Queries (RTO database)

1. Find the average test score for each invigilator, only considering invigilators who have conducted more than one test

Query | Query History

```
SELECT invigilator, AVG(test_score) AS avg_score
FROM (
    SELECT t.*,COUNT(*) OVER(PARTITION BY invigilator) AS test_count
    FROM "Test" t
) AS subquery
WHERE test_count > 1
GROUP BY invigilator;
```

Data Output | Messages | Notifications

	invigilator character varying (15)	avg_score numeric
1	Dr. Kumar	41.5000000000000000
2	Mr. Sharma	61.1666666666666667
3	Mrs. Patel	56.0000000000000000
4	Ms. Joshi	37.2000000000000000
5	Prof. Singh	42.8000000000000000

2. List the number of appointments per branch along with the total number of appointments for each applying category

```
SELECT a.branch, a.applying_for, COUNT(*) AS num_appointments
FROM "Appointment" a
INNER JOIN "Applied_To" at ON a.appointment_no = at.appointment_no
GROUP BY a.branch, a.applying_for
ORDER BY a.branch, a.applying_for;
```

Data Output | Messages | Notifications

	branch character varying (30)	applying_for character varying (20)	num_appointments bigint
1	Bangalore Koramangala	Driving License	2
2	Bangalore Koramangala	Learning License	10
3	Chennai T-Nagar	Driving License	2
4	Delhi Janakpuri	Driving License	1
5	Mumbai Andheri	Learning License	5

3. Find the average test score for each applying category for appointments that are scheduled on weekdays (Monday to Friday)

Query Query History

```
1  SELECT a.applying_for, AVG(t.test_score) AS avg_score
2  FROM "Appointment" a
3  INNER JOIN "Applied_To" at ON a.appointment_no = at.appointment_no
4  INNER JOIN "Test" t ON at.roll_no = t.roll_no
5  WHERE EXTRACT(ISODOW FROM a.date) BETWEEN 1 AND 5
6  GROUP BY a.applying_for;
```

Data Output Messages Notifications

	applying_for character varying (20)	avg_score numeric
1	Driving License	68.5000000000000000
2	Learning License	45.3000000000000000

4. To find the count of appointments for each payment mode, ordered by the appointment count in descending order.

Query Query History

```
1  SELECT p."payment_mode", COUNT(DISTINCT a."appointment_no") AS "appointment_count"
2  FROM "Payment" p
3  JOIN "Appointment" a ON p."transaction_id" = a."app_txn_id"
4  GROUP BY p."payment_mode"
5  ORDER BY "appointment_count" DESC;
```

Data Output Messages Notifications

	payment_mode character varying (10)	appointment_count bigint
1	cash	42
2	card	18
3	online	16

5. To find the appointment numbers, dates, and statuses for appointments scheduled at counters where the employee names start with 'A'

Query Query History

```
SELECT a.appointment_no, a.date, a.status
FROM "Appointment" a
INNER JOIN "Counter" c ON a.counter_no = c.counter_no
WHERE c.employee_name LIKE 'A%';
```

Data Output Messages Notifications

	appointment_no [PK] character varying (20)	date date	status character varying (15)
1	APP00033	2024-08-02	Confirmed
2	APP00037	2024-08-02	Confirmed
3	APP00041	2024-08-02	Confirmed
4	APP00045	2024-08-02	Confirmed
5	APP00049	2025-02-16	Pending
6	APP00053	2025-02-16	Pending
7	APP00057	2025-02-16	Pending
8	APP00061	2025-02-16	Pending
9	APP00062	2025-02-01	Confirmed
10	APP00064	2024-12-11	Pending
11	APP00065	2025-02-01	Confirmed
12	APP00067	2024-12-11	Pending
13	APP00068	2025-02-01	Confirmed
14	APP00070	2024-12-11	Pending
15	APP00071	2025-02-01	Confirmed
16	APP00073	2024-12-11	Pending
17	APP00074	2025-02-01	Confirmed
18	APP00076	2024-12-11	Pending

6. List the vehicle registration details for vehicles registered by users whose first names contain the letter 'A' and are in cities with a pincode ending in '1'

Query Query History

```
SELECT vr.*
FROM "Vehicle_Registration" vr
INNER JOIN "User" u ON vr.user_id = u.user_id
INNER JOIN "Pincode" p ON vr.pincode = p.pincode
WHERE u.first_name LIKE '%A%' AND p.pincode % 10 = 1;
```

Data Output Messages Notifications

	registration_no [PK] character varying (20)	registration_date date	vehicle_type character varying (50)	model character varying (100)	year_of_manufacture integer	engine_no character varying (50)	chassis_no character varying (50)	color character varying (50)	temp_reg_no character varying (20)	user_id character
1	DL48TC8505	2021-05-10	Four Wheeler	Toyota Innova	2011	hlv0682lMM497	FjP2202iNd465	lime	TEMP46378	ID00021
2	KA92TC4962	2020-09-24	Four Wheeler	Hyundai i10	2020	kZF2282jNx151	dgJ8002XBC956	white	TEMP68076	ID00024
3	MH48TC2654	2021-07-26	Two Wheeler	Royal Enfield Classic	2020	fEg1239wCA905	uoP7428cWC045	silver	TEMP63883	ID00031
4	DL96TC8222	2020-08-16	Four Wheeler	Tata Nexon	2018	PZW3135uMD288	GWV1240yMz601	white	TEMP17104	ID00025
5	BR45TC8777	2020-12-20	Four Wheeler	Toyota Innova	2021	xPK9903ECX330	RHg5468cBo476	teal	TEMP79322	ID00027
6	UP10TC4020	2023-08-29	Two Wheeler	TVS Apache	2023	qqx2079Xtx081	XQK3007pzG759	black	TEMP34180	ID00038
7	PB93TC9731	2021-12-30	Two Wheeler	Honda Activa	2021	Moh3006xmn158	jcD7245Vjl106	fuchsia	TEMP37833	ID00029
8	DL33TC4002	2022-11-02	Two Wheeler	Royal Enfield Classic	2019	duS5997acc020	rNp1401mXo044	gray	TEMP62519	ID00030
9	TN28TC3592	2022-02-16	Four Wheeler	Mahindra Thar	2012	eRF7299EiG029	ZZE0802xfq843	navy	TEMP86888	ID00026

