WELUP DIGITAL MODULE 5 ASSESSMENT

NAME: Adanna Opara

STUDENT ID: WDS007

MODULE: 5 (Data Analytics with SQL)

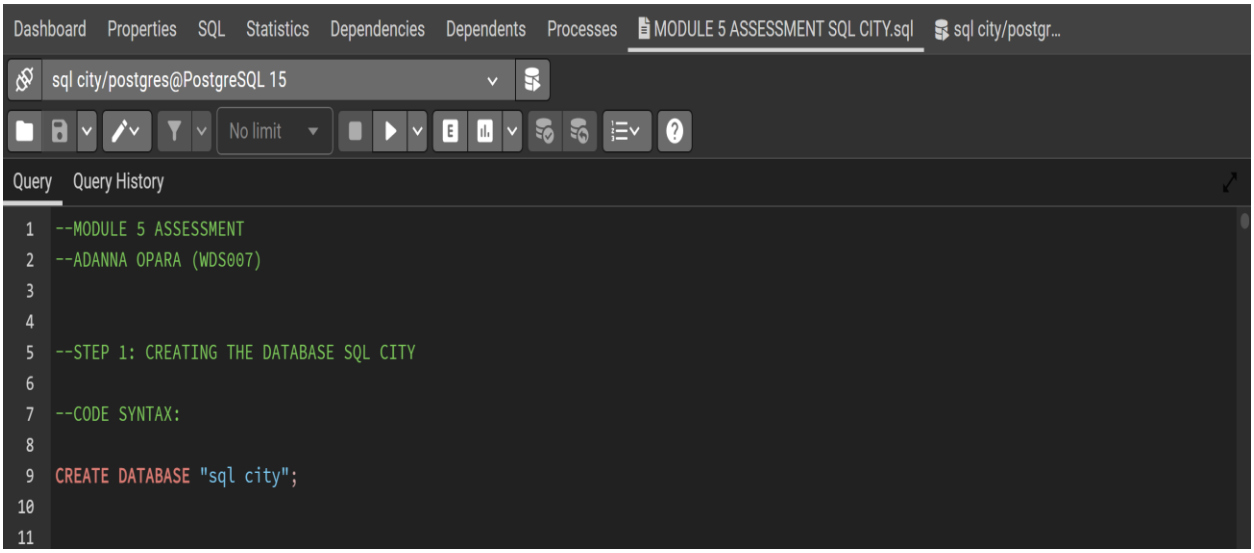
PROCESS:

QUESTION 1:

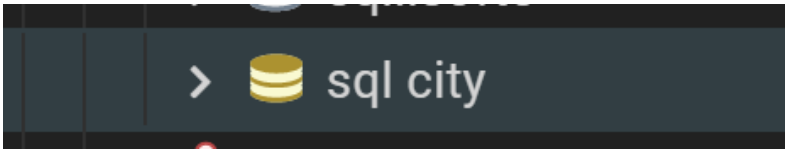
As the community database analyst, you have been approached to perform some tasks. A fundraising committee has been established. Their task is to approach wealthy members of the community for donations towards the annual local festival. You are to help them generate a table of the top 25 highest-earning individuals including their Name, address_number, address_street_name, and annual income.

SOLUTION:

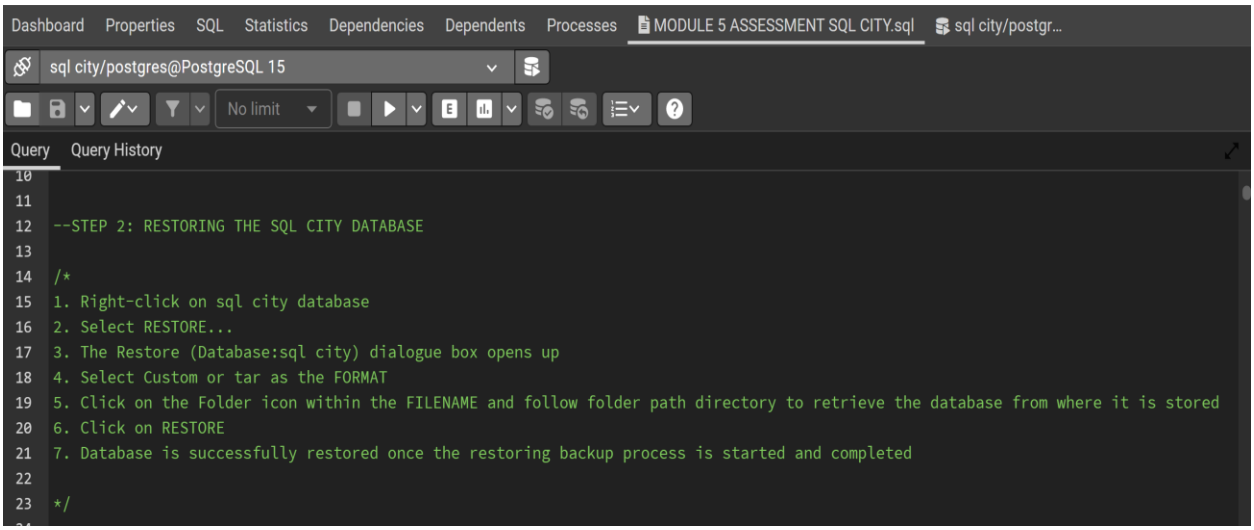
STEP1: CREATING THE DATABASE "sql city"



