Q: What are the reasons why some customers received corrupt tracks when they placed their orders?

My approach to this question was as follows

Firstly I need to know more about the corrupt tracks and the orders that had the corrupt tracks? Can I identify these orders? Can I identify these corrupt tracks?
Once identified, can I spot any patterns to do with the corrupt tracks, the orders or the customers? For example, could it have anything to do with the bytes = e.g. 0? Or mediatype of the track vs media type of the customer?

Q1 - Are there any tracks with bytes = 0, or milliseconds = 0 which would indicate a corrupt file?

To answer this question I need to create a table which searches for any tracks with bytes - 0 using the following syntax:

```
SELECT *
FROM public."Track"
WHERE "Bytes" = 0
ORDER BY "Bytes" ASC;
```

Then did the same for Milliseconds by changing the word “Bytes” for “ “Milliseconds” in the syntax.

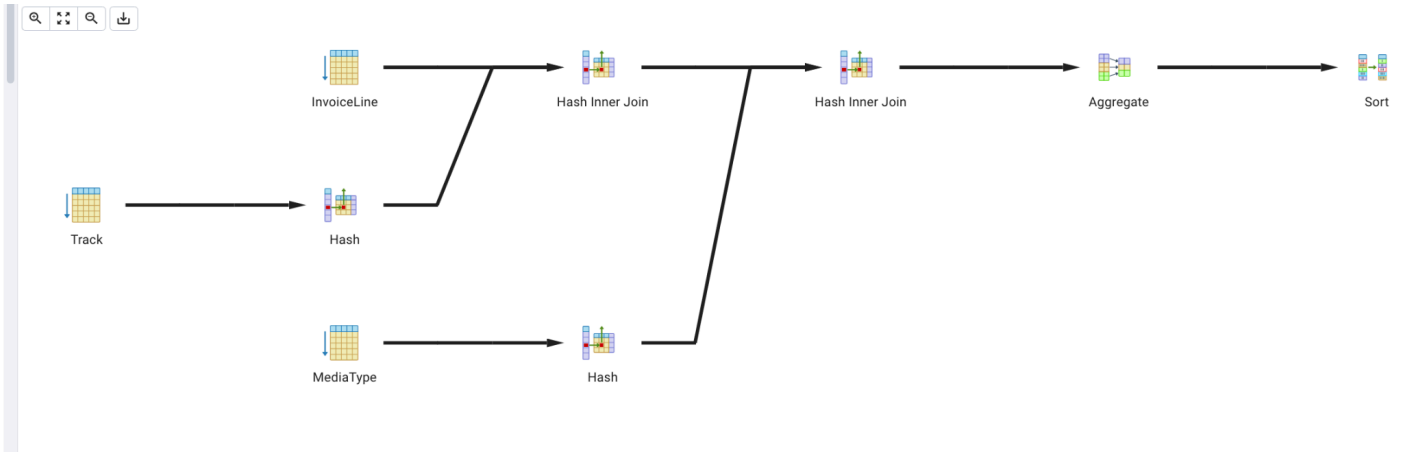
This came up with no results so suggested its not to do with the bytes or Milliseconds

Q2 - how many orders were made for each type of media file? Are there any media types that could indicate a corrupt file? Can I establish how many orders are made for tracks from each type of media file? A low number of orders from a particular media type may indicate it could be the source of a corrupt file. It would also be helpful to understand the different types of media type file.

I used this query to return the count of invoices grouped by each media type shown as No_of_Orders, grouped by each media type), ordered by the invoice count in descending order.

I used a left hand join to link orders (invoiceid) to tracks and then to media types. I also used the aggregate function COUNT to only give me the total number of orders by media type.

```
SELECT mt."MediaTypeId", mt."Name" AS Media_Type,
COUNT (il."InvoiceId") AS No_of_orders
FROM public."MediaType" mt
JOIN public."Track" t USING ("MediaTypeId")
JOIN public."InvoiceLine" il USING ("TrackId")
GROUP BY mt."MediaTypeId", mt."Name"
ORDER BY No_of_orders DESC;
```




```
ORDER BY mt."Name" DESC;

--Q3 - how many orders were made for each type of media file?

SELECT mt."MediaTypeId", mt."Name" AS Media_Type,
COUNT (il."InvoiceId") AS No_of_orders
FROM public."MediaType" mt
JOIN public."Track" t USING ("MediaTypeId")
JOIN public."InvoiceLine" il USING ("TrackId")
GROUP BY mt."MediaTypeId", mt."Name"
ORDER BY No_of_orders DESC;
```

