relational databases. It's not a property, not whatever, it is an integral part of the water relational database is. OK, not right sure. Let's take a break until 1119 and then we will dive back into this and. We're getting close to the end here. We just start getting a lot of little picky things and I want you to understand, OK?

and if i look under there it will have the definition which going to have And if I look under there, it will have the definition which is going to have that it isn't. It is. And all that precision stuff that is valid. Don't worry about statistics or storage constraints. We'll show you, oddly enough, that default, which is I don't think of as constraint, but I guess by their definition it is. This is the thing that says, ah, I'm going to take the this this sequence. And add one to it. When I create that, that's my default value. Right. Now I want to show you something about that, which is that you can actually set. So the city ID. You can set this value. It cannot be one of the existing ones. And whatever you use, it's the next one is going to be higher than. So right now if our city ID's. Can't remember how many cities we have. We have a third. They're like in the we're about 350 something because of our we added and deleted a couple of cities over there if we. The set of city ID as 1000. Then the one after that is going to be 1001. So it's just gonna sequentially do it based on that? But to show you again I guess. That default value then is the thing that happens if I don't set the value. This null not null thing. Alright, says. Is it a constraint? Is it allowed to be a null value? Can we have a null value? And again, that's partly up. Do I need to set that value when I am, you know, inserting a row? But it also means that in that whole join scenario, am I ever allowed to come up with a value for this column which is null right from the purpose of the constraints right now. What's more important is I can I set it to null or can I leave it to be null? Because I have a default value. If I don't set it, it's going to have a value, so we will not be null, so this would have require my trying to explicitly remove a value. In order to to get to that constraint following your account. But if I go to something like my population, let's look at my population in my area 'cause. Those are the two were abducted deal with. So if I look at my population, my properties and the constraints, population cannot be null. And it has a default value of 0. Area. Right. Cannot be null, but it has no default value. Right. If you think about this, it is perfectly possible to have a town or city or something out there and have nobody left living in it. There are ghost towns floating around. They may have a name, they may have all the information. They have no population at all. But they can't have no size. So that's why it is set up that way. So we're going to add Smallville, Kansas. And we're gonna try to do it without all its columns and just see sort of what we get when we try to do this. So we're going to say this is the not null. We're going to say, alright, so. Insert into. City. We're just going to say the city name, it will create one, an idea on its own state abbreviation. That is required. Now use. Smallville. Kansas, OK. So that is gonna try to add a city called Smallville, Kansas and only specifies the state name is state abbreviation so it doesn't specify the city ID which will get generated automatically or the population of the area. When I try to do that. It's gonna say null value in column area violates not null constraint. Now partners that tell you to like one of the things you're going to have to do once we get into Java calling these things is these error messages that come back. They actually have, you know, state values and numeric value for it. And that will give you like some textual value and you're going to need to start being able to say, OK, how do I deal with that? But I'm, but I'm trying to do something and it won't let me do it. You're gonna have to think about exception handling or anything else you're gonna have to deal with how to deal with that. Gives an error. Yeah, question sure. Why does it just says not no value for area and not not and not the same thing for population. Also excellent question and that is going to we'll we'll see. In the next one, why that works? Why it's like, no, I mean excellent question that's. Correct. I'm now going to do this and I'm going to say I'm going to give it an area. And by the way, the order of these. These columns. Doesn't matter at all, except that your values need to match up. I could put area first. I could put City name first. I could put whatever I can do any order. It is usually a good idea to put it in the same order that they actually show up in the table. If you're, you know when you're designing it, but you can't. You don't always have the design of the table here. You know the values. You see an energy, you see a PRD. It doesn't necessarily put them in the order that they actually are created so. You can put these in different orders or I can skip values, but in this case I'm skipping a value, so I'm going to say that Smallville, as I don't know 0.5. Square clouds and taquitos point. I only got a complaint here about the notnull. Alright, this is a constraint which will allow me to operate. It did successfully do it and the reason there was able to use the population even though I didn't have that even though the I didn't, it wouldn't allow me to do it without the area, but it would allow you with population is because. When we looked at our properties and our constraints. Population has a default value. So when I didn't specify it. It went ahead and specified it for me. It used the default value. And so if I now went out and looked at Smallville. OK. So we're gonna say select. Start from. City. Sure. Name. I'm gonna look at all the Smallville's because I don't really care that much. Right. And so here we have. If Jet Auto generated the city ID and because we didn't specify the population, it gave it a population of 0. And it set the