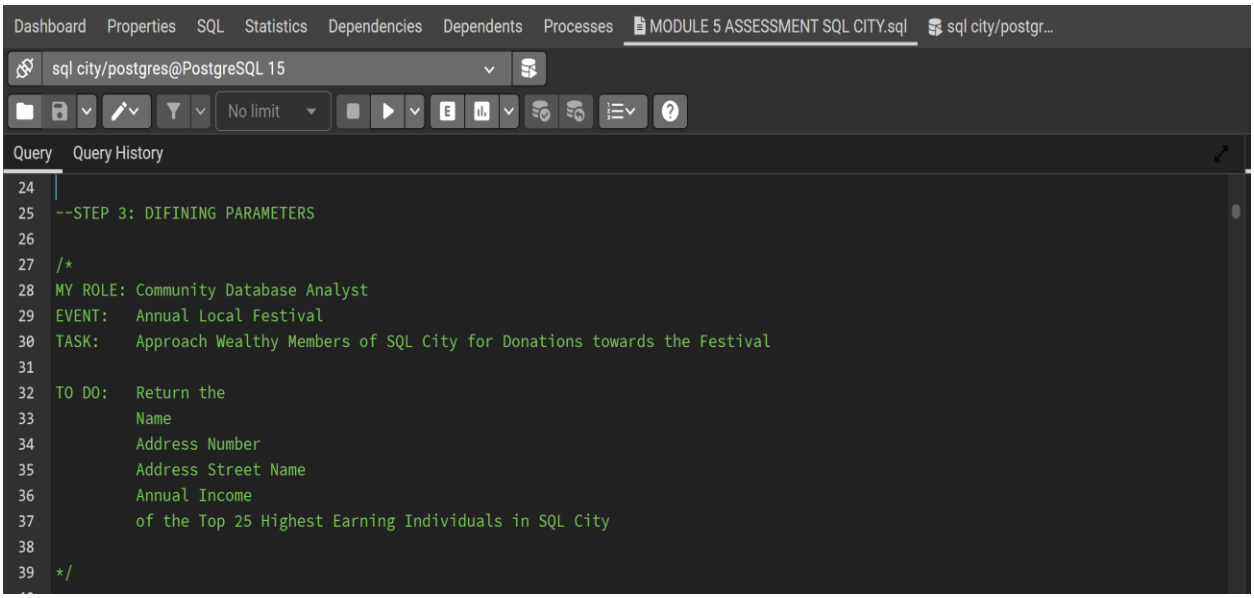
"SQL City" Datebase



STEP 2: RESTORING A BACKUP COPY OF THE DATABASE

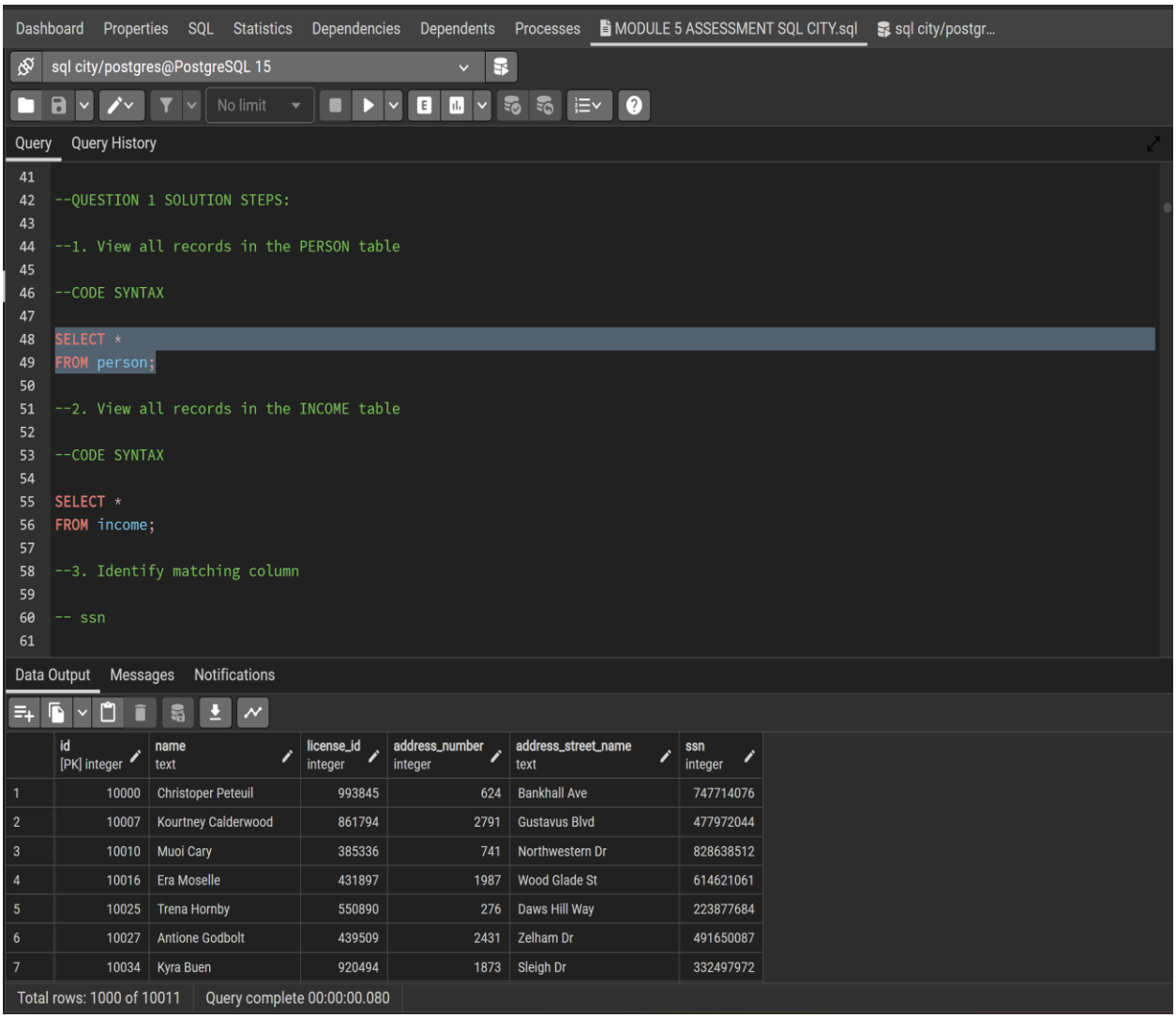


STEP 3: DEFINING PARAMETERS

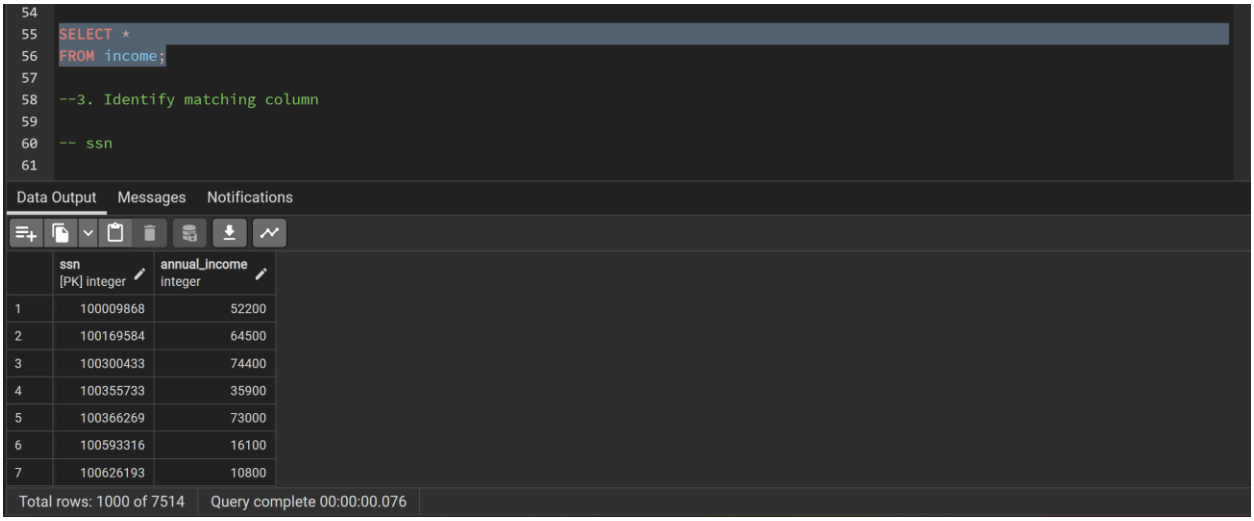


STEP 4: QUESTION 1 SOLUTION STEPS

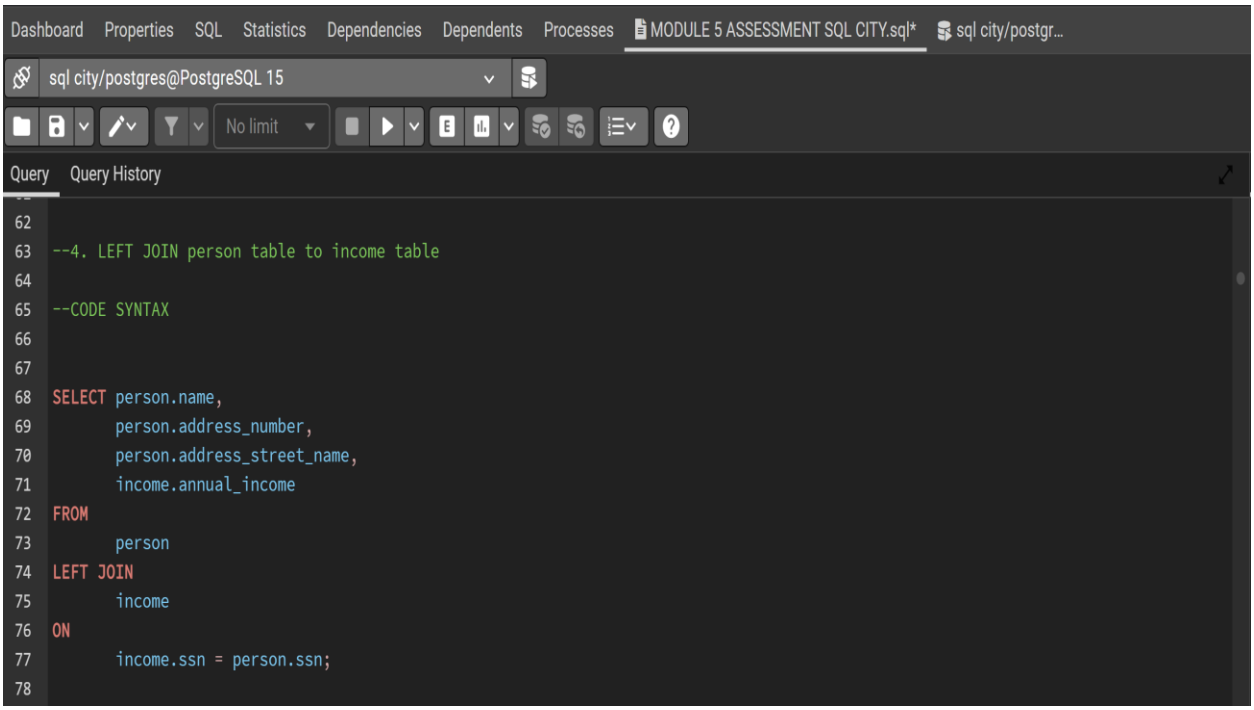
Step 4a: View all records in the "Person" Table



Step 4b: View all records in the "Income" Table and Identify matching columns in both tables

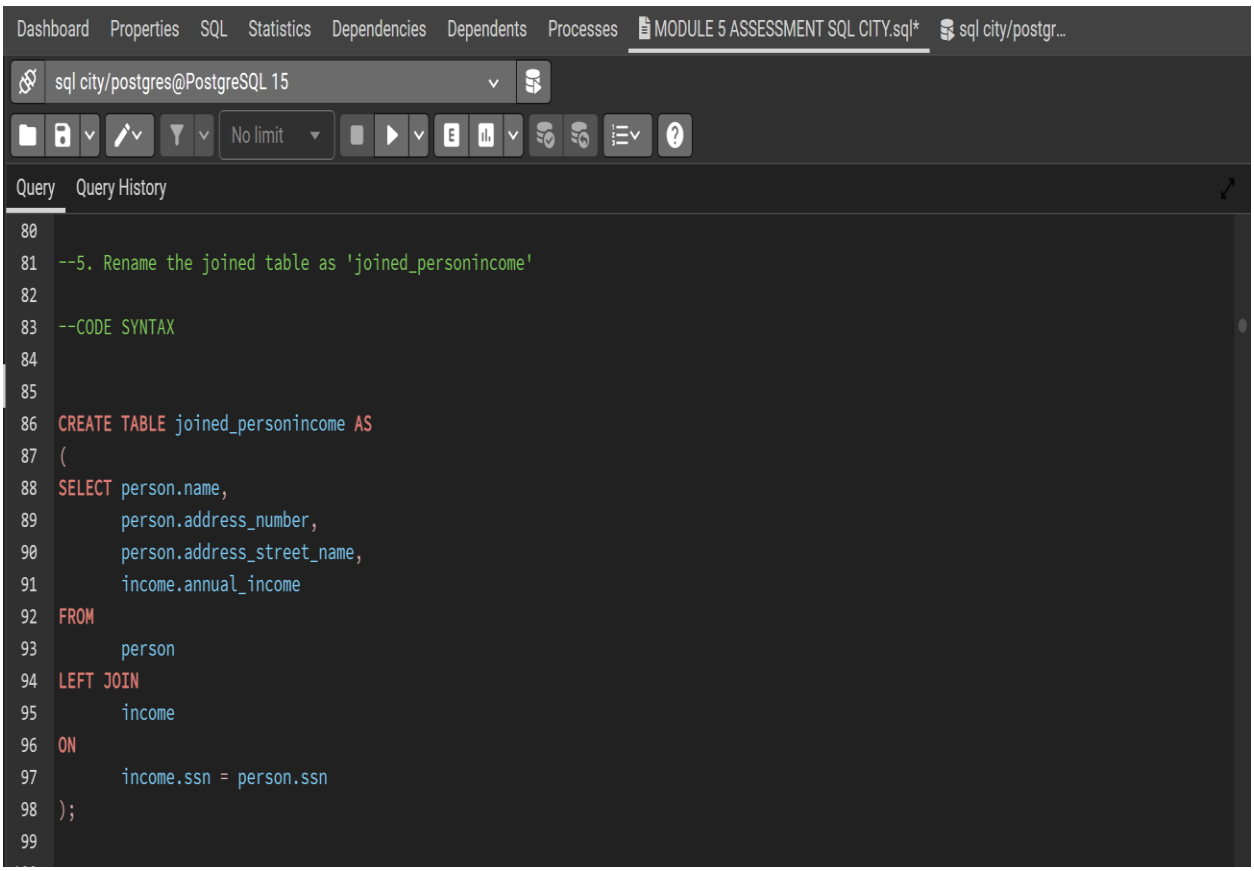


Step 4c: LEFT JOIN “person” table to “income” table



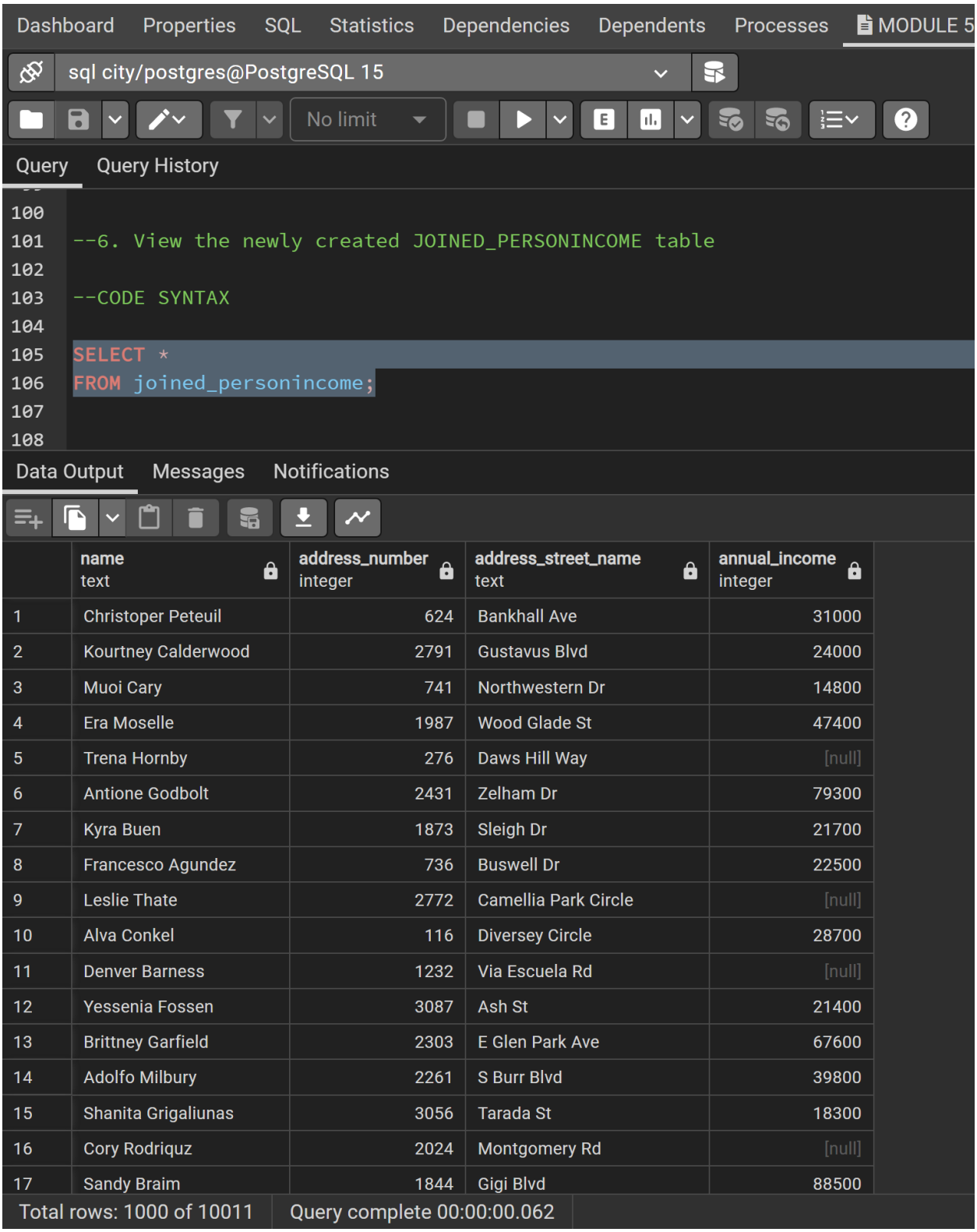

```
62
63  --4. LEFT JOIN person table to income table
64
65  --CODE SYNTAX
66
67
68  SELECT person.name,
69         person.address_number,
70         person.address_street_name,
71         income.annual_income
72  FROM
73         person
74  LEFT JOIN
75         income
76  ON
77         income.ssn = person.ssn;
78
```

Step 4d: Rename joined table as “joined_personincome”




```
80
81  --5. Rename the joined table as 'joined_personincome'
82
83  --CODE SYNTAX
84
85
86  CREATE TABLE joined_personincome AS
87  (
88  SELECT person.name,
89         person.address_number,
90         person.address_street_name,
91         income.annual_income
92  FROM
93         person
94  LEFT JOIN
95         income
96  ON
97         income.ssn = person.ssn
98  );
99
```

Step 4e: View the newly created joined table



Step 4f: Order “Annual income” table in descending order & LIMIT to top 25 individuals

Dashboard Properties SQL Statistics Dependencies Dependents Processes MODULE 5 ASSESSMENT SQL CITY

sql city/postgres@PostgreSQL 15

No limit

Query Query History

```
108
109 --7. Order the ANNUAL INCOME in descending order and limit to top 25 individuals
110
111 --CODE SYNTAX
112
113 SELECT *
114 FROM joined_personincome
115 ORDER BY annual_income desc
116 LIMIT 25;
117
```

