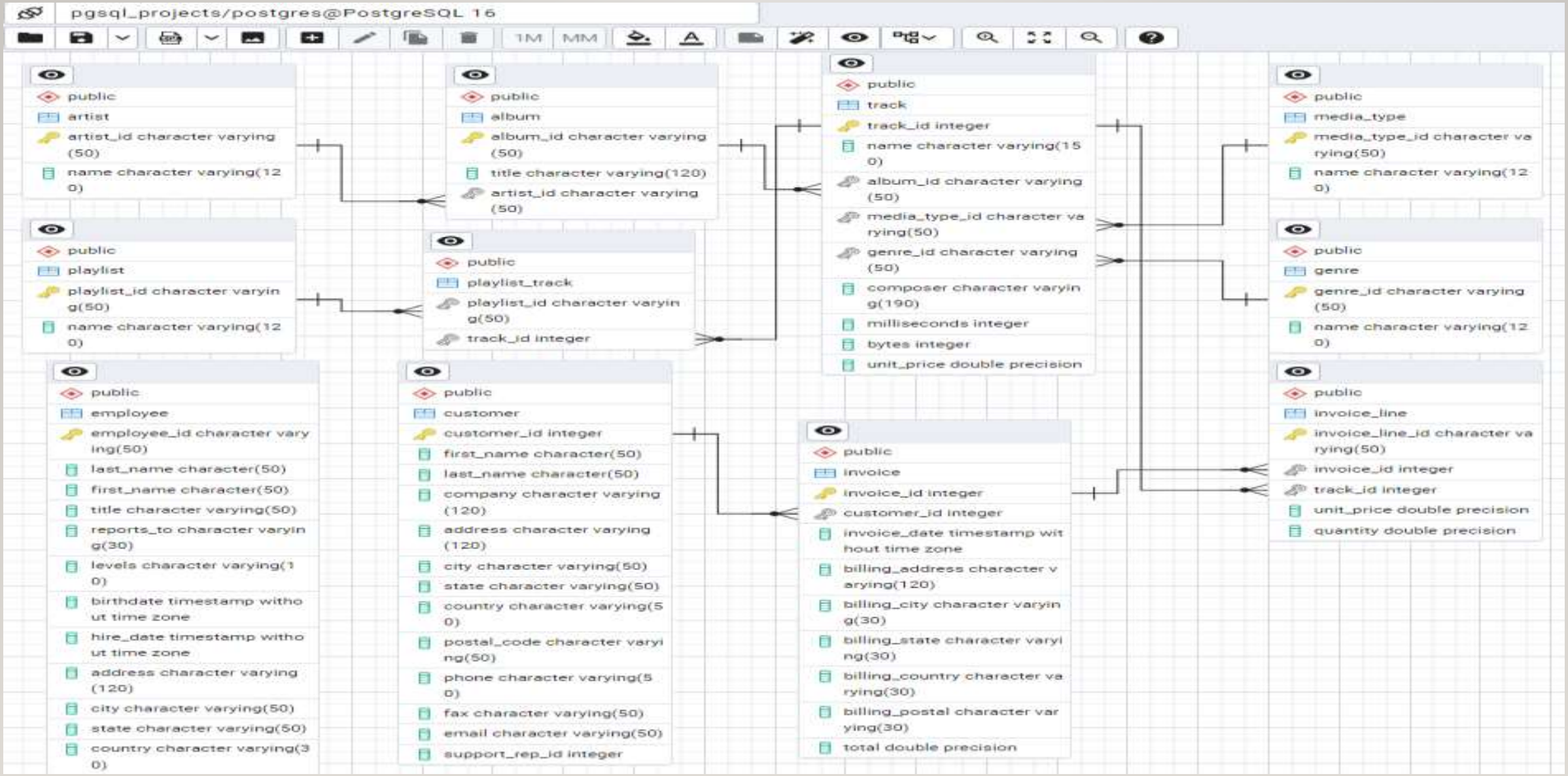
SQL PROJECT

MUSIC STORE DATA ANALYSIS

DATA MODEL


pgsql_projects/postgres@PostgreSQL 16
public
artist
artist_id character varying (50)
name character varying(120)
public
album
album_id character varying (50)
title character varying(120)
artist_id character varying (50)
public
track
track_id integer
name character varying(150)
album_id character varying (50)
media_type_id character varying(50)
genre_id character varying (50)
composer character varying(190)
milliseconds integer
bytes integer
unit_price double precision
public
media_type
media_type_id character varying(50)
name character varying(120)
public
playlist
playlist_id character varying(50)
name character varying(120)
public
playlist_track
playlist_id character varying(50)
track_id integer
public
genre
genre_id character varying (50)
name character varying(120)
public
employee
employee_id character varying(50)
last_name character(50)
first_name character(50)
title character varying(50)
reports_to character varying(30)
levels character varying(10)
birthdate timestamp without time zone
hire_date timestamp without time zone
address character varying (120)
city character varying(50)
state character varying(50)
country character varying(30)
public
customer
customer_id integer
first_name character(50)
last_name character(50)
company character varying (120)
address character varying (120)
city character varying(50)
state character varying(50)
country character varying(50)
postal_code character varying(50)
phone character varying(50)
fax character varying(50)
email character varying(50)
support_rep_id integer
public
invoice
invoice_id integer
customer_id integer
invoice_date timestamp without time zone
billing_address character varying(120)
billing_city character varying(30)
billing_state character varying(30)
billing_country character varying(30)
billing_postal character varying(30)
total double precision
public
invoice_line
invoice_line_id character varying(50)
invoice_id integer
track_id integer
unit_price double precision
quantity double precision

1. WHO IS SENIOR MOST EMPLOYEE BASED ON JOB TITLE ?
2. WHICH COUNTRIES HAVE THE MOST INVOICES ?

Query | Query History

```
1  --Q1. who is senior most employee based on job title?
2
3  select employee_id,first_name,last_name
4  from employee
5  order by levels desc
6  limit 1
7
```

Data Output | Messages | Notifications

	employee_id [PK] character varying (50)	first_name character (50)	last_name character (50)