7. Find the average test score for appointments applied to by users who have applied to less than two appointments:

Query Query History

```
SELECT AVG(t.test_score) AS avg_score
FROM "Applied_To" at
INNER JOIN "Test" t ON at.roll_no = t.roll_no
WHERE at.roll_no IN (
    SELECT roll_no
    FROM "Applied_To"
    GROUP BY roll_no
    HAVING COUNT(appointment_no) < 2
);
```

Data Output Messages Notifications

	avg_score numeric
1	52.0000000000000000

8. Show all permits due to expire within the next 3 months, sorted by expiration date:

Query Query History

```
SELECT permit_no, purpose, validity
FROM "Permit"
WHERE validity BETWEEN CURRENT_DATE AND (CURRENT_DATE + INTERVAL '3 months')
ORDER BY validity;
```

Data Output Messages Notifications

	permit_no [PK] character varying (20)	purpose character varying (20)	validity date
1	PERMIT2	Tourism	2024-05-10
2	PERMIT12	Business	2024-06-01
3	PERMIT4	Medical	2024-07-15

9. Show users who applied for learning license and have paid for an appointment but have not taken the test

```
Query  Query History
1  SELECT U.first_name, U.last_name
2  FROM "User" U
3  JOIN "Learning_License" ll ON U.user_id = ll.user_id
4  JOIN "Payment" P ON ll.app_txn_id = P.transaction_id
5  LEFT JOIN "Test" T ON U.user_id = T.roll_no
6  WHERE T.date IS NULL;
```

Data Output Messages Notifications

	first_name character varying (20)	last_name character varying (20)
1	Arjun	Jain
2	Vivaan	Mehra
3	Siddharth	Kumar
4	Ansh	Jain
5	Ansh	Kapoor
6	Aryan	Jain
7	Aarav	Gupta
8	Dhruv	Patel
9	Aditya	Singh
10	Arjun	Malhotra
11	Arjun	Gupta
12	Vivaan	Mehra
13	Ishaan	Kapoor
14	Ishaan	Sharma
15	Dhruv	Jain

10. Display the next 10 appointments starting from an offset of 5

```
Query  Query History
1  SELECT *
2  FROM "Appointment"
3  ORDER BY date, timeslot
4  LIMIT 10 OFFSET 5;
```

Data Output Messages Notifications

	appointment_no [PK] character varying (20)	date date	status character varying (15)	applying_for character varying (20)	branch character varying (30)	timeslot timestamp without time zone	counter_no character varying (5)	app_txn_id character varying (20)
1	APP00042	2024-06-09	Pending	Driving License	Chennai T-Nagar	2025-02-24 15:19:00	C9	TX00042
2	APP00038	2024-06-09	Pending	Driving License	Chennai T-Nagar	2025-02-24 15:19:00	C9	TX00038
3	APP00020	2024-06-10	Pending	Learning License	Bangalore Koramangala	2024-11-27 19:55:00	C10	TX00020
4	APP00026	2024-06-10	Pending	Learning License	Bangalore Koramangala	2024-11-27 19:55:00	C10	TX00026
5	APP00029	2024-06-10	Pending	Learning License	Bangalore Koramangala	2024-11-27 19:55:00	C10	TX00029
6	APP00017	2024-06-10	Pending	Learning License	Bangalore Koramangala	2024-11-27 19:55:00	C10	TX00017
7	APP00023	2024-06-10	Pending	Learning License	Bangalore Koramangala	2024-11-27 19:55:00	C10	TX00023
8	APP00032	2024-07-10	Pending	Driving License	Delhi Janakpuri	2024-05-24 16:49:00	C2	TX00032
9	APP00040	2024-07-10	Pending	Driving License	Delhi Janakpuri	2024-05-24 16:49:00	C2	TX00040
10	APP00036	2024-07-10	Pending	Driving License	Delhi Janakpuri	2024-05-24 16:49:00	C2	TX00036

11. Find the count of total appointments for each RTO location (city and state), ordered by the number of appointments in descending order.

```
Query  Query History
1  SELECT vr."registration_no", vr."vehicle_type", vr."model", vr."year_of_manufacture", COUNT(vo."owner_name") AS "number_of_owners"
2  FROM "Vehicle_Registration" vr
3  LEFT JOIN "Vehicle_Owner" vo ON vr."registration_no" = vo."registration_no"
4  GROUP BY vr."registration_no", vr."vehicle_type", vr."model", vr."year_of_manufacture"
5  HAVING COUNT(vo."owner_name") > 1
6  ORDER BY "number_of_owners" DESC;
```

Data Output | Messages | Notifications

	registration_no [PK] character varying (20)	vehicle_type character varying (50)	model character varying (100)	year_of_manufacture integer	number_of_owners bigint
1	PB93TC9731	Two Wheeler	Honda Activa	2021	2
2	KA92TC4962	Four Wheeler	Hyundai i10	2020	2
3	MH48TC2654	Two Wheeler	Royal Enfield Classic	2020	2
4	MH44TC6859	Two Wheeler	Royal Enfield Classic	2013	2