TOP 25 Wealthy Individuals in SQL City

Dashboard Properties SQL Statistics Dependencies Dependents Processes

sql city/postgres@PostgreSQL 15

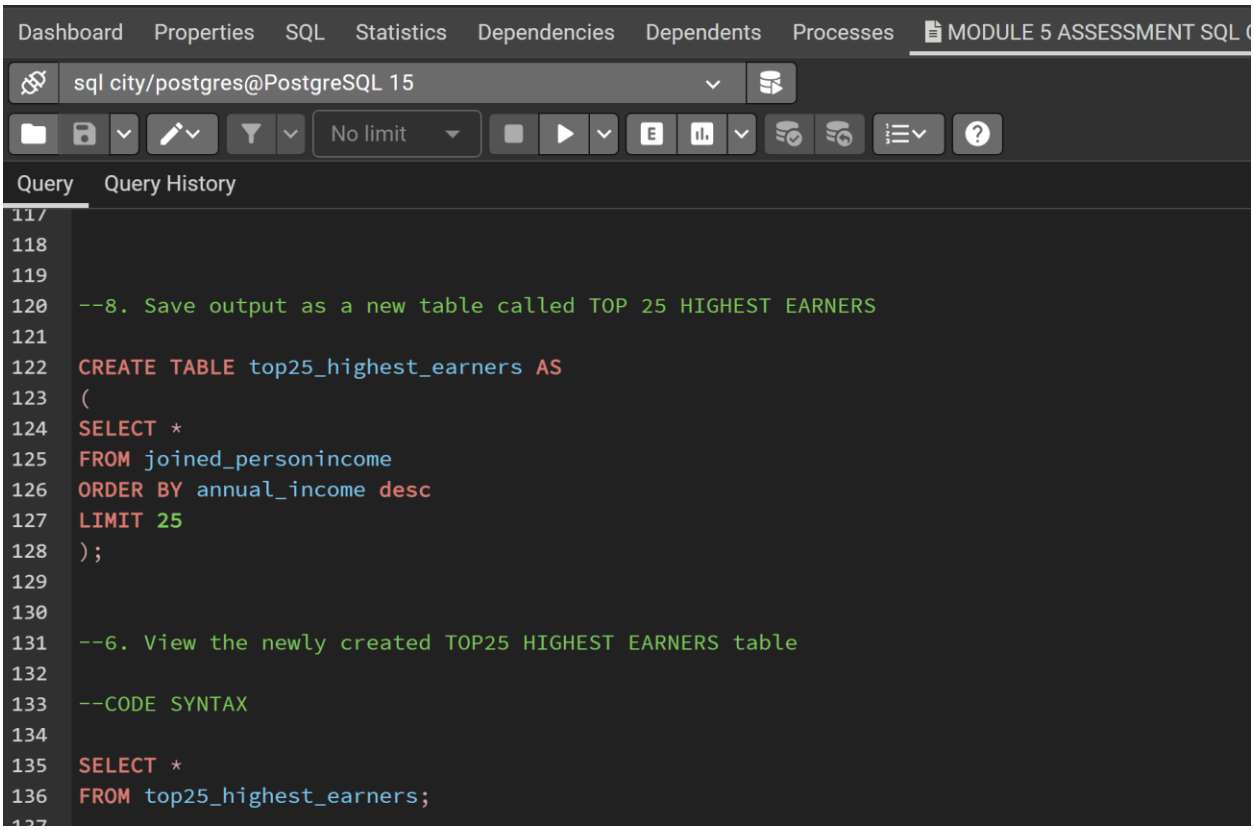
No limit

Data Output Messages Notifications

	name text	address_number integer	address_street_name text	annual_income integer
1	Cayla Dittberner	3385	Uralla Ave	[null]
2	Sherron Mckiernan	1275	Deborah Lee St	[null]
3	Lawrence Angiolillo	2895	Cameron Pond Ave	[null]
4	Mariann Tremaine	2833	Rahlves St	[null]
5	Bradly Goodness	1183	Mina St	[null]
6	Gerald Reil	2456	Watervale Ave	[null]
7	Vernia Eshlerman	3004	Endon Circle	[null]
8	Adriane Pelligra	948	Emba Ave	[null]
9	Otto Bednar	2098	Golden West Way	[null]
10	Kory Beauliev	2532	Charles Mary Dr	[null]
11	Jami Freiberger	3376	Diesel Way	[null]
12	Davina Gangwer	1360	Thorby Circle	[null]
13	Nisha Thielman	3812	Litho Ave	[null]
14	Rogelio Gundersen	75	West Oakdale Rd	[null]
15	Cory Rodriquz	2024	Montgomery Rd	[null]
16	Milford Digsby	2435	Borick Rd	[null]
17	Leslie Thate	2772	Camellia Park Circle	[null]
18	Brad Bowthorpe	3932	Valiant Way	[null]
19	Trena Hornby	276	Daws Hill Way	[null]
20	Hortense Schleich	290	Inchbonnie Circle	[null]
21	Denver Barness	1232	Via Escuela Rd	[null]
22	Manda Steady	696	Almaden Blvd	[null]
23	Britt List	923	Marbury Rd	[null]
24	Sally Claborn	2621	North Croydon Dr	[null]
25	Latoyia Syta	3177	Mount Olive St	[null]

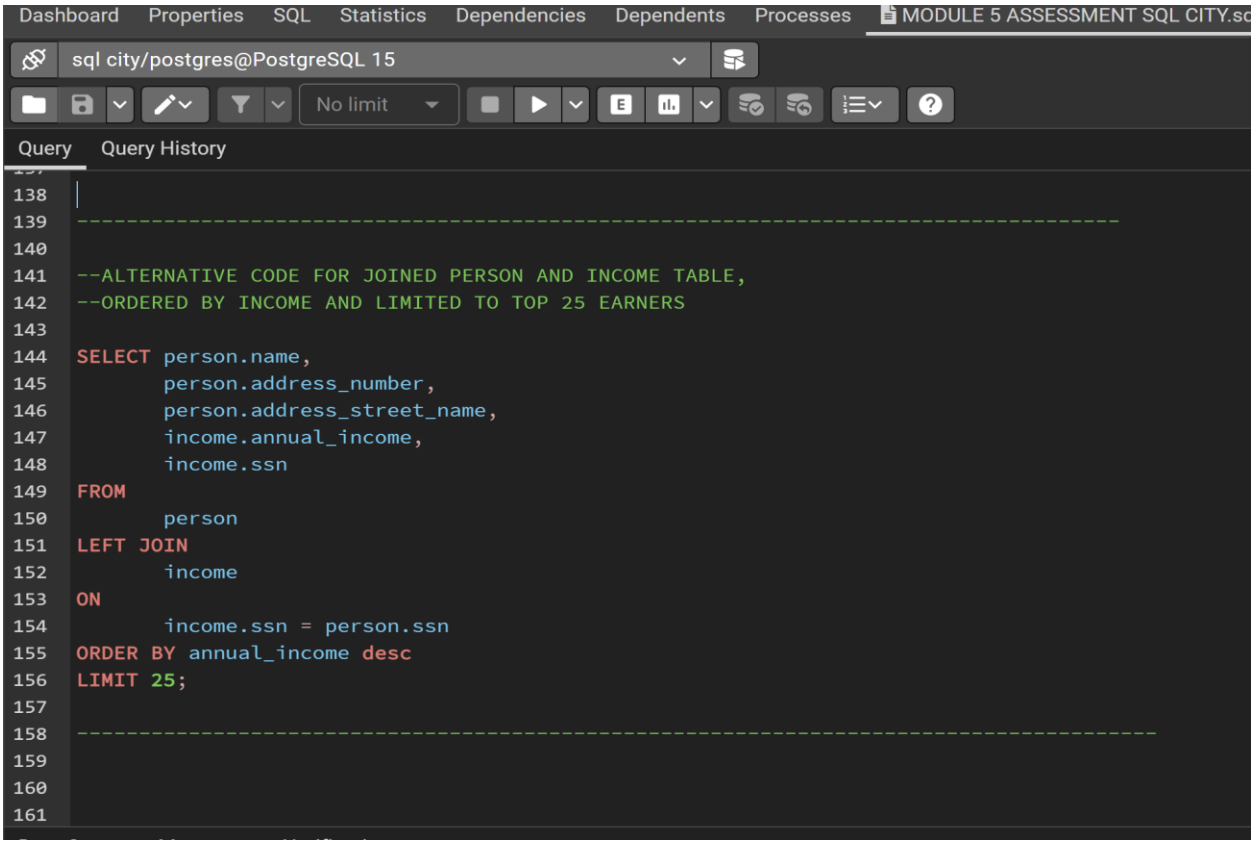
Total rows: 25 of 25 Query complete 00:00:00.066

Step 4g: Save output as TOP 25 HIGHEST EARNERS AND VIEW NEW TABLE



Note: New table (top25_highest_earners) is the same as Top 25 Wealthy Individuals in SQL City (shown on page 5).

Alternative code syntax



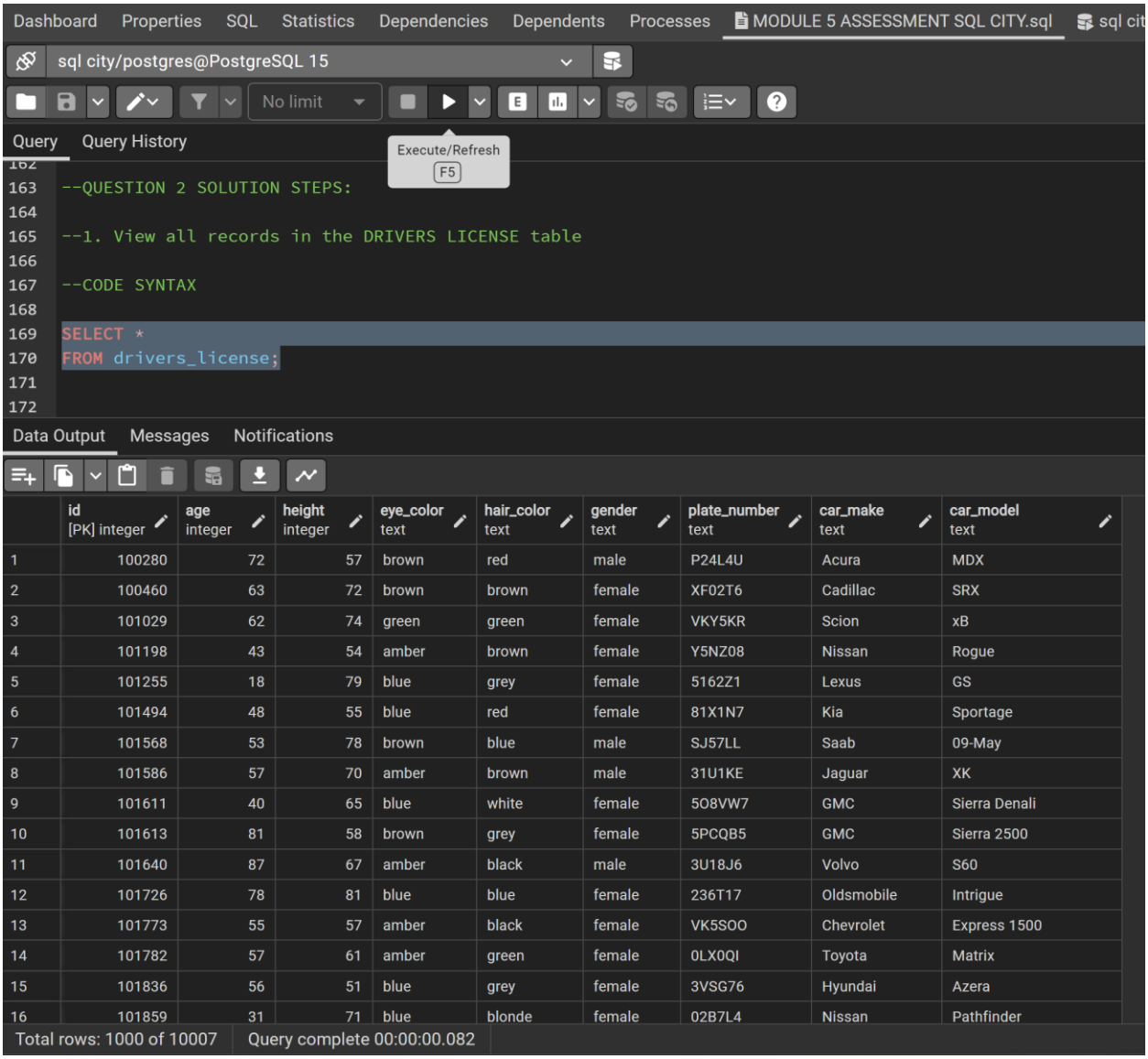
QUESTION 2:

To make the Festival a remarkable one, car manufacturers have reached out to the heads of the communities, they want a breakdown of the different vehicles and the number of persons in the community that drives such vehicle. You are to generate a summary table that reveals the different types of cars (car_make), and how many people (count) owned them in the community.

SOLUTION:

STEP 5: QUESTION 2 SOLUTION STEPS

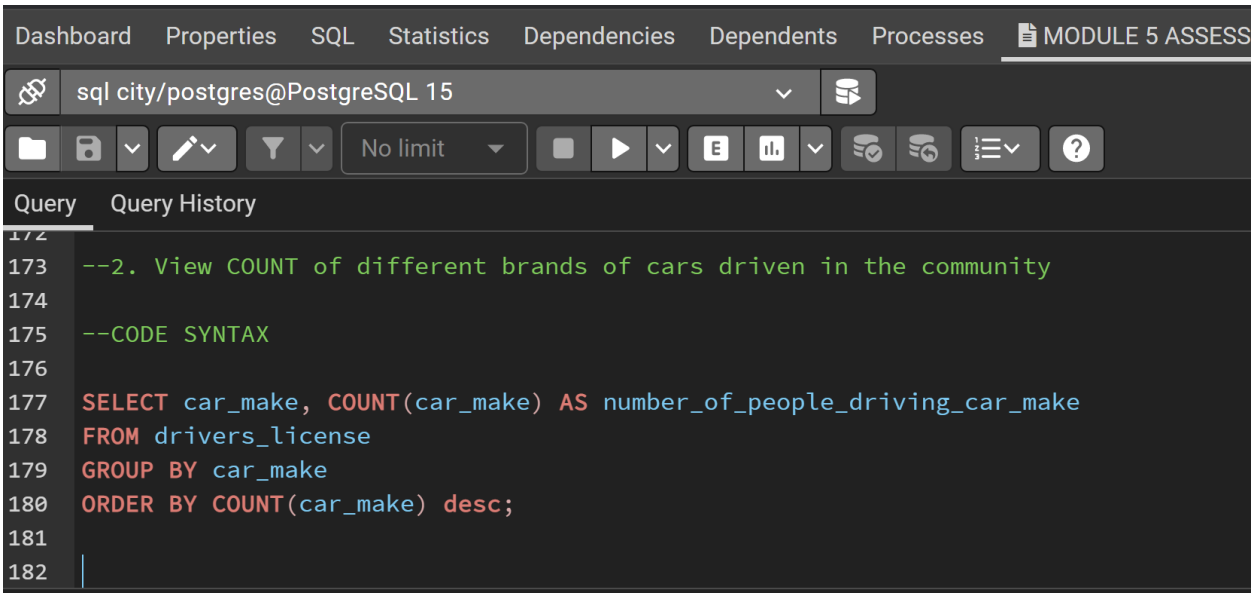
Step 5a: View all records in the "drivers_licence" Table



