Theo Musa – Personal Reflection

Spend 10 hours on the questions on my 6 questions (1, 6, 11, 16, 21, 26). My dyslexia is probably the reason for why I may have spent longer than average on all of the questions but also at the time of completion I was still familiarising myself with SQL. I think in future my answers may have been different knowing all the knowledge I know now.

For Question 1, I used SELECT DISTINCT. In hindsight for the nature of this question – querying names of employees – a record that is likely to only appear once, DISTINCT seems a little unnecessary – as well as in the case if there is 2 records – we would probably want to see both as the implication for this field is that there are 2 employees with the same name, rather than the same employees appearing twice which we would want to use the DISTINCT command for. I would also in future include the ORDER BY command to alphabetically order users to ease of users reading through the data.

For Question 6, I did not use table names or aliases, and whilst the program works it would have been clearer if each column directly said which table it was referring from as this can help solve future errors if the tables names changed to where there was an overlapping column name. I would in future also change the searching to be more intuitively allowing users to search by the FirstName & LastName fields is they want to see if any specific user has however many invoices, or allow the user to search invoices via a date, as these pieces of information are more likely to be queried than the specific invoice id number which may easily be lost or forgotten.

For Question 11, my spacing does not align with SELECT DISTINCT – and whilst this does not inherently affect the running of the code, it would be better notation to change this to be inline with the others. It might also be useful to include their customer id as this could help for querying through the rest of the database – or act as a username almost. Again I used the SELECT DISTINCT command and whilst this does not affect the running of my program, in this scenario it would be useful to not include this as this then would not exclude customers with identical names, though this problem would also be negated if we included customer id's due to them being distinct in nature. I would also choose ascending order as this would go through A to Z rather than Z to A with Z representing a larger value in case of the alphabet, as this might be more intuitive when we talk about sorting items alphabetically.

For Question 16, I would use in future the ORDER BY command to make the result more readable for the user, and preferably I would sort by the date hired, due to the date here seeming to be the factor we care most about here we are limiting employees we are searching through by the year they joined the company. I would also not use wildcard % on both sides of 2002, I would use 2002% due to the fact that in the DATETIME notation I believe the year is at the start of the string, and so it is unneeded to search through the whole string when only the start is needed for the LIKE operation which may improve performance slightly. I am glad I renamed the table as Date, as this make the result more clear for the user .

For Question 21, I would use in future aliases in my statement to improve the readability along with reducing the chances of an error due to less manual writing. I would also in future change the ORDER BY to be based on number of albums produced as that seems more useful to the user than sorting by an abstract ArtistID, especially since we are selecting files based on if they go over the threshold of 10 albums. I believe using INNER JOIN was the correct move here due to the fact that this query is useless if we cannot see both the artist's name to see which artist produced over 10 as well as the number of albums to see if it meets the threshold of 10 albums, which is why I believe the INNER JOIN is more logical than a left or right join or any other.

For my final question that I did, 26, I would in future use aliases to reduce the chances of a mistake in typing as well as improving the readability of my code. I also believe doing a INNER JOIN was the correct move here as if we could not recollect the track, this would be pretty pointless as a playlist not knowing what song, or if we wanted to use a play function in future, not being able to play it, hence meaning records linking to both tables are essential.