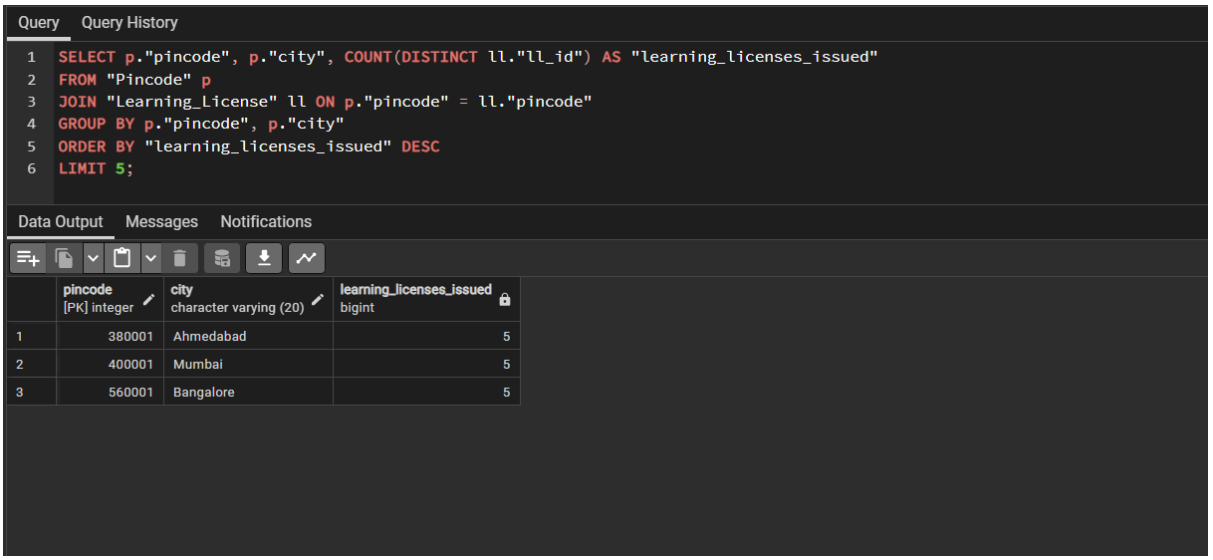
12. Find the roll numbers, test scores, invigilators, feedback, test dates, and counter types for tests where the score is less than 40 and feedback is provided, ordered by the test scores in ascending order.

```
Query  Query History
1  SELECT t."roll_no", t."test_score", t."invigilator", t."feedback", t."date", c."counter_type"
2  FROM "Test" t
3  JOIN "Counter" c ON t."computer_no" = c."counter_no"
4  WHERE t."test_score" < 40 AND t."feedback" IS NOT NULL
5  ORDER BY t."test_score" ASC;
```

Data Output | Messages | Notifications

	roll_no character varying (20)	test_score integer	invigilator character varying (15)	feedback character varying (30)	date date	counter_type character varying (5)
1	RL00012	2	Dr. Kumar	Excellent performance	2023-12-21	D
2	RL00004	14	Ms. Joshi	Well prepared	2023-04-11	A
3	RL00024	14	Ms. Joshi	Well prepared	2023-04-11	A
4	RL00030	18	Mrs. Patel	Satisfactory	2023-12-26	D
5	RL00010	18	Mrs. Patel	Satisfactory	2023-12-26	D
6	RL00028	32	Mrs. Patel	Poor understanding	2023-05-04	B
7	RL00008	32	Mrs. Patel	Poor understanding	2023-05-04	B
8	RL00027	36	Prof. Singh	None	2023-04-04	B
9	RL00007	36	Prof. Singh	None	2023-04-04	B
10	RL00025	38	Mr. Sharma	Satisfactory	2024-02-09	D
11	RL00005	38	Mr. Sharma	Satisfactory	2024-02-09	D
12	RL00001	39	Ms. Joshi	None	2023-07-18	C
13	RL00021	39	Ms. Joshi	None	2023-07-18	C

13. Find the top 5 pincodes and cities with the highest number of learning licenses issued, displaying the pincode, city name, and the count of learning licenses.

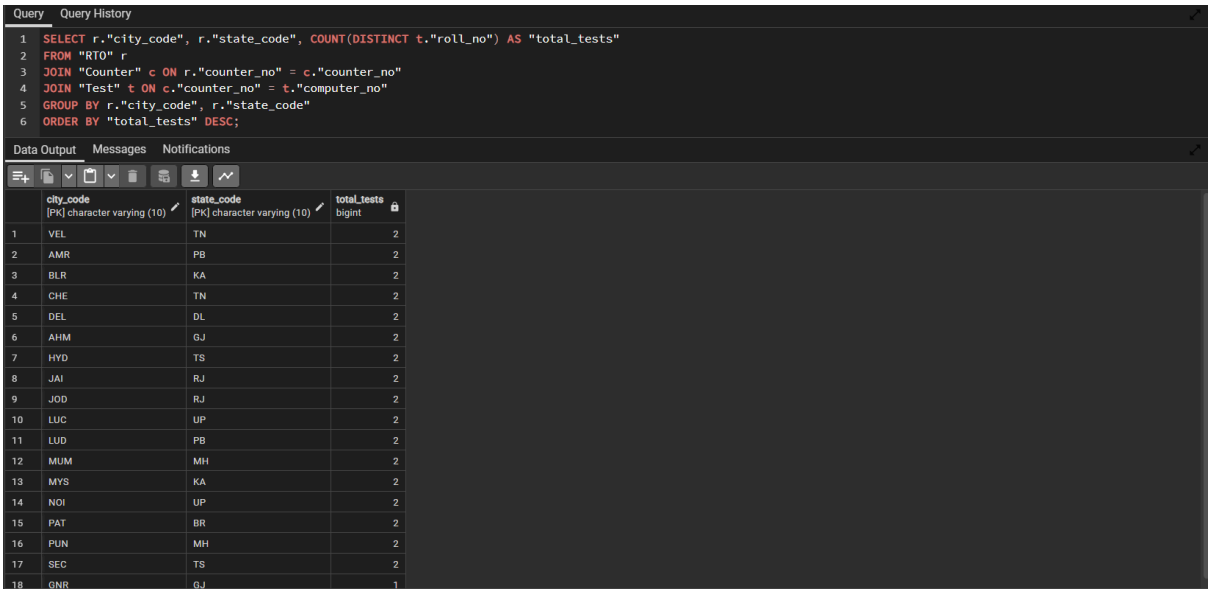


```
Query  Query History
1 SELECT p."pincode", p."city", COUNT(DISTINCT ll."ll_id") AS "learning_licenses_issued"
2 FROM "Pincode" p
3 JOIN "Learning_License" ll ON p."pincode" = ll."pincode"
4 GROUP BY p."pincode", p."city"
5 ORDER BY "learning_licenses_issued" DESC
6 LIMIT 5;
```