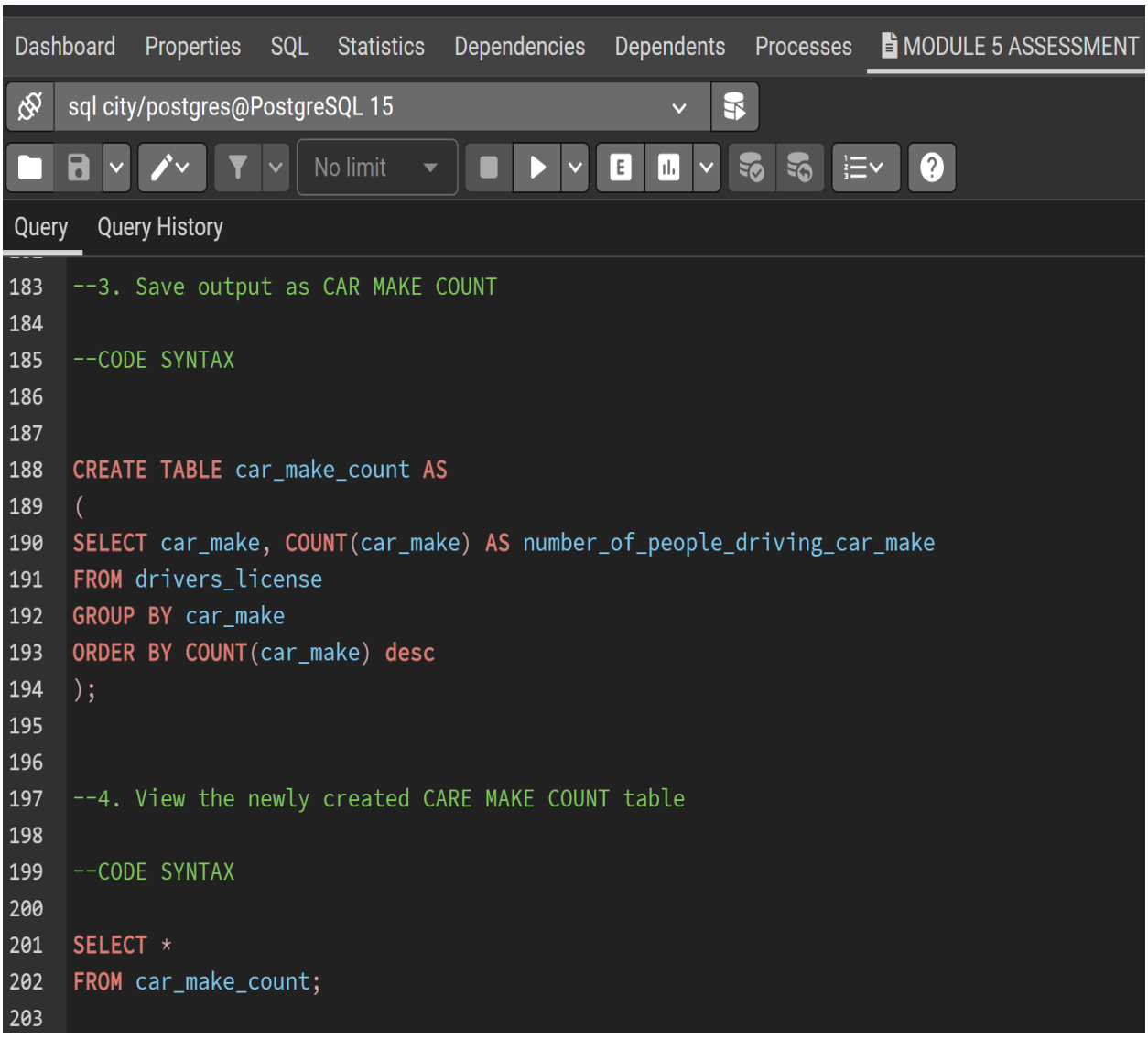
	id [PK] integer	age integer	height integer	eye_color text	hair_color text	gender text	plate_number text	car_make text	car_model text
1	100280	72	57	brown	red	male	P24L4U	Acura	MDX
2	100460	63	72	brown	brown	female	XF02T6	Cadillac	SRX
3	101029	62	74	green	green	female	VKY5KR	Scion	xB
4	101198	43	54	amber	brown	female	Y5NZ08	Nissan	Rogue
5	101255	18	79	blue	grey	female	5162Z1	Lexus	GS
6	101494	48	55	blue	red	female	81X1N7	Kia	Sportage
7	101568	53	78	brown	blue	male	SJ57LL	Saab	09-May
8	101586	57	70	amber	brown	male	31U1KE	Jaguar	XK
9	101611	40	65	blue	white	female	5O8VW7	GMC	Sierra Denali
10	101613	81	58	brown	grey	female	5PCQB5	GMC	Sierra 2500
11	101640	87	67	amber	black	male	3U18J6	Volvo	S60
12	101726	78	81	blue	blue	female	236T17	Oldsmobile	Intrigue
13	101773	55	57	amber	black	female	VK5SOO	Chevrolet	Express 1500
14	101782	57	61	amber	green	female	0LX0QI	Toyota	Matrix
15	101836	56	51	blue	grey	female	3VSG76	Hyundai	Azera
16	101859	31	71	blue	blonde	female	02B7L4	Nissan	Pathfinder

Total rows: 1000 of 10007 Query complete 00:00:00.082

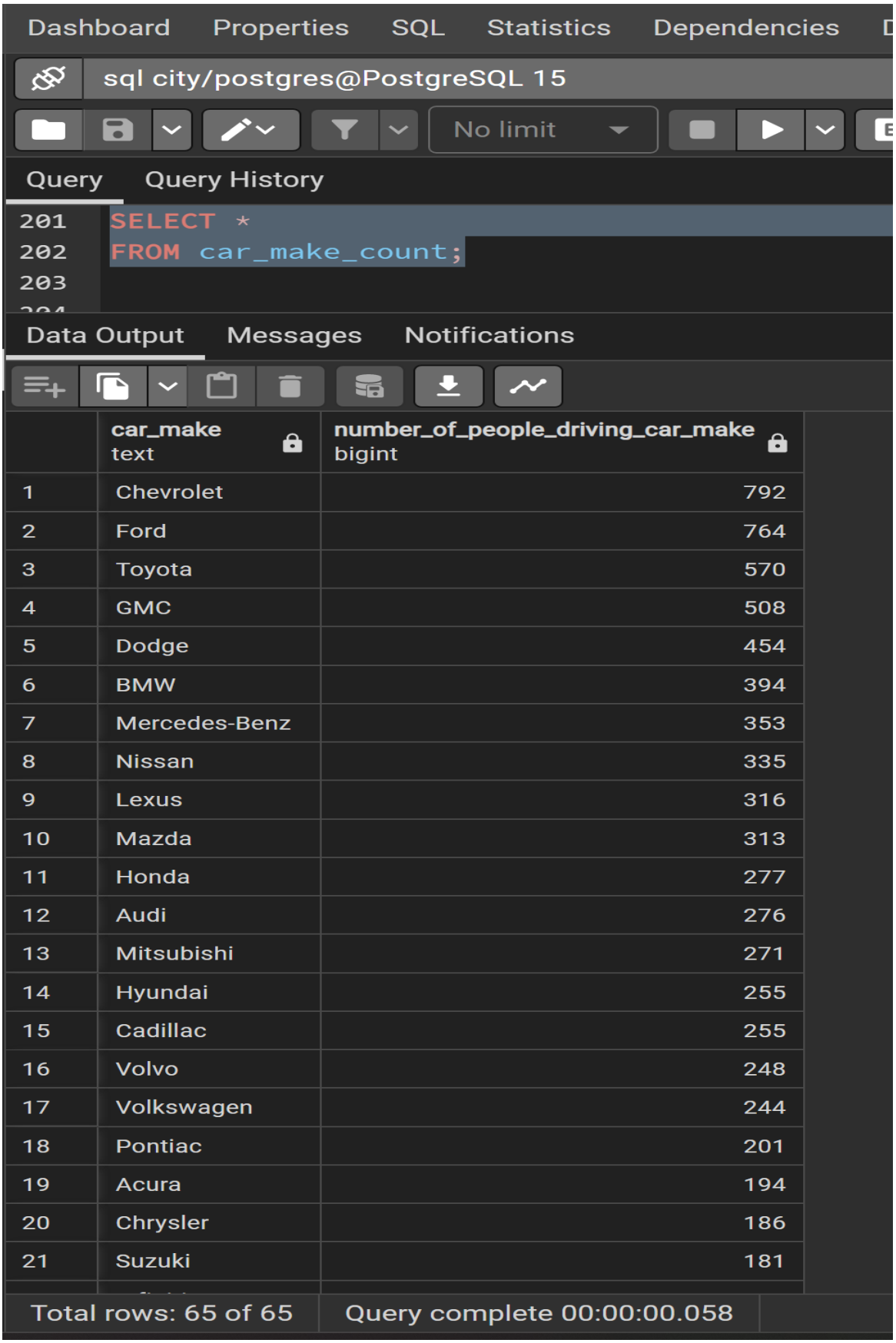
Step 5b: View COUNT of different brands of cars driven in SQL City community



Step 5c: Save output as "Care Make Count" & view new table



Car Make Count



	car_make text	number_of_people_driving_car_make bigint
1	Chevrolet	792
2	Ford	764
3	Toyota	570
4	GMC	508
5	Dodge	454
6	BMW	394
7	Mercedes-Benz	353
8	Nissan	335
9	Lexus	316
10	Mazda	313
11	Honda	277
12	Audi	276
13	Mitsubishi	271
14	Hyundai	255
15	Cadillac	255
16	Volvo	248
17	Volkswagen	244
18	Pontiac	201
19	Acura	194
20	Chrysler	186
21	Suzuki	181

QUESTION 3:

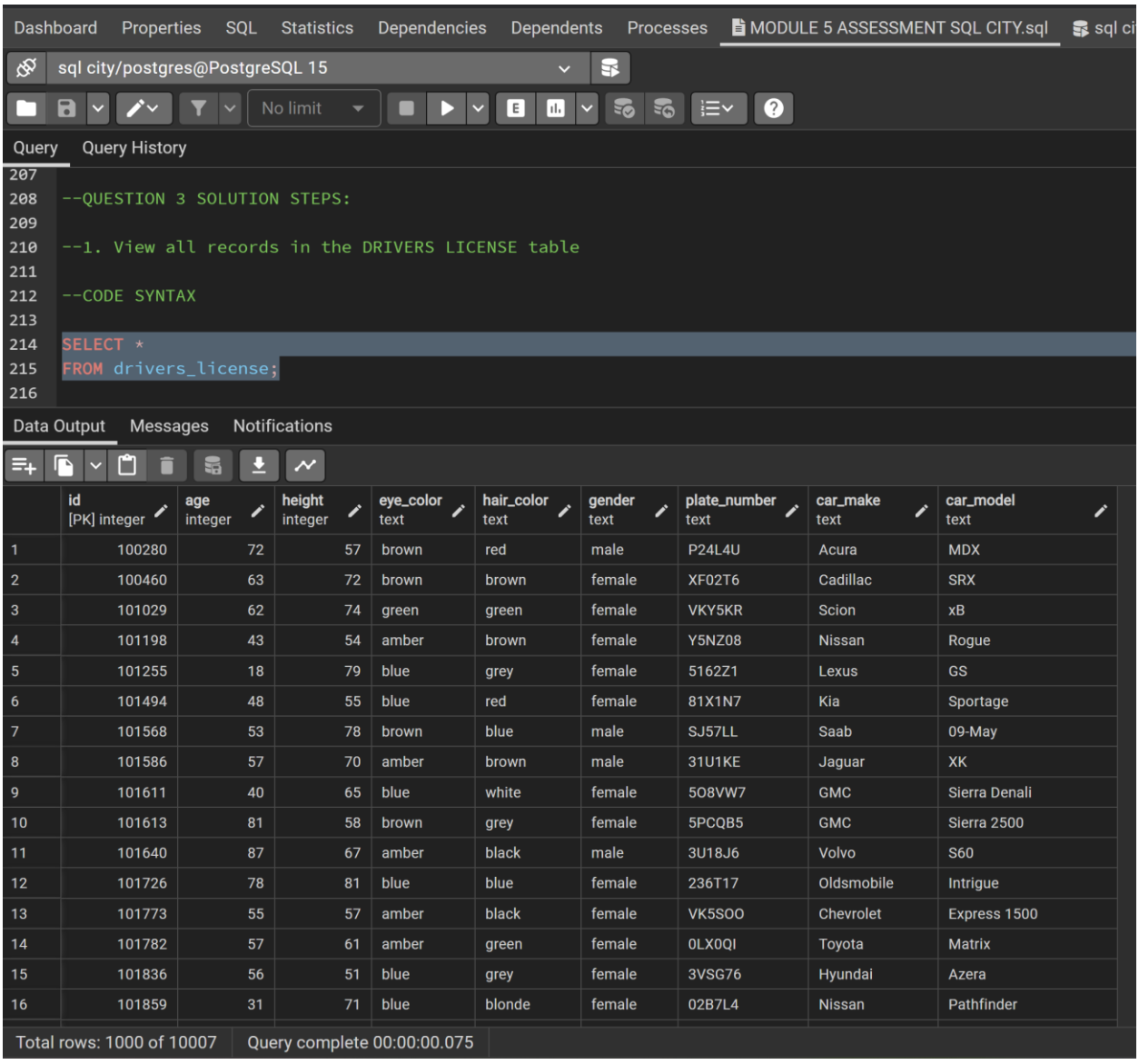
It’s good we get rid of the bad eggs in the city. Jeremy Bowers has confessed to who hired him:

i.) I was hired by a woman with a lot of money.
ii.) I don't know her name but I know she's around 5'5" (65") or 5'7" (67").
iii.) She has red hair and
iv.) She drives a Tesla Model S.
v.) I know that she attended the SQL Symphony Concert 3 times in December 2017.’