values for the city name and the state name, abbreviation and the area that we gave it. So I was able to create it without all the columns because one of the columns auto generated in one had a fault. So now we've dealt with the not null, not null. It says if I have no default value and you're trying to set it to null. Or not set it, it's gonna fail. The default is is more of a positive constraint in that if I had not set the value, it will set it to that value. Right. The unique constraint is back to more of a preventing you from doing things. And so if I say. Update. State. Set state nickname I think. Doesn't matter that multiple ones are. Doesn't matter if there's no value at all. State nickname, for instance, is not null, is set to no. So it doesn't matter. It has no default value and it doesn't matter whether it's null and that's why there are some states with no state in. OK. Let's see if Trevor, where we see our constraint on this one. I'm not entirely sure how we set it. I know how to set it in like from code, but I'm not sure how we see it under constraints. Did I miss that type whereas identity? I think that might be it. I don't know 'cause it is already set. Like I know this name is already was defined. Back to me, go for a brief moment. We are going to go back to here since we now know enough to be able to read this a little bit more. We're going to say, OK, so here is where we are creating a table. They again, you don't need to know the details of creating the table or whatever else, but here is where we created that state table. Alright. And the constraint on the nickname and on the state name are that they are unique. Right. So that says we are only allowed to to use them once. I'm just not entirely sure where. Peach I'm. I'm slowly learning PG admin to figure out where to find the information, so I already know that it has this constraint and I will dig through it and find it. But like just the way I found some of these other stuff right that moment I'm not positive in any way. I do know because we added it right here. That it has that constraint. So now if I look back to my lecture. And I go down to where I am. My constraints and I try to set the state nickname. So Vacationland, first of all I better set a where clause. He's gonna try to change all the states so. So I'm going to try to set the state of previous so the state nickname vacationland which has already used. So when I go ahead and do that. I'm gonna say duplicate key value violates unique constraint on this thing. OK, so when I create a particular column matter how I see it, but I create it, I can set constraints on it but say I don't want you to ever allow to states to have the same nickname. But I don't really care if they have any. Nickname at all? I think about what that means, though. Unique then doesn't mean you can't have two states with with annulment mean. No, it is not considered a duplicate in this case. So if you want them to be able to have a value. But not have to have a value. You can still say that value if they do have it is unique. Check constraint is if you looked at our erardi, OK, you've got your city, you got your state, you got your part. You've got park state. OK, we have this thing called Census region. So what are the constraints on the Census region? So that we don't just, you know. Put it somewhere heaven knows where. OK, you don't put it in a non existent. Sensors region. We have a couple weeks to do this. We could have a table of valid census regions. Right. And we could say then that becomes a key that we're using it with a foreign key in the state and it's a primary key in the sentence region and that would work. And if we had more interesting information in our Census region, like if we cared about the Census region more, if we dealt with in more, we might do that. But if really all we have is just the the value and we want to constrain what it is, then we will. Do it then. We have a constraint, a different route. I'll show you that other constraint. So we're going to say update state. Where no set. Census region. Set the sentence region to southeast. So now it turns out that is not a valid sentence speech. OK, so I'm going to say where. Create. So it is going to complain when I do this. I hope no it violates the check constraint. So check constraint says I made up a rule of my own that it needed to check. Like it's not one of these, you know, generalized things. OK, so I'm going to say my sense, this region. I have my properties, I have my constraints. Mary, hearing me, I did this just a minute ago. I'll find this thing. I have terrible but funny these things PG catalog default no. Come on. All right. Well, I'll go show you in here because I I even saw this in ago, and now I don't see it anymore. But if you go into your. I think if you check the state properties you can find it in there. OK, that's quite right. Might well do it there constraints. Stated group but parameters, no columns. OK so in strengths and check there are multiple tabs and constraints as well that you can swipe through. Oh, there it is. Check. Right, so this you have to look at the state team instead of looking at the individual column. So check would say this is my my rule like that I wrote up since this region is null or census region: text equals you know. Baba blah northeast. It does all kinds of stuff. Why? Why would you not make like an enumerated table for the census regions? Why would you hide it in a check rule? Well, I mean it. I don't know database design decision. They they decided they didn't need that table and there's just another table they don't need. No I mean, but it's a valid question. But the nice thing about check, I mean the reason why it's worth seeing here and again I want to show you how it shows up when you create this thing. Is that I just say OK check and then it just checks a thing. And whatever that thing is, is going to be my. Like it shows up in it, it's difficult in the definition, but here where you see it it's pretty clear. I just want to check this thing. It's like an if statement on a or a four loop or