Data Output | Messages | Notifications

	MediaTypeId [PK] integer	media_type character varying (120)	no_of_orders bigint
1	1	MPEG audio file	1976
2	2	Protected AAC audio file	146
3	3	Protected MPEG-4 video file	111
4	4	Purchased AAC audio file	4
5	5	AAC audio file	3

There are very few orders under media types 4 & 5 (which is a media type that did not appear on the earlier list of mediatypes) - my next step would be to look at all these orders in more detail, with details of the customers who ordered them to establish if these were the corrupt files. It could be that they are not protected files and therefore liable to being corrupt.

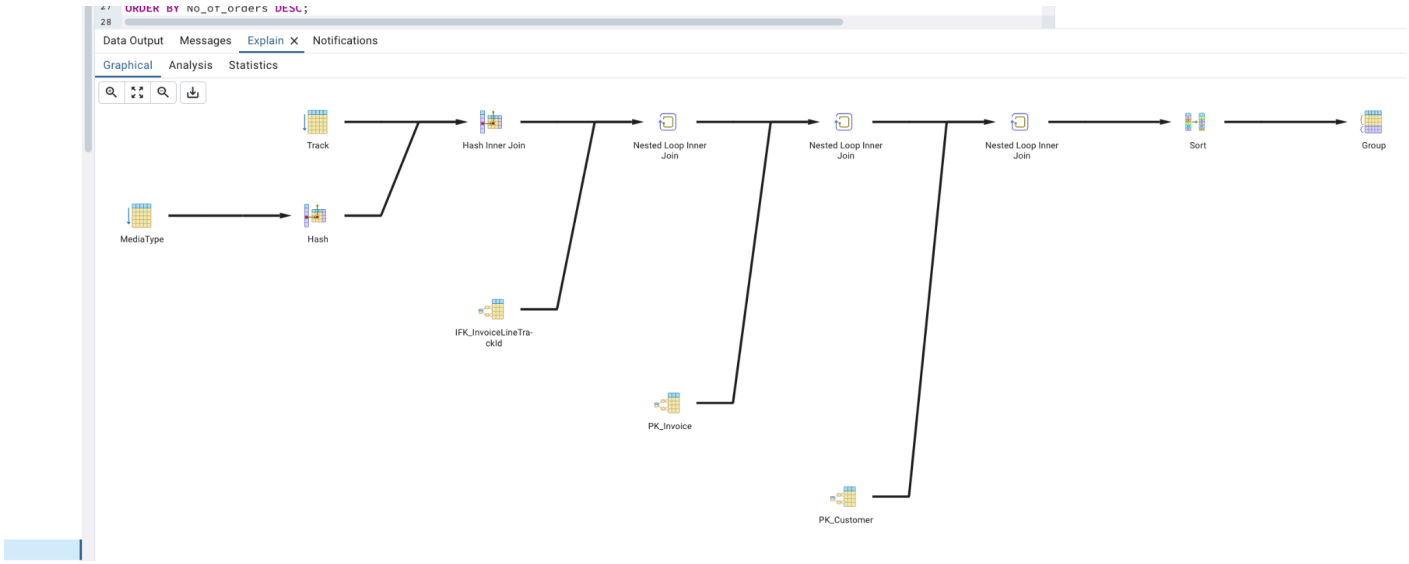
Q3) show all orders that included media types 4 & 5, including customer name, country, phone number track names and invoice id's

I used the following syntax to show all orders that included media types 4 & 5, track names, invoice id's and customer details such as name, country, phone number and email.

I used AS to distinguish between the two name categories, and aliases and differentiates the tables. I used the IN syntax to filter for media types 4 & 5 only.