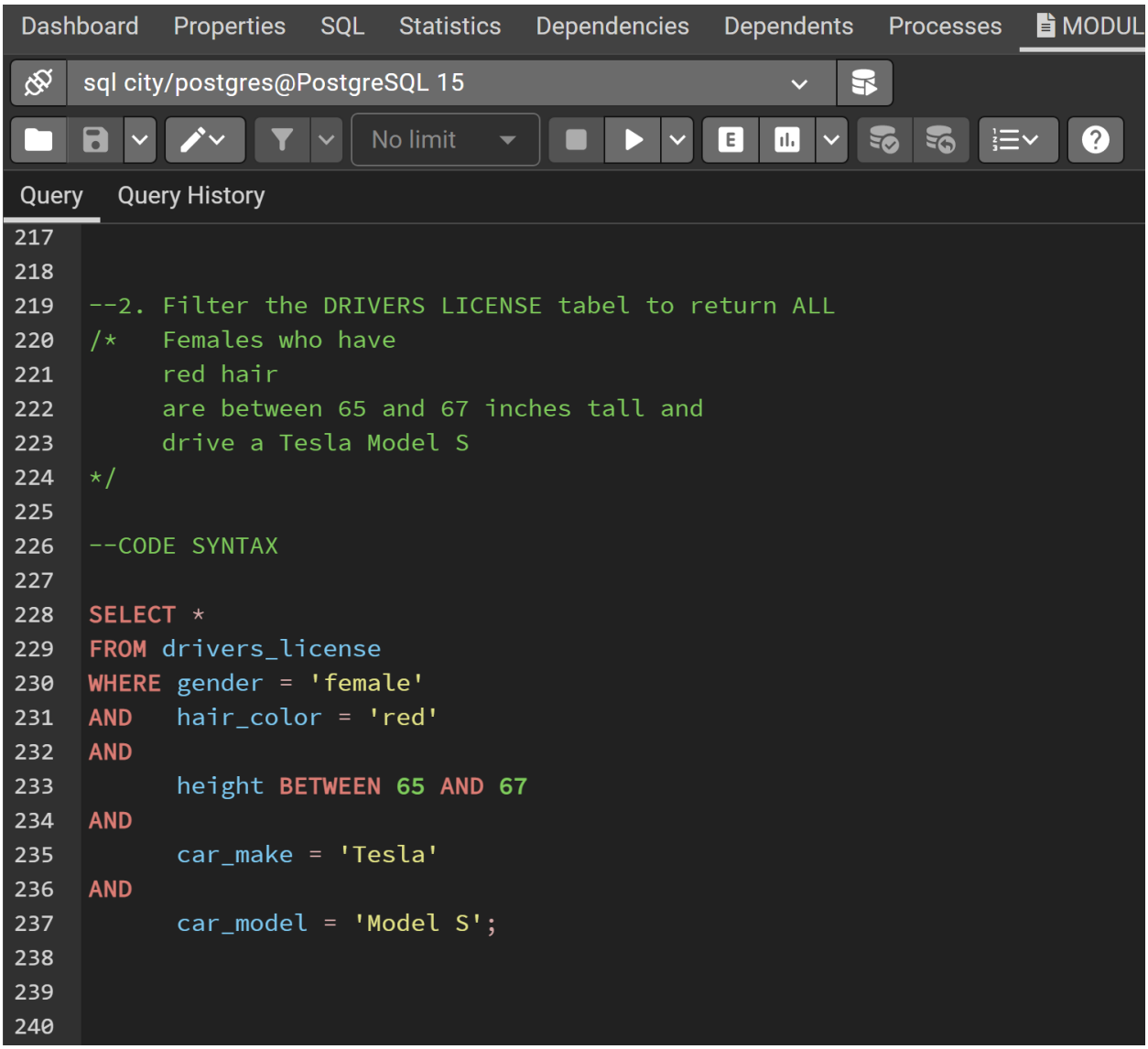
SOLUTION:

STEP 6: QUESTION 3 SOLUTION STEPS

Step 6a: View all records in the “drivers_licence” Table



Step 6b: Filter the "drivers_licence table to return the required conditions



Step 6c: Save the output as a new table called “FEMALE SUSPECTS” and view the new table

Dashboard Properties SQL Statistics Dependencies Dependents Processes MODULE 5 ASSESSMENT SQL CITY.sq

sql city/postgres@PostgreSQL 15

No limit

Query Query History

```
240
241 --3. Save the output as a new table called FEMALE SUSPECTS
242
243 --CODE SYNTAX
244
245 CREATE TABLE female_suspects AS (
246 SELECT *
247 FROM drivers_license
248 WHERE gender = 'female'
249 AND   hair_color = 'red'
250 AND
251       height BETWEEN 65 AND 67
252 AND
253       car_make = 'Tesla'
254 AND
255       car_model = 'Model S'
256 );
257
258
259 --4. View the new table FEMALE SUSPECTS
260
261 --CODE SYNTAX
262
263 SELECT *
264 FROM female_suspects;
265
```

Data Output Messages Notifications

	license_id integer	age integer	height integer	eye_color text	hair_color text	gender text	plate_number text	car_make text	car_model text
1	202298	68	66	green	red	female	500123	Tesla	Model S
2	291182	65	66	blue	red	female	08CM64	Tesla	Model S
3	918773	48	65	black	red	female	917UU3	Tesla	Model S

Total rows: 3 of 3 Query complete 00:00:00.070

Important Notes

Dashboard Properties SQL Statistics Dependencies Dependents Processes MODUL

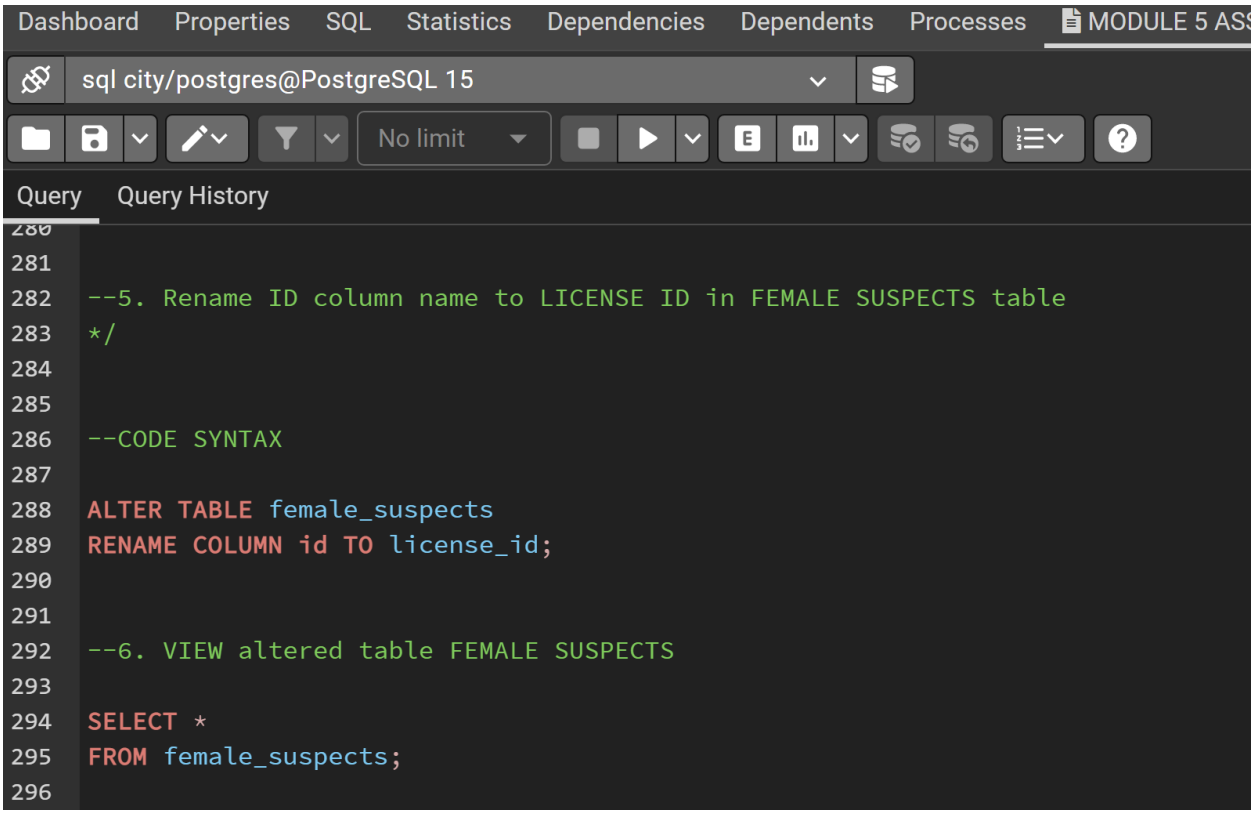
sql city/postgres@PostgreSQL 15

No limit

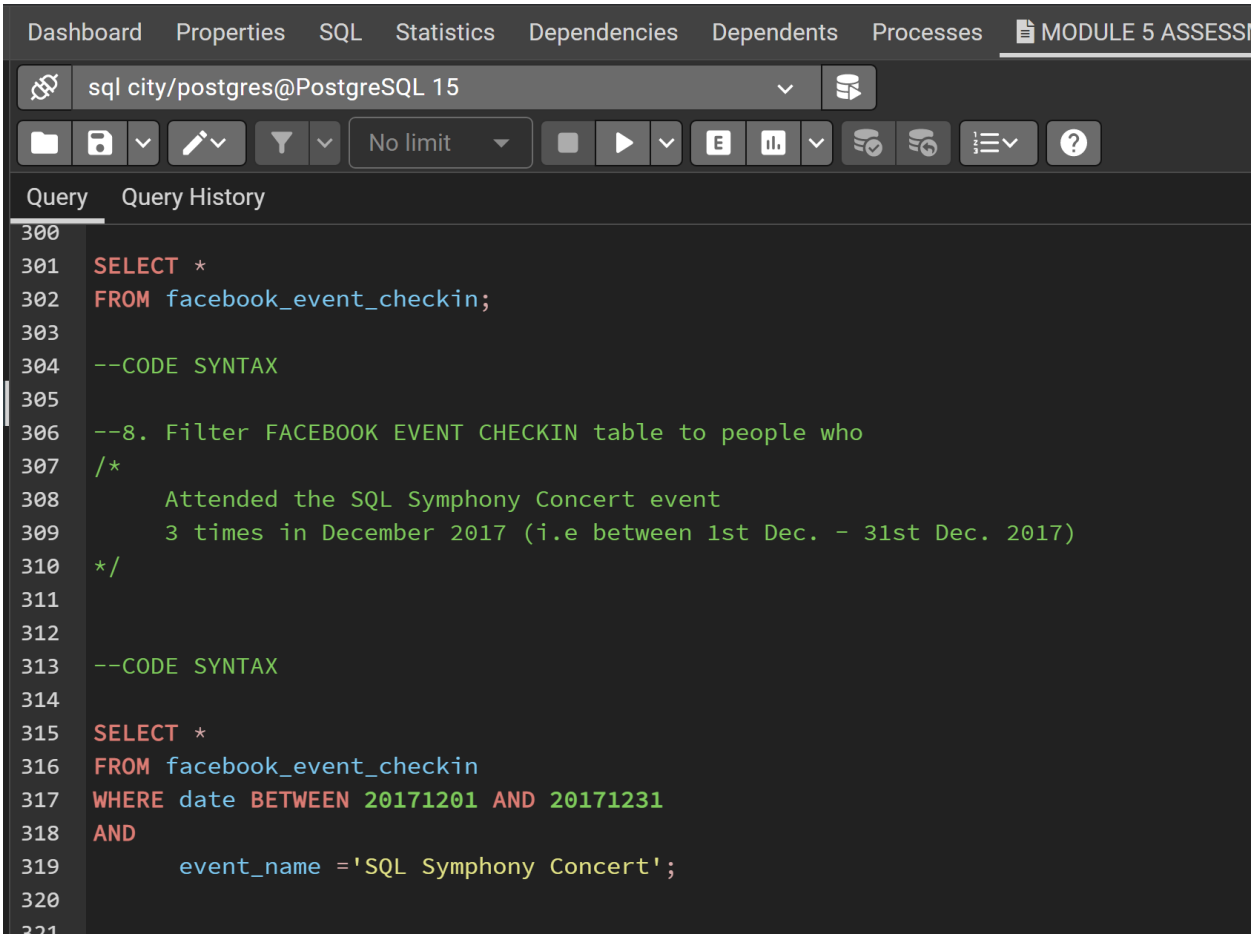
Query Query History

```
265
266 --IMPORTANT NOTES
267
268 /*
269 1. There are 3 female suspects who meet this criteria
270    i.e. have red hair
271         has a height between 65" and 67" and
272         drives a Tesla Model S
273
274 2. Their licence_id number are:
275         202298
276         291182
277         918773
278
279 However, this is named as simply 'id' in the table.
280
```

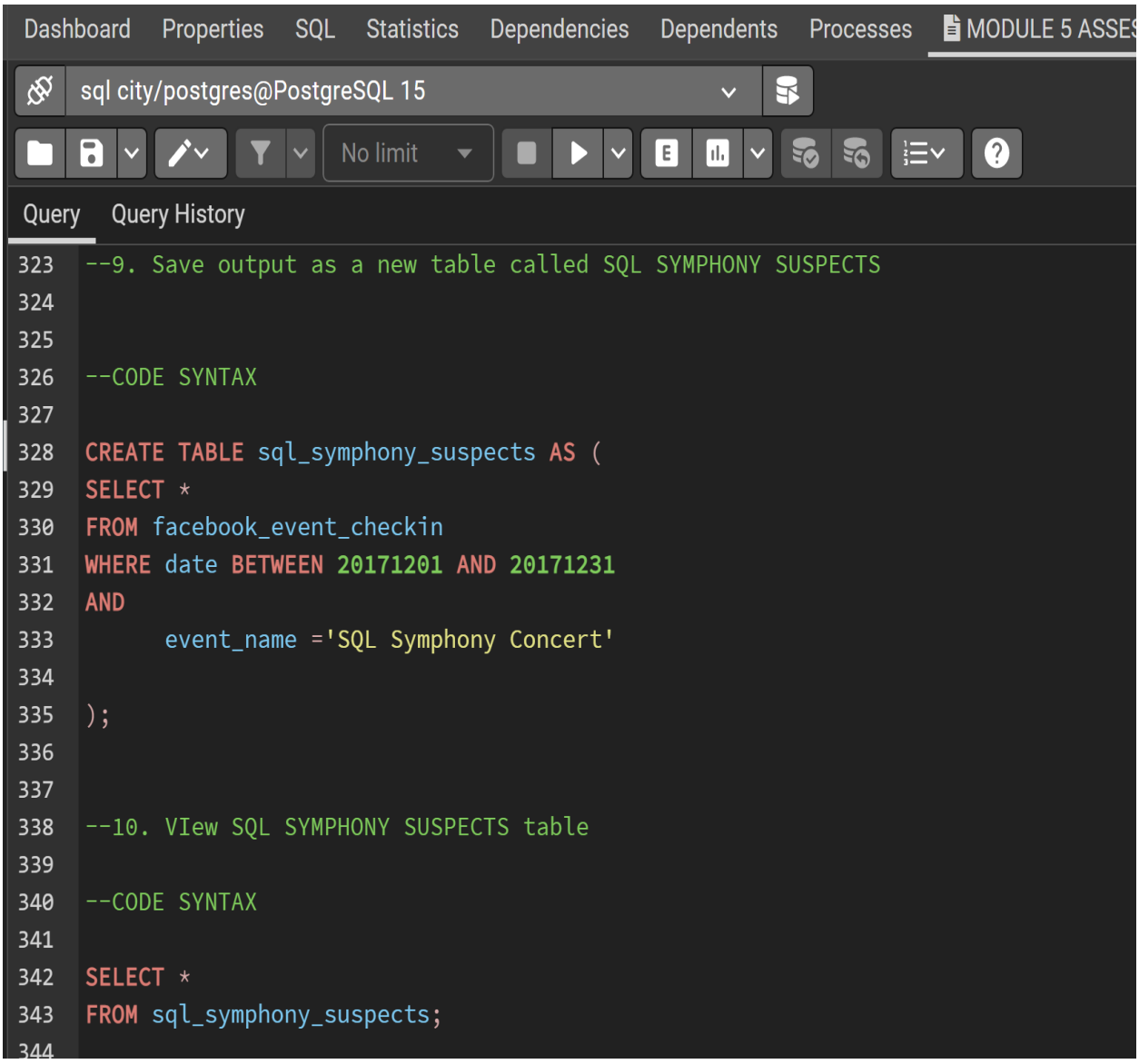
Step 6d: Rename “ID” column to “Licence ID”