whatever. It's your Boolean condition which just says. What is the best returns true right? Then it's OK if it rains. False. I'm going to have an error. So I could do all kinds of interesting things here that are hard to like. This is actually just a set of values and maybe the other reason is because they want it to be possible for it to be null, but also have that you know this selection. And I'm trying to figure out how that would work in the table. So possibly that's the. In case all you're gonna have a table where the primary key can be null, it just can only be one of them, so you know it's a. It's a design decision, but it's worth knowing that you can make a special rule about a particular column. That is hard to articulate with any standard logic, but where you just come up with a Boolean expression that says you can only have ones that do this. And remember, it's not just when you insert it. It means if you try to update it, if you try to, you know, let's say you didn't want something to have a population that was evenly divisible by 50. You could put something in there that would do that math and say not allowed to have. OK, side note but interesting story. When they went out to measure. Mount Everest. And they were trying to measure it and they they came up with the height. The height was an even. Like evenly almost exactly some number of meters tall. And it was like with it, I mean, you know multiple 50 or multiple of 100, I mean it was it was and they intentionally changed it from that valid number. To a slightly different number because they wanted to make it clear how careful they had measured it. So in order to show the care with which they measured it, they actually made it wrong. So that it could be a number that was not, you know, rounded up 'cause. It looks like it's rounded. There are times when you actually say no. I I will not have a population that isn't even multiple of a million 'cause somebody's gonna think you just said 3 million people and they want, you know, to know the real number of people so weird. But you know, all those things are possible. I know where come up with this stuff, OK? Start sitting around talking for, you know, three hours. See, Paula has a weird stuff out of your head. OK, so. Yeah, actually try it. I don't know what to do. Yes, I did. OK, so I I did my census reaching that is just telling you that the whole point of all that constraint table that constraint part of this lesson is really to prepare you for Monday when you're designing your own databases and you need to start. Thinking about what your rules are, what the rules you want to do, these are the kinds of rules you can do. Alright, so transactions and this is all that is left of the left. Just so you know, we don't have a lot left. Transactions are. They are the way in which we say, and this was in reading, and they have actually a very good example in the reading about the situation which you guys dealt with before, where you're deleting from one bank account and adding to another bank account. When you do not want a situation where you have deleted it from one and not yet added it to the other and then everything goes South and it it breaks OK, you don't ever want to be left in that position where the money has come out of one thing and not gone into the other. Well, that kind of situation can happen where you don't ever let's say. That you have some associated table with Park state. You don't ever want a situation where that park. Is not associated with any states. So you can say I need all of these different statements to happen at one time. Transaction say I'm going to start a transaction, I can do all this stuff. I can do all kinds of weird things. I can delete things I can, you know, add things. And none of it actually affects the real database. In fact, some sort of mirror database that exists in memory. It's all way too complicated under the coverage, but in theory, think of it as just existing in memory until you can apply it all at once. That's what transactions are good for. So. I will warn you that the way you start a transaction. Is it's. I always get it wrong. It's start transaction for the Java folks using post graphs and it's begin transaction. I think for the C students or the other way around I will figure it out soon because they they use different terminology and I had, you know help people a lot last over and now I can never keep track of which one is which. So the reader said they get up again, I'm sure. What? Think for us it's begin 'cause. I saw it on the and when you load in the database is it starts with begin transaction. Oh OK, so maybe even that allows both. I'm going to. That's a very good point. I think it did say begin transaction. I'm curious, but we're going to try this will do it. We'll do it both ways and we'll just see. So we're going to say, alright, I'm down here. You done all the stuff. OK, so we're gonna try to delete that small bill. Do we still have that small bill? Why don't we remember anymore I slept. City. Now select. City. Name. State abbreviation of just like start fun. Why? Why do this stuff to myself from city where? City name. It will Smallville. Wanna make sure we still have it? If we don't, we can always go insert it. There's our select, we run, we certainly do have. OK. So we're going to delete all the records from that and then undo it. Right. So we are going to begin the transaction. Action. And then we are going to From city. Where she name close. Yeah. I really should and you should get used to the idea that you should really do work. Yes. Chaos. So actually. I have lost my my train here. I'm in the wrong place. Just 'cause the comments about. I wanted to see it, so I wrote it and then I went down below. OK, so this is where we're going to delete it within a transaction. OK, all we have done is put it inside the transaction. If we want to finish a transaction and we want it to tape, we want to actually use it. Then what we have to do is commit. Right commit says commit all these. Take all these changes all at once. And one fell swoop do that. Right. What