SELECT mt."MediaTypeId", mt."Name" AS Media_Type, t."Name" AS Track, il."InvoiceId", i."CustomerId", CONCAT(c."FirstName",' ',c."LastName"),c."Country",c."Phone",c."Email"
FROM public."MediaType" mt

JOIN public."Track" t USING ("MediaTypeId")
JOIN public."InvoiceLine" il USING ("TrackId")
JOIN public."Invoice" i USING ("InvoiceId")
JOIN public."Customer" c USING ("CustomerId")
WHERE mt."MediaTypeId" IN('4','5')
GROUP BY mt."MediaTypeId", mt."Name", t."Name", il."InvoiceId", i."CustomerId",
CONCAT(c."FirstName",' ',c."LastName"),c."Country",c."Phone",c."Email"
ORDER BY "MediaTypeId" DESC;




```
27 ORDER BY No_of_orders DESC;
28
29 --Q4 - show all orders that included media types 4 & 5, including customer name, country, phone number track names and invoice id's
30
31 SELECT mt."MediaTypeId", mt."Name" AS Media_Type, t."Name" AS Track, il."InvoiceId", i."CustomerId", CONCAT(c."FirstName",' ',c."LastNa
32 FROM public."MediaType" mt
33 JOIN public."Track" t USING ("MediaTypeId")
34 JOIN public."InvoiceLine" il USING ("TrackId")
35 JOIN public."Invoice" i USING ("InvoiceId")
36 JOIN public."Customer" c USING ("CustomerId")
37 WHERE mt."MediaTypeId" IN('4','5')
38 GROUP BY mt."MediaTypeId", mt."Name", t."Name", il."InvoiceId", i."CustomerId", CONCAT(c."FirstName",' ',c."LastName"),c."Country",c."Pho
39 ORDER BY "MediaTypeId" DESC;
40
41
42
43 SELECT ar."ArtistId", ar."Name", al."Title",
```

Data Output | Messages | Notifications

	MediaTypeId integer	media_type character varying (120)	track character varying (200)	InvoiceId integer	CustomerId integer	concat text	Country character varying (40)	Phone character varying (24)	Email character varying (60)
1	5	AAC audio file	Din Din Wo (Little Child)	208	4	Bjørn Hansen	Norway	+47 22 44 22 22	bjorn.hansen@yahoo.no
2	5	AAC audio file	Love Comes	313	43	Isabelle Mercier	France	+33 03 80 73 66 99	isabelle_mercier@apple.fr
3	5	AAC audio file	Muita Bobeira	103	24	Frank Ralston	USA	+1 (312) 332-3232	fralston@gmail.com
4	4	Purchased AAC audio file	Prometheus Overture, Op. 43	105	39	Camille Bernard	France	+33 01 49 70 65 65	camille.bernard@yahoo.fr
5	4	Purchased AAC audio file	Sonata for Solo Violin: IV: Presto	105	39	Camille Bernard	France	+33 01 49 70 65 65	camille.bernard@yahoo.fr
6	4	Purchased AAC audio file	Symphony No. 104 in D Major "London": IV. Finale: Spirito...	208	4	Bjørn Hansen	Norway	+47 22 44 22 22	bjorn.hansen@yahoo.no
7	4	Purchased AAC audio file	Étude 1, In C Major - Preludio (Presto) - Liszt	108	47	Lucas Mancini	Italy	+39 06 39733434	lucas.mancini@yahoo.it

I would suggest the corrupt files were due to not being protected media_type files. A solution would be to test all media types 4 & 5 (7 tracks in total) to see if they were corrupted. CustomerId 4 could be contacted to identify if one or both of the files he purchased (4 & 5) were corrupt which would identify which media types were the problem.

Observations for writing better syntax

1) It’s helpful to write out the question you’re trying to answer first before attempting the syntax
2) Using the question then map out which tables and columns would be required to answer it from the ERD
3) Remember that all columns listed under SELECT also need to be listed in the GROUP BY line
4) The order of the JOINS written in the syntax should follow the order of the relationship flow between the tables - its helpful to write this flow down first before attempting the syntax.
5) Using ‘AS’ and aliases are very useful to distinguish between different tables and rows with the same name.
6) Use the expand function in the query tool to double check the join of tables is set up as you expected it to be.
7) Use the IN function if you want to filter by more than one attribute.