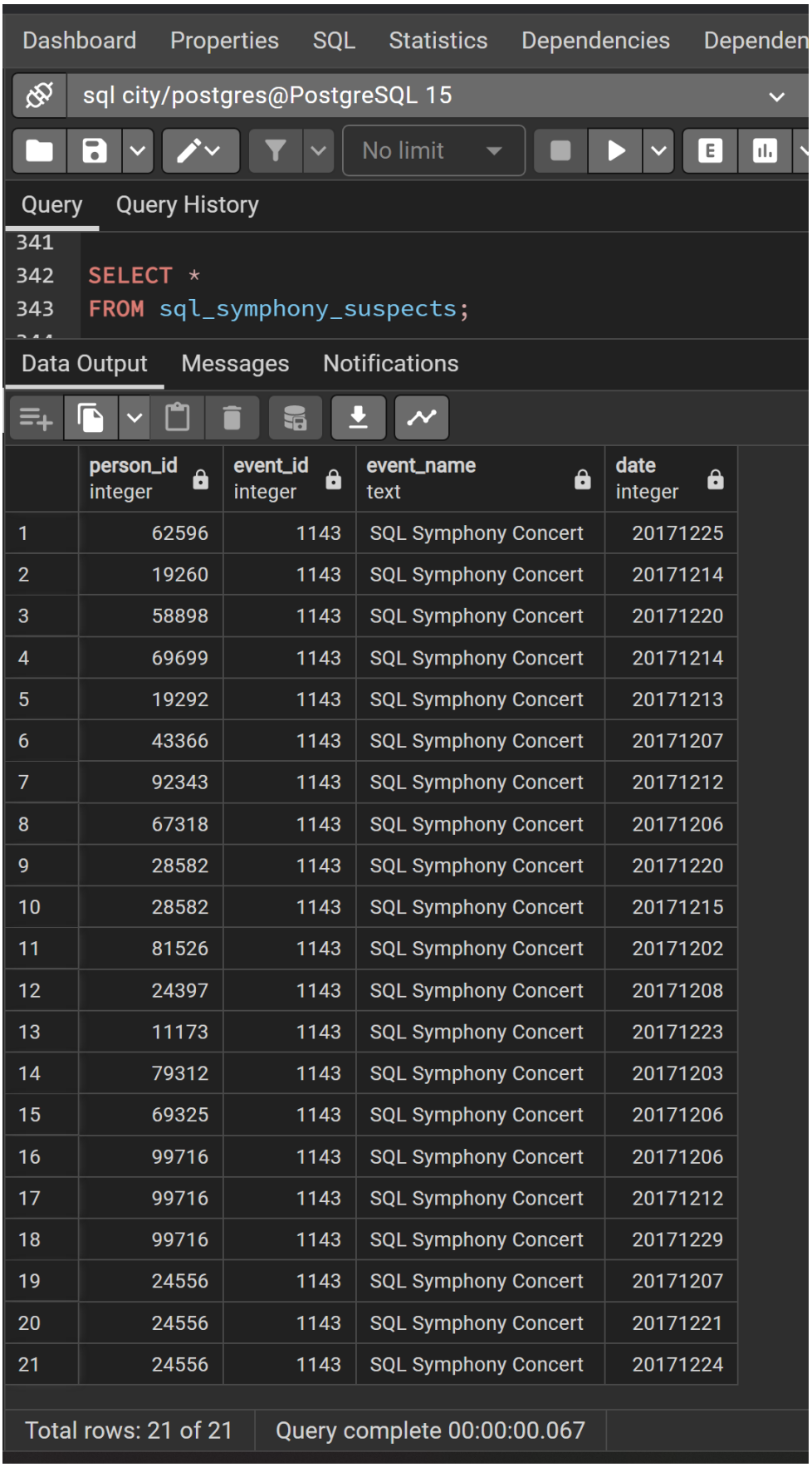
Step 6e: View and Filter “facebook_events_checkin” table using specified conditions



Step 6f: Save the output as a new table called SQL SYMPHONY SUSPECTS and view the new table



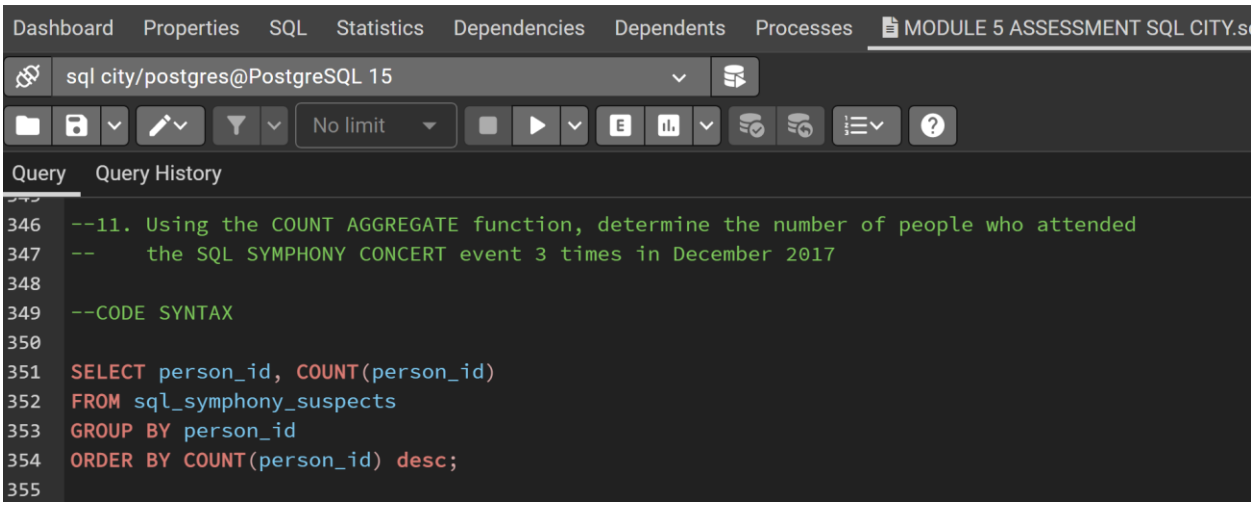
SQL SYMPHONY SUSPECTS



	person_id integer	event_id integer	event_name text	date integer
1	62596	1143	SQL Symphony Concert	20171225
2	19260	1143	SQL Symphony Concert	20171214
3	58898	1143	SQL Symphony Concert	20171220
4	69699	1143	SQL Symphony Concert	20171214
5	19292	1143	SQL Symphony Concert	20171213
6	43366	1143	SQL Symphony Concert	20171207
7	92343	1143	SQL Symphony Concert	20171212
8	67318	1143	SQL Symphony Concert	20171206
9	28582	1143	SQL Symphony Concert	20171220
10	28582	1143	SQL Symphony Concert	20171215
11	81526	1143	SQL Symphony Concert	20171202
12	24397	1143	SQL Symphony Concert	20171208
13	11173	1143	SQL Symphony Concert	20171223
14	79312	1143	SQL Symphony Concert	20171203
15	69325	1143	SQL Symphony Concert	20171206
16	99716	1143	SQL Symphony Concert	20171206
17	99716	1143	SQL Symphony Concert	20171212
18	99716	1143	SQL Symphony Concert	20171229
19	24556	1143	SQL Symphony Concert	20171207
20	24556	1143	SQL Symphony Concert	20171221
21	24556	1143	SQL Symphony Concert	20171224

Total rows: 21 of 21 | Query complete 00:00:00.067

Step 6g: COUNT the number of people who attended the SQL SYMPHONY CONCERT event 3 times in December 2017

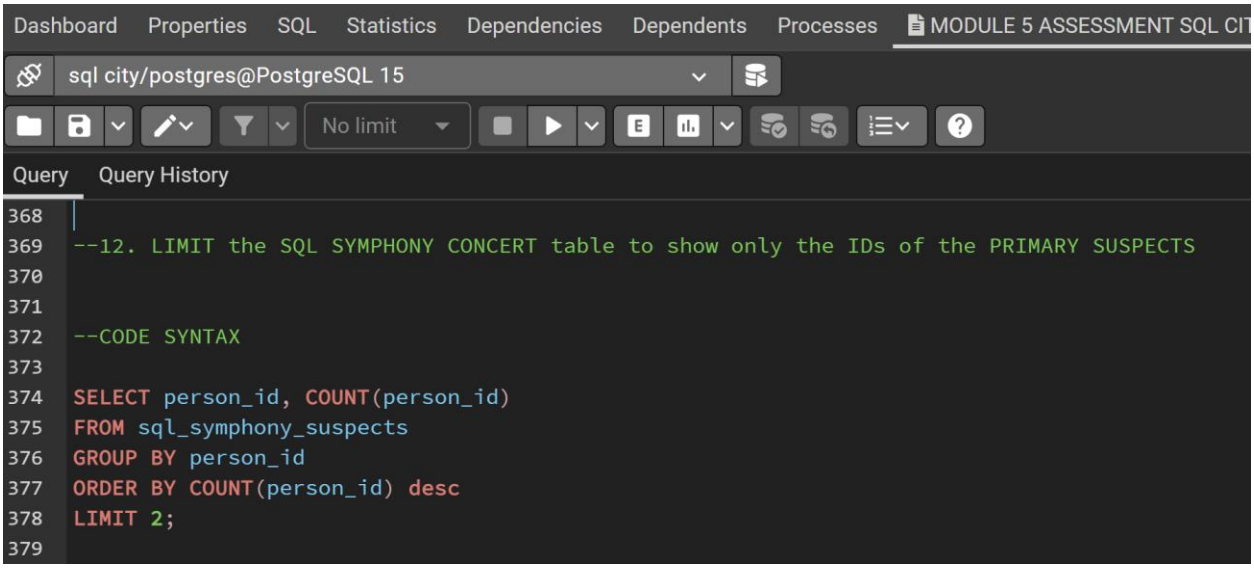



```
346 --11. Using the COUNT AGGREGATE function, determine the number of people who attended
347 --    the SQL SYMPHONY CONCERT event 3 times in December 2017
348
349 --CODE SYNTAX
350
351 SELECT person_id, COUNT(person_id)
352 FROM sql_symphony_suspects
353 GROUP BY person_id
354 ORDER BY COUNT(person_id) desc;
355
```

Important Notes:

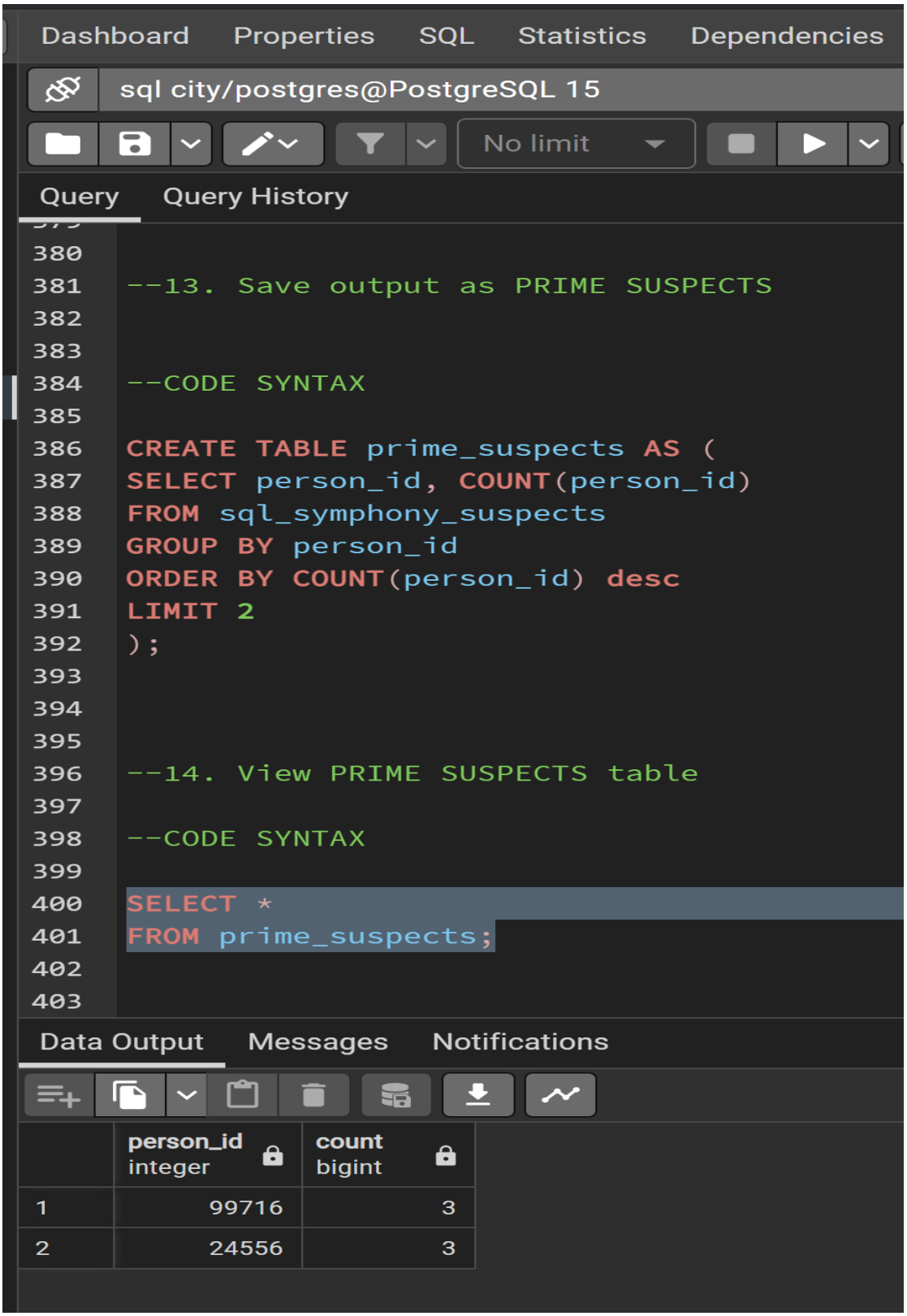
```
355
356 --IMPORTANT NOTES
357
358 /* 2 people attended the SQL SYMPHONY CONCERT event 3 times each in December 2017
359
360    Their person_id are
361    99716
362    24556
363
364    These are the PRIME SUSPECTS!
365 */
366
```