that means is the database is going to briefly lock itself from other people making changes. So there's no chance that if I've got several interactions, several transactions, there's no chance that somebody else will make a transaction in the middle. I'm going to lock the transaction date entire until all of it is done. Yeah, I I tell you that probably 'cause if you do something really long and slow, you're actually going to like. Drag everybody down. Transactions should be quick. Or they should be done in a way where you don't share the striking it down when we create the United States database, we're starting from scratch. You running on your own local database, you know, it's it's nothing. If you were in a production database, you might have certain hours a day when you do things that are more time consuming, that kind of thing. Alright, this is going to begin the transaction, I'm gonna run it all together. And it said all it said is commit. You may have noticed that when I do the United States, then it just says commit. That isn't means I've committed the whole transaction. It's all done. So if I go back in, he'd do this. Alright. There are no values. Now we're going to do the. Now we're going to do the undo alright, which is called the rollback. Rollback says go back to the beginning. Forget everything I just told you. No matter what I've done, you know. Ignore it. Go back to where we were. So I'm gonna start a transaction actually let's try begin transaction, see if it complaints others. I'm sorry. Listen, let's try start transaction. And see if they both work. I think they both work. Yeah, I just tried it. They both work. If they both work, I don't care which one you use. Whatever it is, I know that SQL Server doesn't allow both or something like that. I know that there's been this ongoing battle thing, or maybe they've both gotten to the point where they're so fed up with people complaining that they've added aliases for both. I I don't necessarily know that does happen overtime. Every database manufacturer you know or RDBMS you know creator wants to keep all their people using that their thing. So they they want to add certain things that are special, but they also do have to deal with the fact that everybody is trained in different models and they shouldn't make it. Too hard. So anyway start transaction and we're going to say what are we doing? We're going to delete all records from the park state table and then undo it. So now started, we're just going to do delete. Farm. Two lines 'cause. I really don't like doing it. From. Alright, that's it. That's going to delete every single thing in our whole record. In our whole table, the table is still going to exist. I do want to be clear about that 'cause that's one of these things. It's like having an empty collection. You can have the collection. And not have any values in. Similarly you can have a table. All the columns are defined. All the constraints are defined. Could later add things to it, so it is different deleting all the rows from a table then dropping the tape. Balcony table gets rid of the definition and everything. OK, so let's do select. Count. Prom park. So that's simply going to say what is the total number of stuff that's in there. And now we're going to do a rollback. And rollback says ignore me go all the bad back to the start of the transaction for 10 this never happened. We will not talk in whatever happened in Vegas stays in Vegas. Cat. Actually I added another one of these which says show me what it was before. Show me after it's deleted. Undo that. Let's see if we can see all of this. Now can we see? Yeah, not sure they will actually show it to us very well. OK, let us. well the main point that you got here which is not particularly easy to say if i if i look at the count just laid You know as. Note one. Note. 2. Right. So it's just gonna say Note 3 because I can't get back to the history. Or I don't I I will not say I can't get back to this. I just don't know how to get back to this tree. Maybe there is some way to get back to this. True if somebody figures it out. Great. I'm gonna do this another way, which is. The fact that it all happens at once doesn't mean it I'm now gonna count how many items are in it. It's zero. There are no more entries, no more, no more rows. Then I'm going to new rollback. Which is just going to say rollback doesn't tell you anything about what it did, just it rollback. And now I'm gonna go ahead and look at the account again. I. 167. Part of what I want to show you about the transaction though that is so important to remember is. Once you've done it, it's very hard to know you're in the middle of it. If you're starting to run things, you hit a start transaction. It's just there and then if you magically say either commit or rollback whole bunches of things are going to happen to your live database, you know, and you won't be able to tell the difference. So it's almost like you're dealing with this copy of the database for as long as the transaction goes on. And if you forget to put your committed rollback, all the new things you do are gonna still be part of it. If you then exit out of PG admin and came back in all of a sudden your database is not coming. None of that stuff is going to have happen. I've heard the reason I mentioned that is that if I were to go out and use. Something else? All right, to access the same data. I would not see the changes and as I go I would not see any of the changes until I got to the end and committed a well. If I rollback, I'm not gonna see them anyway. If I committed then I'm going to see the changes right? So with the transaction you have to end with a commit or rollback. And if you don't, you will be very confused about what's going on. It will all seem perfectly normal, it just won't end where you expect. Yeah, we're just gonna do. I'm not gonna. I'm not gonna do that last one, 'cause. It's hard. Hard to do with our context when there's one more of these where you see again. I'm gonna. This is my my warning to you again about the dangers of. Some of what we have is they