Data Output | Messages | Notifications

	pincode [PK] integer	city character varying (20)	learning_licenses_issued bigint
1	380001	Ahmedabad	5
2	400001	Mumbai	5
3	560001	Bangalore	5

14. This query retrieves the count of total tests conducted for each RTO location (city and state), ordered by the number of tests in descending order.



```
Query  Query History
1 SELECT r."city_code", r."state_code", COUNT(DISTINCT t."roll_no") AS "total_tests"
2 FROM "RTO" r
3 JOIN "Counter" c ON r."counter_no" = c."counter_no"
4 JOIN "Test" t ON c."counter_no" = t."computer_no"
5 GROUP BY r."city_code", r."state_code"
6 ORDER BY "total_tests" DESC;
```

Data Output | Messages | Notifications

	city_code [PK] character varying (10)	state_code [PK] character varying (10)	total_tests bigint
1	VEL	TN	2
2	AMR	PB	2
3	BLR	KA	2
4	CHE	TN	2
5	DEL	DL	2
6	AHM	GJ	2
7	HYD	TS	2
8	JAI	RJ	2
9	JOD	RJ	2
10	LUC	UP	2
11	LUD	PB	2
12	MUM	MH	2
13	MYS	KA	2
14	NOI	UP	2
15	PAT	BR	2
16	PUN	MH	2
17	SEC	TS	2
18	GNR	GJ	1

15. Retrieve all vehicle types and the count of vehicles registered for each type, limited to the top 5 most common vehicle types

Query Query History

```
1  SELECT vehicle_type, COUNT(*) as total
2  FROM "Vehicle_Registration"
3  GROUP BY vehicle_type
4  ORDER BY total DESC
5  LIMIT 5;
```

Data Output Messages Notifications

	vehicle_type character varying (50)	total bigint
1	Four Wheeler	9
2	Two Wheeler	7

16. Retrieve the last 10 payments made for renewals

Query Query History

```
1  SELECT RP.renewal_txn_id
2  FROM "Renewal_Payment" RP
3  ORDER BY RP.timestamp DESC
4  LIMIT 10;
```

Data Output Messages Notifications

	renewal_txn_id [PK] character varying (20)
1	TX00010
2	TX00002
3	TX00015
4	TX00012
5	TX00014
6	TX00005
7	TX00008
8	TX00001
9	TX00003
10	TX00007

17. Select the test scores of users along with their average score

```
SELECT T.roll_no, T.test_score, A.avg_score
FROM "Test" T,
(SELECT roll_no, AVG(test_score) as avg_score FROM "Test" GROUP BY roll_no) A
WHERE T.roll_no = A.roll_no;
```

	roll_no [PK] character varying (20)	test_score integer	avg_score numeric
1	RL00001	39	39.0000000000000000
2	RL00002	50	50.0000000000000000
3	RL00003	86	86.0000000000000000
4	RL00004	14	14.0000000000000000
5	RL00005	38	38.0000000000000000
6	RL00006	55	55.0000000000000000
7	RL00007	36	36.0000000000000000
8	RL00008	32	32.0000000000000000
9	RL00009	85	85.0000000000000000
10	RL00010	18	18.0000000000000000
11	RL00011	54	54.0000000000000000
12	RL00012	2	2.0000000000000000
13	RL00013	80	80.0000000000000000
14	RL00014	92	92.0000000000000000
15	RL00015	42	42.0000000000000000
16	RL00016	74	74.0000000000000000
17	RL00017	47	47.0000000000000000
18	RL00018	57	57.0000000000000000
19	RL00019	43	43.0000000000000000

18. Find the top 3 counter types with the highest average test scores, displaying the counter type and the average test score rounded to 2 decimal places.

```
SELECT c."counter_type", ROUND(AVG(t."test_score"), 2) AS "avg_test_score"
FROM "Counter" c
JOIN "Test" t ON c."counter_no" = t."computer_no"
GROUP BY c."counter_type"
ORDER BY "avg_test_score" DESC
LIMIT 3;
```

	counter_type character varying (5)	avg_test_score numeric
1	C	59.11
2	B	47.00
3	D	46.64

19. Find all users who have the same first name and last name but different pincodes.

Query | Query History

```
1  SELECT u1.first_name, u1.last_name
2  FROM "User" u1
3  JOIN "User" u2 ON u1.first_name = u2.first_name AND u1.last_name = u2.last_name
4  WHERE u1.pincode <> u2.pincode
5  GROUP BY u1.first_name, u1.last_name
6  HAVING COUNT(DISTINCT u1.pincode) > 1;
7
```

Data Output | Messages | Notifications

	first_name character varying (20)	last_name character varying (20)
1	Aditya	Singh
2	Vivaan	Mehra
3	Vivaan	Mehta

20. Get all RTO offices in the state Maharashtra with the highest number of employees

Query | Query History

```
1  SELECT *
2  FROM "RTO"
3  WHERE state_code = 'MH'
4  ORDER BY number_of_employees DESC;
5
```

Data Output | Messages | Notifications

	city_code [PK] character varying (10)	state_code [PK] character varying (10)	pincode integer	number_of_employees integer	roll_no character varying (20)	counter_no character varying (5)
1	MUM	MH	400001	198	RL00001	C3
2	PUN	MH	410014	192	RL00011	C2