1	9	Mohan	Madan

Query | Query History

```
7
8  --Q2. which countries  have the most invoices?
9
10 select count(invoice_id) as c,billing_country
11 from invoice
12 group by billing_country
13 order by c desc
```

Data Output | Messages | Notifications

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland
14	10	Australia
15	10	Netherlands

1. WHAT ARE TOP3 VALUES OF THE TOTAL INVOICE ?
2. WHICH CITY HAS THE BEST CUSTOMER. WRITE A QUERY THAT RETURN ONE CITY THAT HAS THE HIGHEST SUM OF INVOICE TOTALS. RETURN BOTH THE CITY NAME AND SUM OF ALL INVOICE TOTALS ?

Query | Query History

```
14
15  --Q3. what are top3 values of the total invoice?
16
17  select total
18  from invoice
19  order by total  desc limit 3
20
```

Data Output | Messages | Notifications

	total double precision
1	23.759999999999998
2	19.8
3	19.8

```
21  --Q4. which city has the best customer.
22
23  select billing_city, sum(total) as a
24  from invoice
25  group by billing_city
26  order by a desc
27  limit 1
```

Data Output | Messages | Notifications

	billing_city character varying (30)	a double precision
1	Prague	273.24000000000007

1. WHO IS THE BEST CUSTOMER ? THE CUSTOMER WHO HAS SPENT THE MOST MONEY WILL BE DECLARED THE BEST CUSTOMER. WHO HAS SPENT MOST MONEY ?

2. WRITE QUERY TO RETURN THE EMAIL, FIRST NAME, LAST NAME, & GENRE OF ALL ROCK MUSIC LISTENERS. RETURN YOUR LIST ORDERED ALPHABETICALLY BY EMAIL STARTING WITH 'A' ?

Query | Query History

```
32  select c.customer_id,c.first_name,
33  sum(i.total) as expense
34  from customer c
35  join invoice i on c.customer_id=i.customer_id
36  group by c.customer_id
37  order by expense desc
38  limit 1
```

Data Output | Messages | Notifications

	customer_id [PK] integer	first_name character (50)	expense double precision
1	5	R	144.54000000000002

```
43  --1.using subquery only
44  select distinct Email, first_name,last_name
45  from customer
46  where customer_id in(
47                       select customer_id from invoice where invoice_id in(
48                       select invoice_id from invoice_line where track_id in(
49                       select track_id from track where genre_id=
50                       (select genre_id  from genre where name='Rock'))))
```

Data Output | Messages | Notifications

	email character varying (50)	first_name character (50)	last_name character (50)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan

1. LET'S INVITE THE ARTISTS WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR DATASET. WRITE A QUERY THAT RETURNS THE ARTIST NAME AND TOTAL TRACK COUNT OF THE TOP 10 ROCK BANDS ?

```
select artist.name,count(artist.artist_id) as total_track_count
from artist
join album on artist.artist_id=album.artist_id
join track on album.album_id=track.album_id
join genre on track.genre_id=genre.genre_id
where genre.name like 'Rock'
group by artist.artist_id
order by total_track_count desc
limit 10;
```

	name character varying (120)	total_track_count bigint
1	Led Zeppelin	114
2	U2	112
3	Deep Purple	92
4	Iron Maiden	81
5	Pearl Jam	54
6	Van Halen	52
7	Queen	45
8	The Rolling Stones	41
9	Creedence Clearwater Revival	40
10	Kiss	35

1. RETURN ALL THE TRACK NAMES THAT HAVE A SONG LENGTH LONGER THAN THE AVERAGE SONG LENGTH. RETURN THE NAME AND MILLISECONDS FOR EACH TRACK. ORDER BY THE SONG LENGTH WITH THE LONGEST SONGS LISTED FIRST

```
select name,milliseconds,bytes
from track
where milliseconds >(select avg(milliseconds) from track)
order by milliseconds desc
```

	name character varying (150)	milliseconds integer	bytes integer
1	Occupation / Precipice	5286953	1054423946
2	Through a Looking Glass	5088838	1059546140
3	Greetings from Earth, Pt. 1	2960293	536824558
4	The Man With Nine Lives	2956998	577829804
5	Battlestar Galactica, Pt. 2	2956081	521387924
6	Battlestar Galactica, Pt. 1	2952702	541359437
7	Murder On the Rising Star	2935894	551759986
8	Battlestar Galactica, Pt. 3	2927802	554509033
9	Take the Celestra	2927677	512381289
10	Fire In Space	2926593	536784757
11	The Long Patrol	2925008	513122217
12	The Magnificent Warriors	2924716	570152232
13	The Living Legend, Pt. 1	2924507	503641007

1. FIND HOW MUCH AMOUNT SPENT BY EACH CUSTOMER ON ARTISTS? WRITE A QUERY TO RETURN CUSTOMER NAME, ARTIST NAME AND TOTAL SPENT

```sql
with best_artist as (
     select artist.artist_id as artist_id,artist.name as name_of_artist,
     sum(invoice_line.unit_price*invoice_line.quantity) as exps
     from invoice_line
     join track on invoice_line.track_id=track.track_id
     join album on track.album_id=album.album_id
     join artist on album.artist_id=artist.artist_id
     group by 1
     order by 3 desc
     limit 1

)

select c.customer_id, c.first_name, c.last_name, ba.name_of_artist as artist_nam,
sum(il.unit_price*il.quantity)as total_exp
from customer c
join invoice i on c.customer_id=i.customer_id
join invoice_line il on i.invoice_id=il.invoice_id
join  track t on il.track_id=t.track_id
join album a on t.album_id=a.album_id
join best_artist ba on a.artist_id=ba.artist_id
group by c.customer_id,c.first_name,c.last_name,ba.name_of_artist
order by total_exp desc
```

	customer_id integer	first_name character (50)	last_name character (50)	artist_nam character varying (120)	total_exp double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96
10	5	R	Madhav	Queen	3.96
11	23	John	Gordon	Queen	2.9699999999999998
12	54	Steve	Murray	Queen	2.9699999999999998
13	31	Martha	Silk	Queen	2.9699999999999998
14	16	Frank	Harris	Queen	1.98
15	17	Jack	Smith	Queen	1.98
16	24	Frank	Ralston	Queen	1.98
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98
19	36	Hannah	Schneider	Queen	1.98

Total rows: 43 of 43 Query complete 00:00:00.433 Ln 94, Col 1

1. WE WANT TO FIND OUT THE MOST POPULAR MUSIC GENRE FOR EACH COUNTRY. WE DETERMINE THE MOST POPULAR GENRE AS THE GENRE WITH THE HIGHEST AMOUNT OF PURCHASES. WRITE A QUERY THAT RETURNS EACH COUNTRY ALONG WITH THE TOP GENRE. FOR COUNTRIES WHERE THE MAXIMUM NUMBER OF PURCHASES IS

SHARED RETURN ALL GENRES.

```
WITH popular_genre AS
(
    SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
    ROW_NUMBER() OVER(PARTITION BY customer.country
    ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
    FROM invoice_line
    JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
    JOIN customer ON customer.customer_id = invoice.customer_id
    JOIN track ON track.track_id = invoice_line.track_id
    JOIN genre ON genre.genre_id = track.genre_id
    GROUP BY 2,3,4
    ORDER BY 2 ASC, 1 DESC
)
```

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1

Total rows: 24 of 24 Query complete 00:00:00.313 Ln 131, Col 2