say, OK, we have this situation, we're going to update all the cities to be in the state of Texas. This can happen really easily because I say, you know, update and then I forgot to put my where clause. So we're going to do our begin transaction. Right. And we're gonna do what do we do? We need to update. Update. City. I wanna set state. VA shun. Pretty Asian. Those taxes? And. City name as you know what? I'm just gonna do this. We're just going to do this because it'll reinforce the thing I was trying to say. We're going to go ahead and begin ~~our~~ transaction. We're gonna update the city and we are going to set this skating revision of every single city to Texas. Right. Update 346 states. Right now, we're going to say, alright, Ruby, let's look at some of those cities. Let's say select star from city. OK. It probably. Didn't highlight it. Current transaction is afforded commands ignored until end of transaction block. Heaven knows what it's trying to do. OK? It wants me to say. You know why it's so? I think I just did it again. Transaction twice by accident. So it complained it so let me go back and let me do my rollback. And will try again. It is pretty much always safe to do a rollback. Rollback says undo something. And so it's pretty much a safe bet to say, alright, I rolled back. Right. Flat. Start Chrome. We have our cities, not just the ones in Ohio, so let's try. That see if it will let us run this life together. So we're gonna update all our cities. We're going to select star from city and see if it lets us do it. OK, everything is in Texas. Which I'm sure some Texans would feel very positively about right now. I'm going to rollback. Right. Because I was still as part of that transaction. Now everything is going to be back where it was. They are in some ways, the transactions are your great saving grace when you start doing things that are more complicated than might adjust a lot of things because you can keep going through them. And if anything fails anywhere during your series of transactions, you know during the course of that transaction. You could say let's roll back and start over. So if you made a mistake, you know 16 lines into your thing, it will not have changed anything. There's 16 statements in. You can deal with that thing that was that is a problem and go back and it will will be OK. It is noon and I think we are. We've done enough for today. I will tell you that the last thing they tell you to do is try to get demonstrate that there are two different if I did a transaction and in the middle of that if I went off and. Query that Postgres database the same Postgres database from mercede be visualized which I've got if anybody wants to stay, I can. I can show you that and you can see it. Do people want to see that, right? Yeah, I'm OK. If anybody wants to go away, go away. You have. No, you have a reading quiz for. For Monday so that stuff I will make sure it is open or whatever, but OK. Anyone wants to see it, I'll just go ahead and try it. We're going to go back and we're going to do exactly the same thing. We're going to begin our transaction. We're going to update the city, Texas. We ban. Can I ask a quick question? Sure. Remember when you first started? I I couldn't really hear when you tried to do the select before you did the rollback. Why wouldn't it let you do it again? I think what happened was I did it begin transaction and then I forgot to select. So it went and tried to start the transaction over again. Oh, OK. And it just got into a confused state. I'm just, that's just a guess. It's just giving me errors. That said you, you're in the middle of a transaction and and it didn't know what to do. So I think what happened is if you try to begin a transaction, then begin it again, it will get into this weird state where it doesn't know what to do. OK, I'm sure there is somebody who knows more about this and knows how you deal with that, but I know that you just at that point when you're confused, you do a rollback so that you like undo, whatever. The weird thing you just did, sometimes what happens is I'll sit here. I think I've selected and I just hit run and it's really going from the very top of my entire file and running absolutely everything up here. Who knows what condition things are in? So OK, that's possibly true. If that is true, we probably have. Some extra cities at the bottom. Just curious if we added some cities. We have Hawkins, Indiana. Didn't we get rid of that? I don't know. No government of Smallville. I don't know. Anyway, I guess maybe we hit a rollback. Alright. Here's where I'm gonna do. I'm gonna go ahead and run these three states. Right. So. I do that. I'm gonna do my select star from city. Hey. Right now I've got everything in Texas. Now I'm gonna start up the visualizer if I can find it again. Amazingly enough. It has a. Connection to the NASD S database as well. Possibly. OK, so now if I go in and I look at. Create. Should we show the data in here? I'm just going to see if I have that or whatever. OK. I'm gonna go into my database. I'm going to. Now I don't remember anything. I don't remember that any anything anymore. OK, so you'll have to give me a minute while I remember now how to get back into here. Let's just do from in here. Is this here in the United States database? One thing I will tell you, I like much more about the visualizer. Is that right up here? It tells you what is mine. In this connection, and is it the sticky database, which means it's the one I'm running on. I really wish that that were true in PGA always tell me which database I'm actually operating. So let's just say select Star City here. Now I have to figure out how to run it again and there we go. And so. I'm dealing with the same database. From a different thing, and I'm not seeing the values change because I have not finished my transaction. Alright so here I am dealing with this shadow version where everything has been changed. Until I do a commit. Nothing is gonna. Commit. If I do a rollback then I will be, which is what I'm going to do right now. I'm going to. probably i don't want