21. Retrieve the vehicle details for all vehicles that have a permit expiring in the next month.

```
SELECT vr.*
FROM "Vehicle_Registration" vr
JOIN "Permit" p ON vr.user_id = p.user_id
WHERE p.validity BETWEEN CURRENT_DATE AND (CURRENT_DATE + INTERVAL '1 month');
```

Data Output | Messages | Notifications

registration_no [PK] character varying (20)	registration_date date	vehicle_type character varying (50)	model character varying (100)	year_of_manufacture integer	engine_no character varying (50)	chassis_no character varying (50)	color character varying (50)	temp_reg_no character varying (20)	user_id character

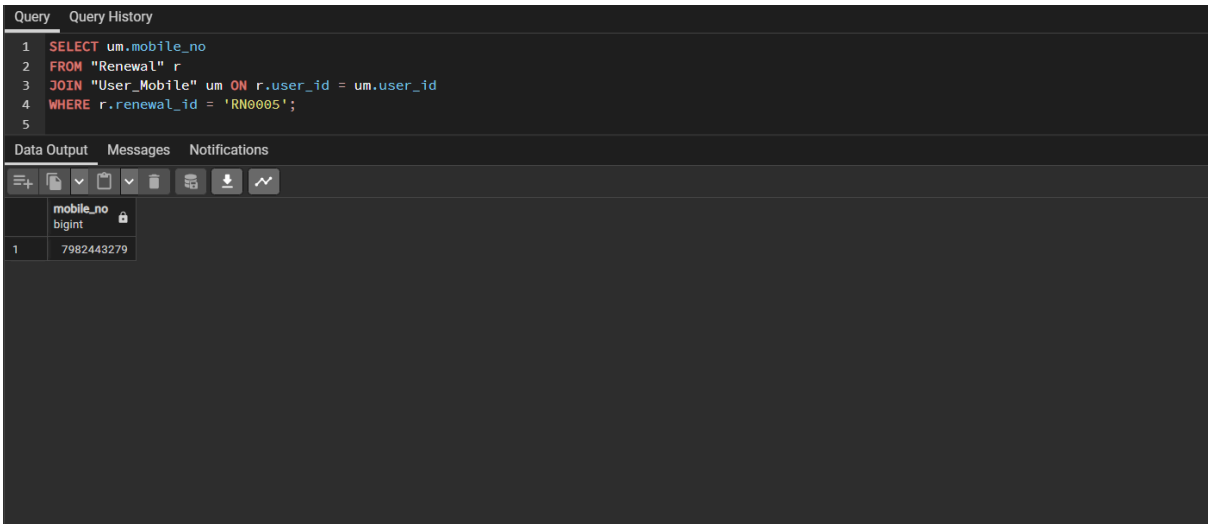
22. List all RTOs along with the number of employees and the count of permits issued

```
SELECT r.city_code, r.state_code, r.number_of_employees, COUNT(p.permit_no) AS permits_issued
FROM "RTO" r
LEFT JOIN "Permit" p ON r.roll_no = p.user_id
GROUP BY r.city_code, r.state_code, r.number_of_employees;
```

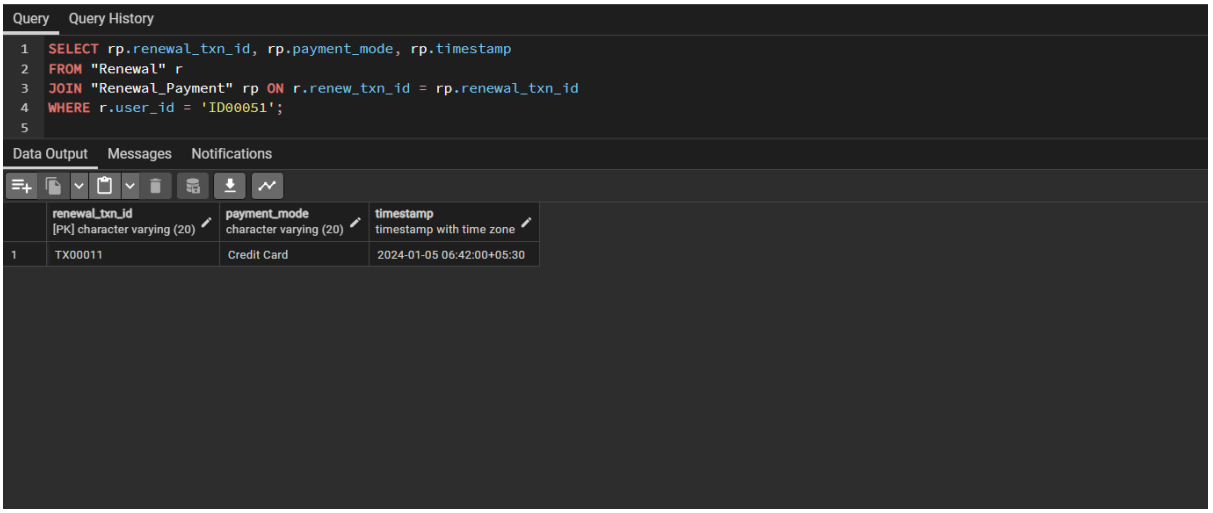
Data Output | Messages | Notifications

	city_code [PK] character varying (10)	state_code [PK] character varying (10)	number_of_employees integer	permits_issued bigint
1	HYD	TS	96	0
2	PUN	MH	192	0
3	CHE	TN	103	0
4	AHM	GJ	274	0
5	JOD	RJ	65	0
6	JAI	RJ	65	0
7	PAT	BR	74	0
8	GNR	GJ	234	0
9	BLR	KA	156	0
10	VEL	TN	102	0
11	NOI	UP	60	0
12	DEL	DL	65	0
13	MYS	KA	150	0
14	SEC	TS	50	0
15	MUM	MH	198	0
16	AMR	PB	235	0
17	LUC	UP	66	0
18	LUD	PB	115	0