Step 6h: LIMIT the table to show only PRIME SUSPECTS

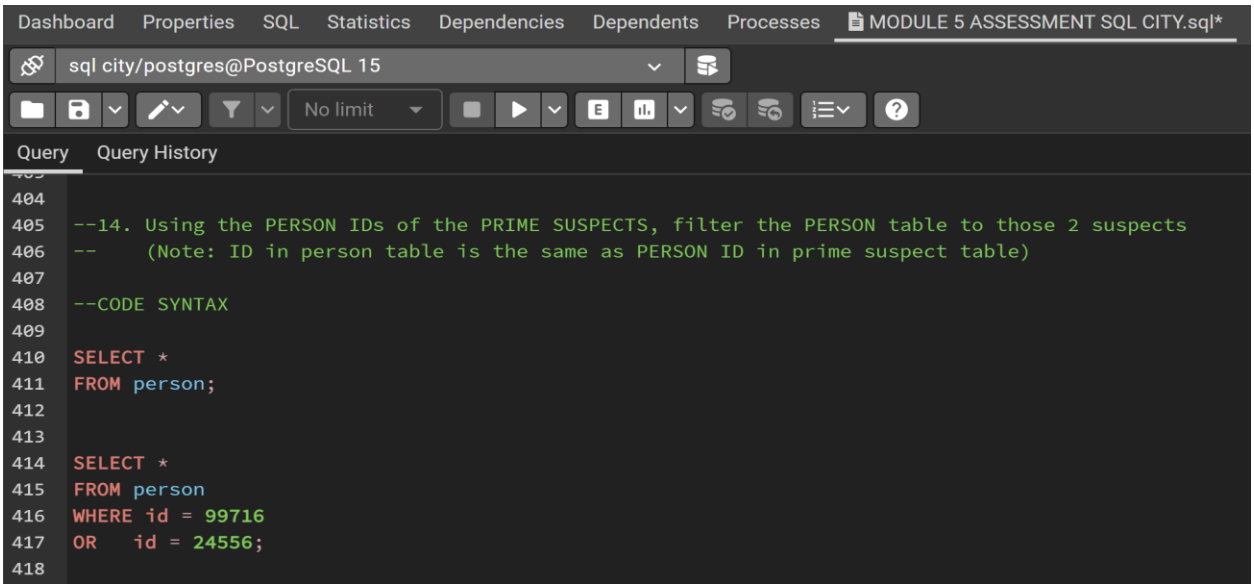



```
368
369 --12. LIMIT the SQL SYMPHONY CONCERT table to show only the IDs of the PRIMARY SUSPECTS
370
371
372 --CODE SYNTAX
373
374 SELECT person_id, COUNT(person_id)
375 FROM sql_symphony_suspects
376 GROUP BY person_id
377 ORDER BY COUNT(person_id) desc
378 LIMIT 2;
379
```

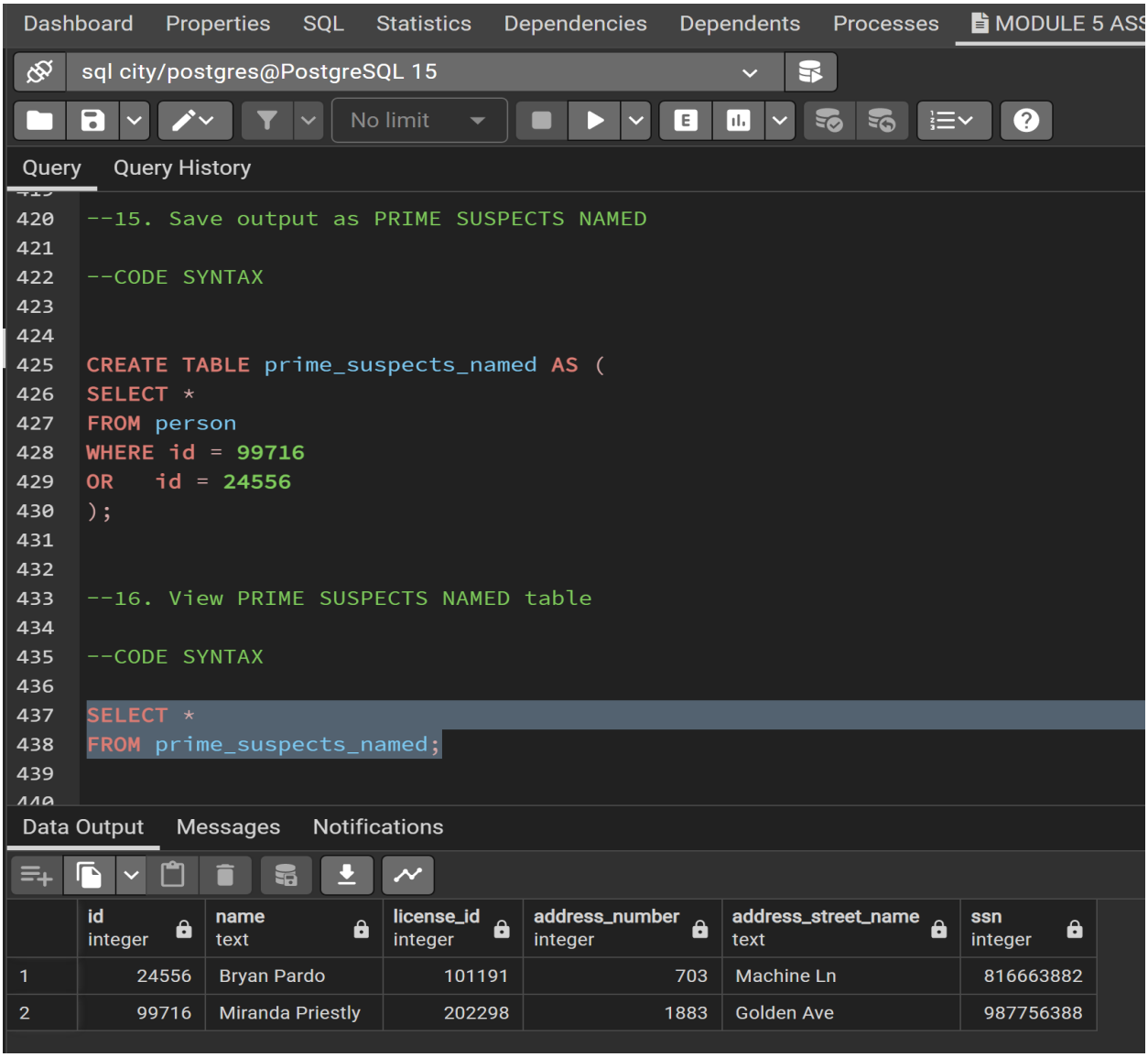
Step 6i: Save the output as a new table called PRIME SUSPECTS and view the table



Step 6j: Filter the "PERSON" table to identify the PRIME SUSPECTS using their person_id

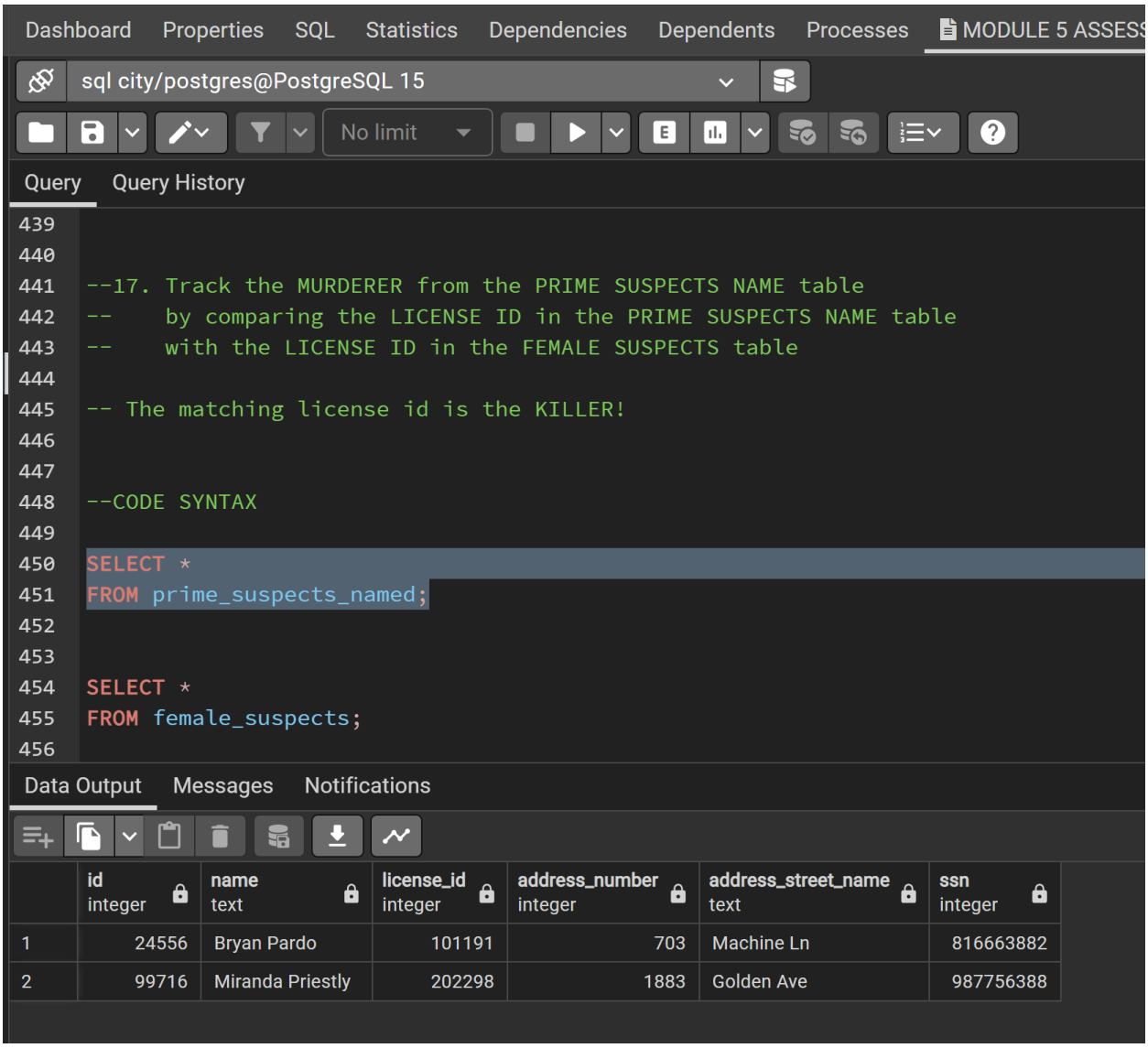


Step 6k: Save the output as PRIME SUSPECTS NAMED and view the new table

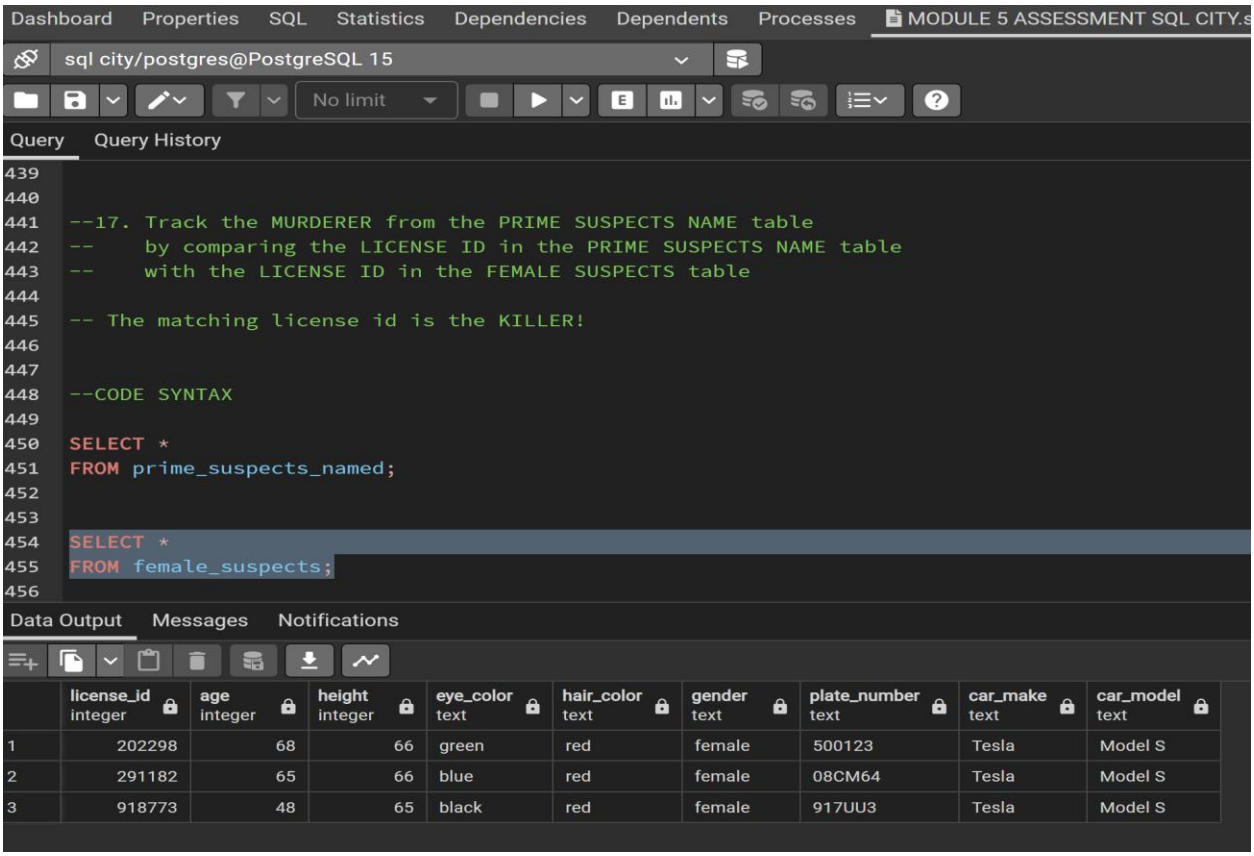


	id integer	name text	license_id integer	address_number integer	address_street_name text	ssn integer
1	24556	Bryan Pardo	101191	703	Machine Ln	816663882
2	99716	Miranda Priestly	202298	1883	Golden Ave	987756388

Step 6l: Closing in on the Killer (prime_suspects_named table output)!