23. Retrieve the mobile number of a user with renewal id RN0005 who has applied for a renewal



24. Get the renewal payment details for the user with id ID00051:



25. Retrieve all renewals along with their corresponding payments and the associated user's mobile number:

```
Query  Query History
1  SELECT r.renewal_id, r.dl_no, r.renew_date, um.mobile_no
2  FROM "Renewal" r
3  JOIN "Renewal_Payment" rp ON r.renew_txn_id = rp.renewal_txn_id
4  JOIN "User_Mobile" um ON r.user_id = um.user_id;
5
```

Data Output Messages Notifications

	renewal_id character varying (20)	dl_no character varying (15)	renew_date date	mobile_no bigint
1	RN0001	DL542880650	2024-01-07	7587109580
2	RN0002	DL021932850	2024-01-19	8980155951
3	RN0003	DL938509366	2024-01-02	7180328735
4	RN0004	DL013845706	2024-01-26	4889656942
5	RN0005	DL460121533	2024-03-02	7982443279
6	RN0006	DL147837428	2024-02-13	7675184103
7	RN0007	DL775892931	2024-03-30	2388912282
8	RN0008	DL766628134	2024-03-16	5085517567
9	RN0009	DL458686123	2024-03-18	8687610231
10	RN0010	DL929949572	2024-03-11	2487774877
11	RN0011	DL524773715	2024-01-21	4583935031
12	RN0012	DL007807739	2024-02-21	1489371643
13	RN0013	DL316378362	2024-06-22	9880140149
14	RN0014	DL925014571	2024-08-31	5584407599
15	RN0015	DL302808457	2024-07-04	1588167516

26. Find the RTOs where the number of employees is greater than the average number of employees across all RTOs

```
Query  Query History
1  SELECT r.city_code, r.state_code, r.number_of_employees
2  FROM "RTO" r
3  WHERE r.number_of_employees > (
4      SELECT AVG(number_of_employees) FROM "RTO"
5  );
6
```

Data Output Messages Notifications

	city_code [PK] character varying (10)	state_code [PK] character varying (10)	number_of_employees integer
1	MUM	MH	198
2	BLR	KA	156
3	AHM	GJ	274
4	AMR	PB	235
5	PUN	MH	192
6	MYS	KA	150
7	GNR	GJ	234

27. Get the users who have applied for a renewal but haven't paid yet

Query | Query History

```
SELECT u.user_id, u.first_name, u.last_name, r.renewal_id
FROM "User" u
JOIN "Renewal" r ON u.user_id = r.user_id
LEFT JOIN "Renewal_Payment" rp ON r.renew_txn_id = rp.renewal_txn_id
WHERE rp.renewal_txn_id IS NULL;
```

Data Output | Messages | Notifications

user_id character varying (20)	first_name character varying (20)	last_name character varying (20)	renewal_id character varying (20)

(without paying it is not possible to get an appointment for renewal, this is the reason why there is no output)

28. List all applied-to entries along with the corresponding learning license appointment and user details

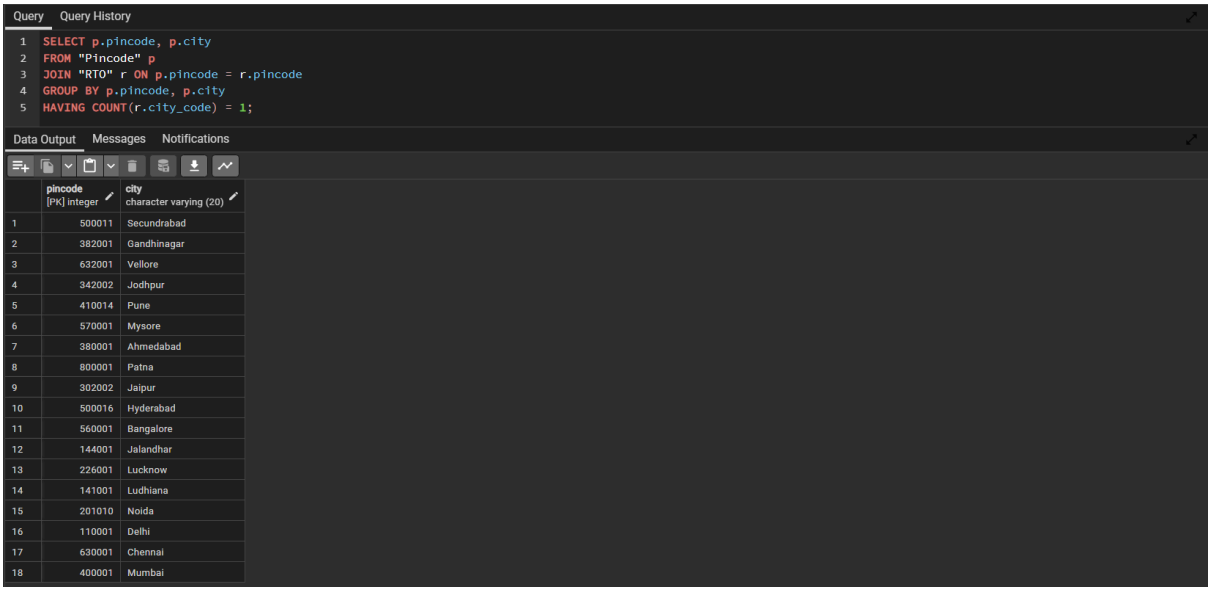
Query | Query History

```
SELECT at.appointment_no, at.roll_no, at.city_code, at.state_code, a.date, u.first_name, u.last_name, um.mobile_no
FROM "Applied_To" at
JOIN "Appointment" a ON at.appointment_no = a.appointment_no
JOIN "Learning_License" ll ON a.app_txn_id = ll.app_txn_id
JOIN "User" u ON ll.user_id = u.user_id
JOIN "User_Mobile" um ON u.user_id = um.user_id;
```

Data Output | Messages | Notifications

	appointment_no character varying (20)	roll_no character varying (20)	city_code character varying (10)	state_code character varying (10)	date date	first_name character varying (20)	last_name character varying (20)	mobile_no bigint
1	APP00016	RL00001	BLR	KA	2024-09-03	Arjun	Jain	7557109580
2	APP00017	RL00002	BLR	KA	2024-06-10	Vivaan	Mehra	8910155951
3	APP00018	RL00003	MUM	MH	2025-03-09	Siddharth	Kumar	7110328735
4	APP00019	RL00004	BLR	KA	2024-09-03	Ansh	Jain	4899656942
5	APP00020	RL00005	BLR	KA	2024-06-10	Ansh	Kapoor	7992443279
6	APP00021	RL00006	MUM	MH	2025-03-09	Aryan	Jain	7685184103
7	APP00022	RL00007	BLR	KA	2024-09-03	Aarav	Gupta	2378912282
8	APP00023	RL00008	BLR	KA	2024-06-10	Dhruv	Patel	5005517567
9	APP00024	RL00009	MUM	MH	2025-03-09	Aditya	Singh	8697610231
10	APP00025	RL00010	BLR	KA	2024-09-03	Arjun	Malhotra	2407774877
11	APP00026	RL00011	BLR	KA	2024-06-10	Arjun	Gupta	4593935031
12	APP00027	RL00012	MUM	MH	2025-03-09	Vivaan	Mehra	1499371643
13	APP00028	RL00013	BLR	KA	2024-09-03	Ishaan	Kapoor	9810140149
14	APP00029	RL00014	BLR	KA	2024-06-10	Ishaan	Sharma	5524407599
15	APP00030	RL00015	MUM	MH	2025-03-09	Dhruv	Jain	1568167516

29. List all pincode entries along with the cities having exactly one RTO

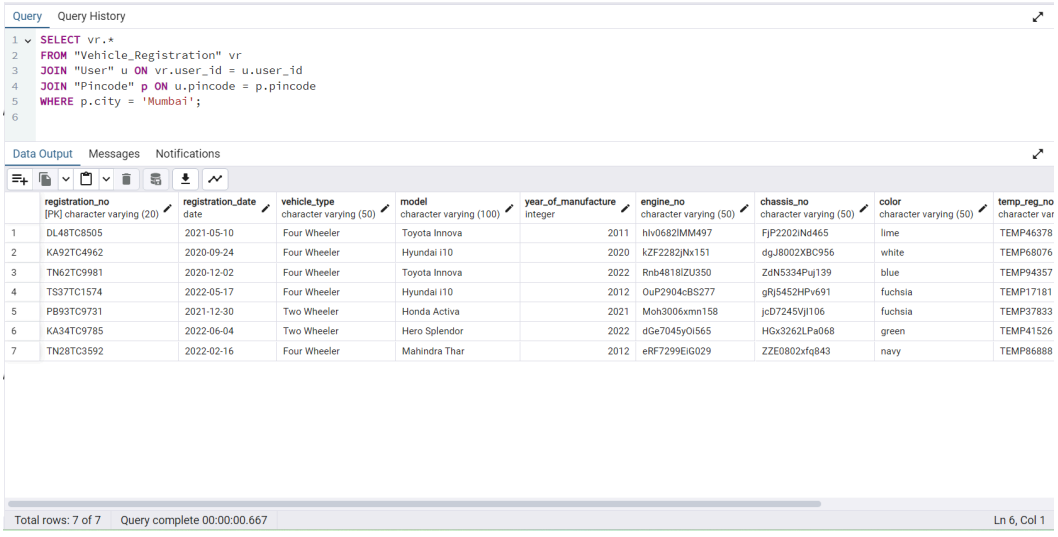


```
Query  Query History
1  SELECT p.pincode, p.city
2  FROM "Pincode" p
3  JOIN "RTO" r ON p.pincode = r.pincode
4  GROUP BY p.pincode, p.city
5  HAVING COUNT(r.city_code) = 1;
```

Data Output Messages Notifications

	pincode [PK] integer	city character varying (20)
1	500011	Secundrabad
2	382001	Gandhinagar
3	632001	Vellore
4	342002	Jodhpur
5	410014	Pune
6	570001	Mysore
7	380001	Ahmedabad
8	800001	Patna
9	302002	Jaipur
10	500016	Hyderabad
11	560001	Bangalore
12	144001	Jalandhar
13	226001	Lucknow
14	141001	Ludhiana
15	201010	Noida
16	110001	Delhi
17	630001	Chennai
18	400001	Mumbai