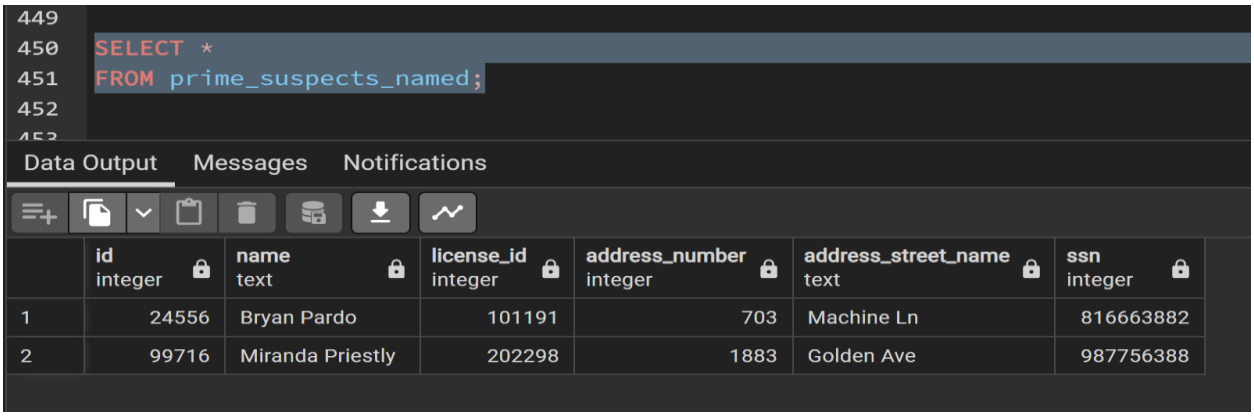


Step 6m: Closing in on the Killer (female_suspects table output)

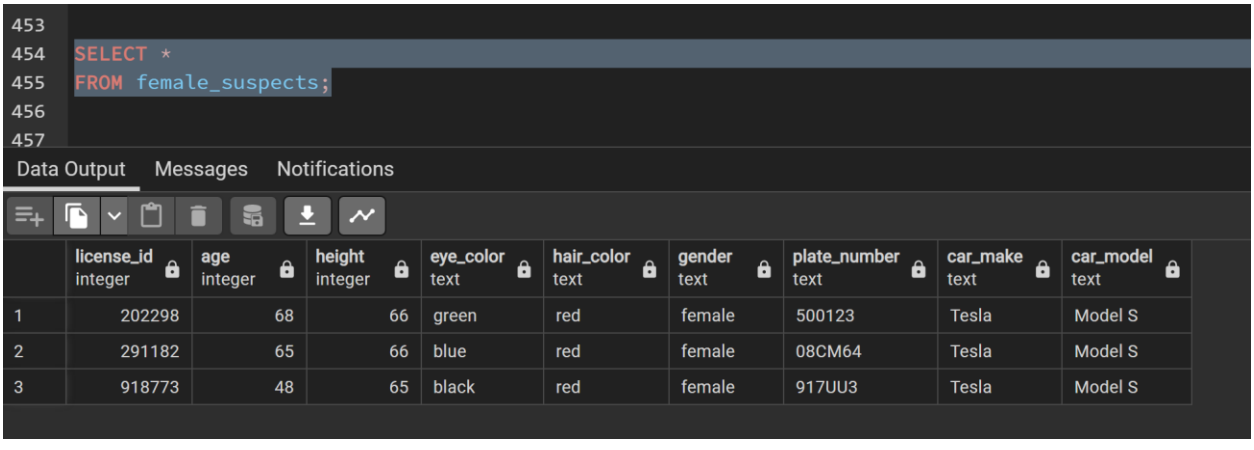


	license_id integer	age integer	height integer	eye_color text	hair_color text	gender text	plate_number text	car_make text	car_model text
1	202298	68	66	green	red	female	500123	Tesla	Model S
2	291182	65	66	blue	red	female	08CM64	Tesla	Model S
3	918773	48	65	black	red	female	917UU3	Tesla	Model S

Step 6n: Closing in on the Killer (Comparing the 2 tables)



	id integer	name text	license_id integer	address_number integer	address_street_name text	ssn integer
1	24556	Bryan Pardo	101191	703	Machine Ln	816663882
2	99716	Miranda Priestly	202298	1883	Golden Ave	987756388



	license_id integer	age integer	height integer	eye_color text	hair_color text	gender text	plate_number text	car_make text	car_model text
1	202298	68	66	green	red	female	500123	Tesla	Model S
2	291182	65	66	blue	red	female	08CM64	Tesla	Model S
3	918773	48	65	black	red	female	917UU3	Tesla	Model S

Important Note:

```
457
458  --18. Matching LICENSE ID
459
460  /*
461        The matching license ID is 202298 and this belongs to MIRANDA PRIESTLY
462        Therefore,
463        MIRANDA PRIESTLY is the KILLER!
464
465  */
466
```

Step 6o: Identify the Killer (using both person_id and license_id)!

```
471
472  SELECT *
473  FROM drivers_license
474  WHERE id = 202298;
475
```

Data Output | Messages | Notifications

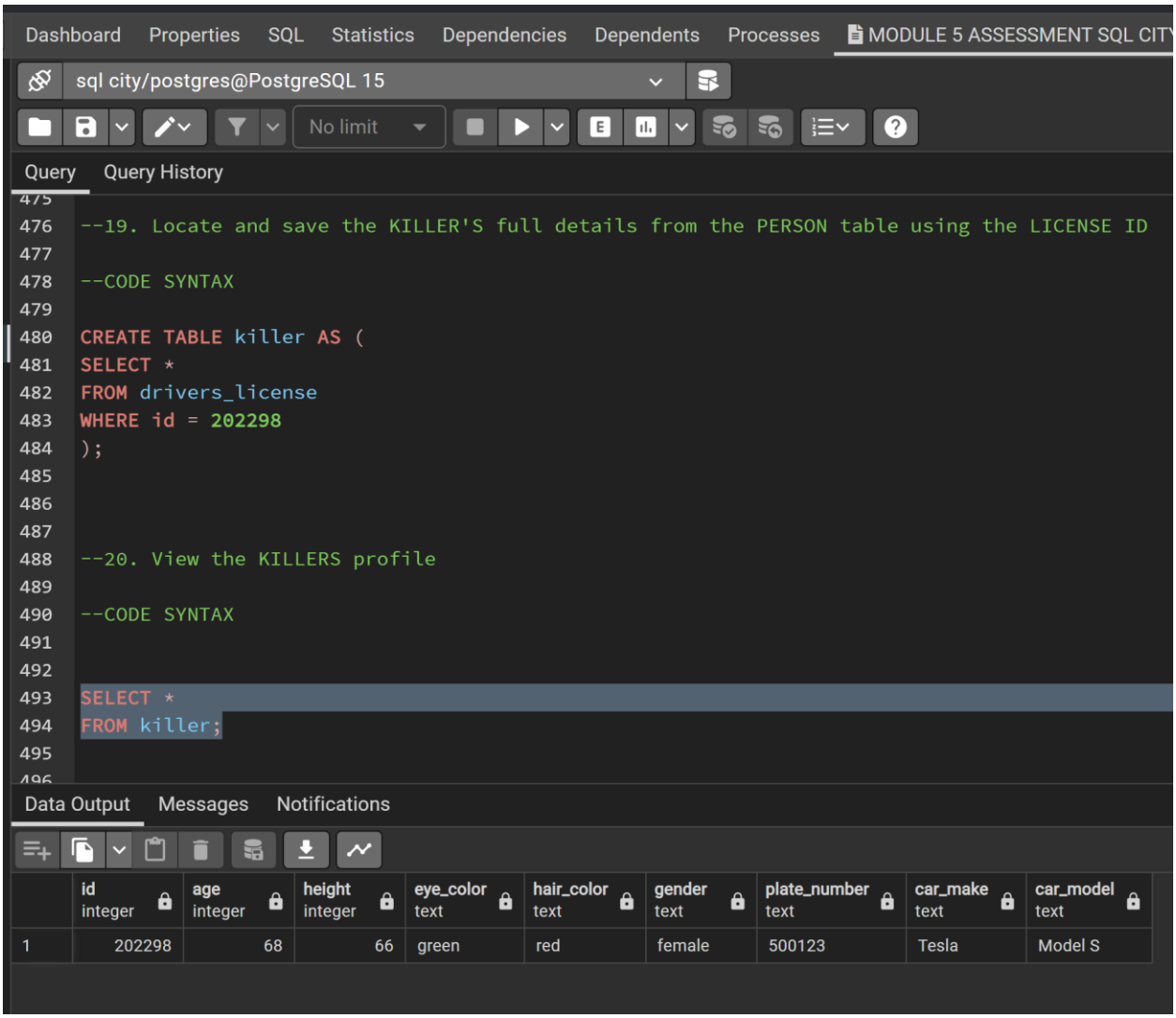
	id [PK] integer	age integer	height integer	eye_color text	hair_color text	gender text	plate_number text	car_make text	car_model text
1	202298	68	66	green	red	female	500123	Tesla	Model S

```
467
468  SELECT *
469  FROM person
470  WHERE id = 99716
471
```

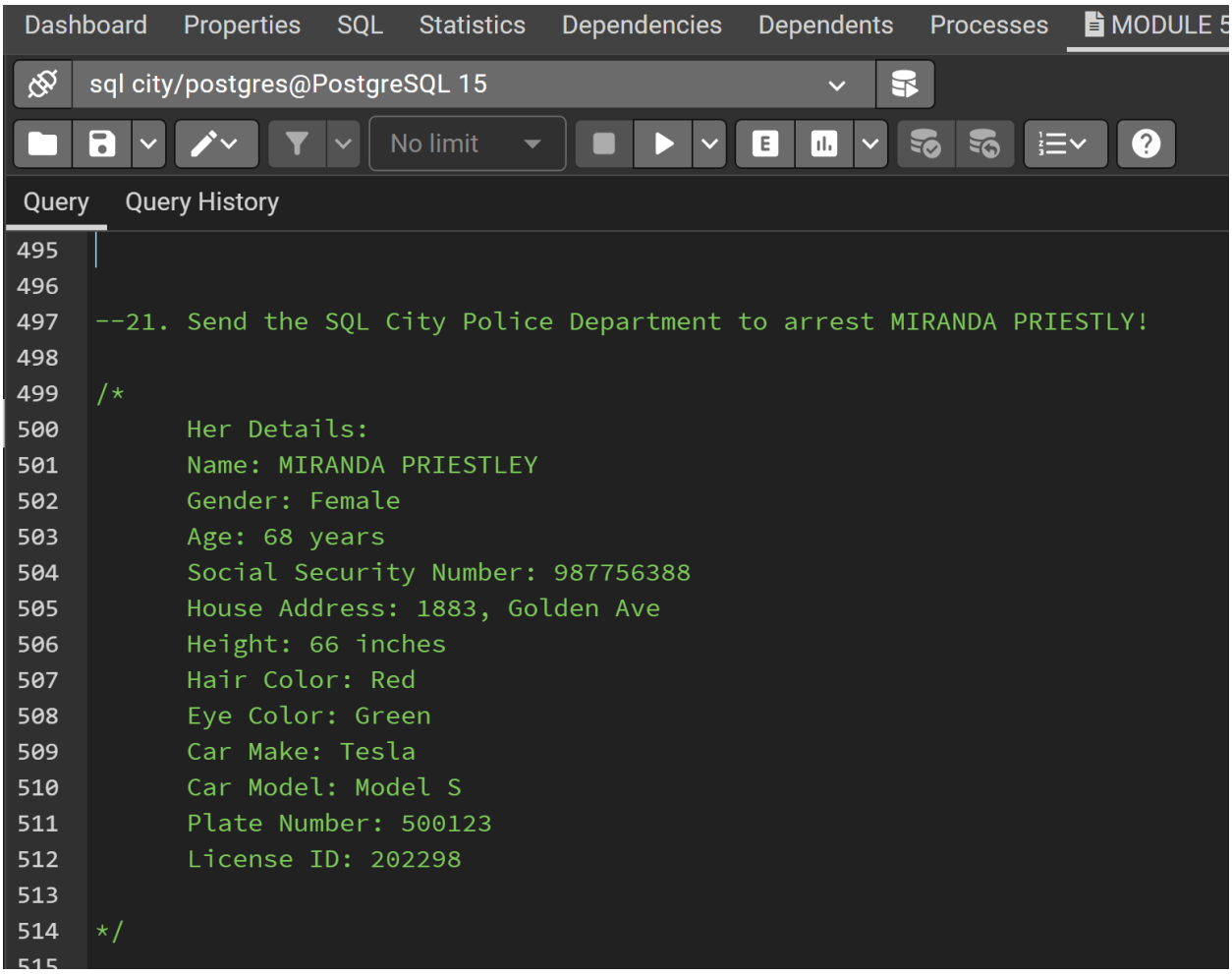
Data Output | Messages | Notifications

	id [PK] integer	name text	license_id integer	address_number integer	address_street_name text	ssn integer
1	99716	Miranda Priestly	202298	1883	Golden Ave	987756388

Step 6p: Retrieve the Killer's details



Step 6q: Profile the Killer and send details to the police!



WHODUNNIT? Miranda Priestly

SQL CITY'S MURDER MYSTERY SOLVED!