30. List the details of all vehicle registrations in the Mumbai city



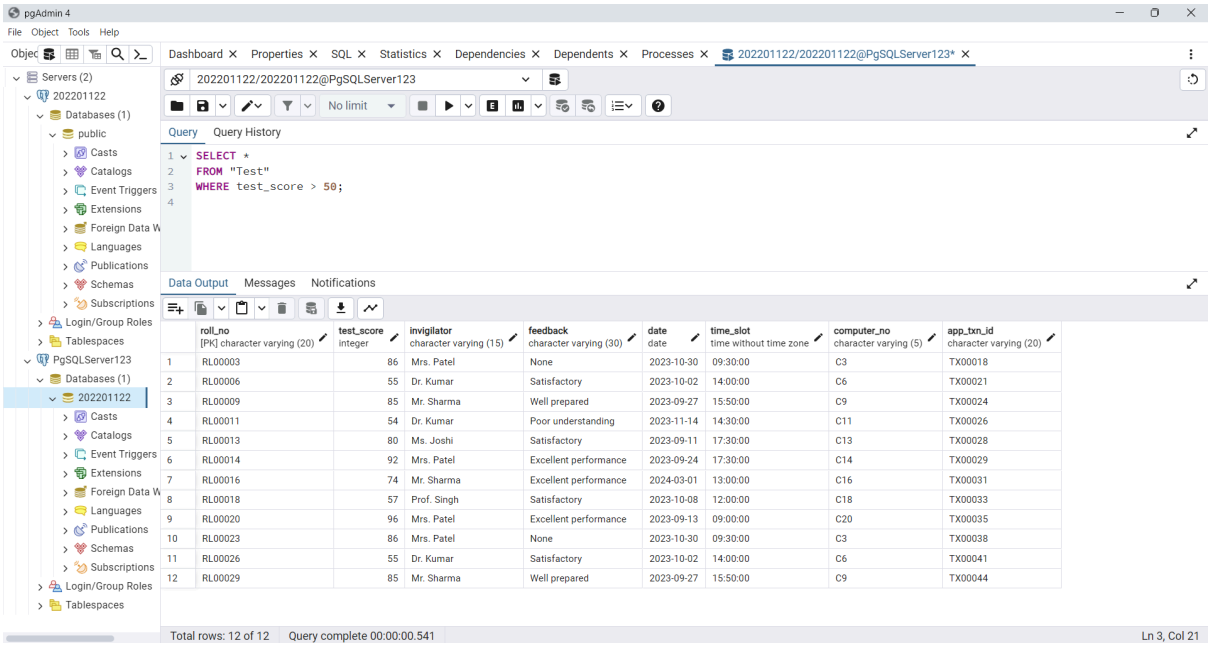
```
Query  Query History
1  SELECT vr.*
2  FROM "Vehicle_Registration" vr
3  JOIN "User" u ON vr.user_id = u.user_id
4  JOIN "Pincode" p ON u.pincode = p.pincode
5  WHERE p.city = 'Mumbai';
6
```

Data Output Messages Notifications

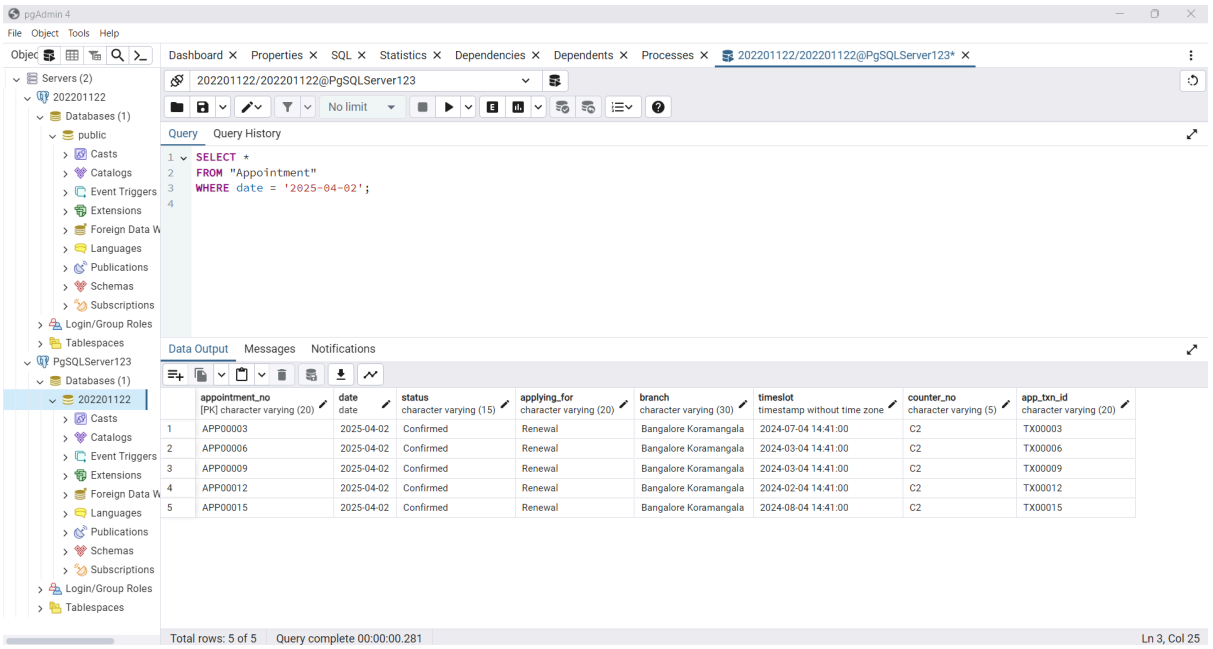
	registration_no [PK] character varying (20)	registration_date date	vehicle_type character varying (50)	model character varying (100)	year_of_manufacture integer	engine_no character varying (50)	chassis_no character varying (50)	color character varying (50)	temp_reg_no character var
1	DL48TC8505	2021-05-10	Four Wheeler	Toyota Innova	2011	hlv0682IMM497	FjP2202iNd465	lime	TEMP46378
2	KA92TC4962	2020-09-24	Four Wheeler	Hyundai i10	2020	kZF2282jNx151	dgJ8002XBC956	white	TEMP68076
3	TN62TC9981	2020-12-02	Four Wheeler	Toyota Innova	2022	Rnb4818IZU350	ZdN5334Puj139	blue	TEMP94357
4	TS37TC1574	2022-05-17	Four Wheeler	Hyundai i10	2012	OuP2904cBS277	gRj5452HPv691	fuchsia	TEMP17181
5	PB93TC9731	2021-12-30	Two Wheeler	Honda Activa	2021	Moh3006xmn158	jcD7245Vjl106	fuchsia	TEMP37833
6	KA34TC9785	2022-06-04	Two Wheeler	Hero Splendor	2022	dGe7045yOi565	HGx3262LPa068	green	TEMP41526
7	TN28TC3592	2022-02-16	Four Wheeler	Mahindra Thar	2012	eRF7299EiG029	ZZE0802xfq843	navy	TEMP86888

Total rows: 7 of 7 Query complete 00:00:00.667 Ln 6, Col 1

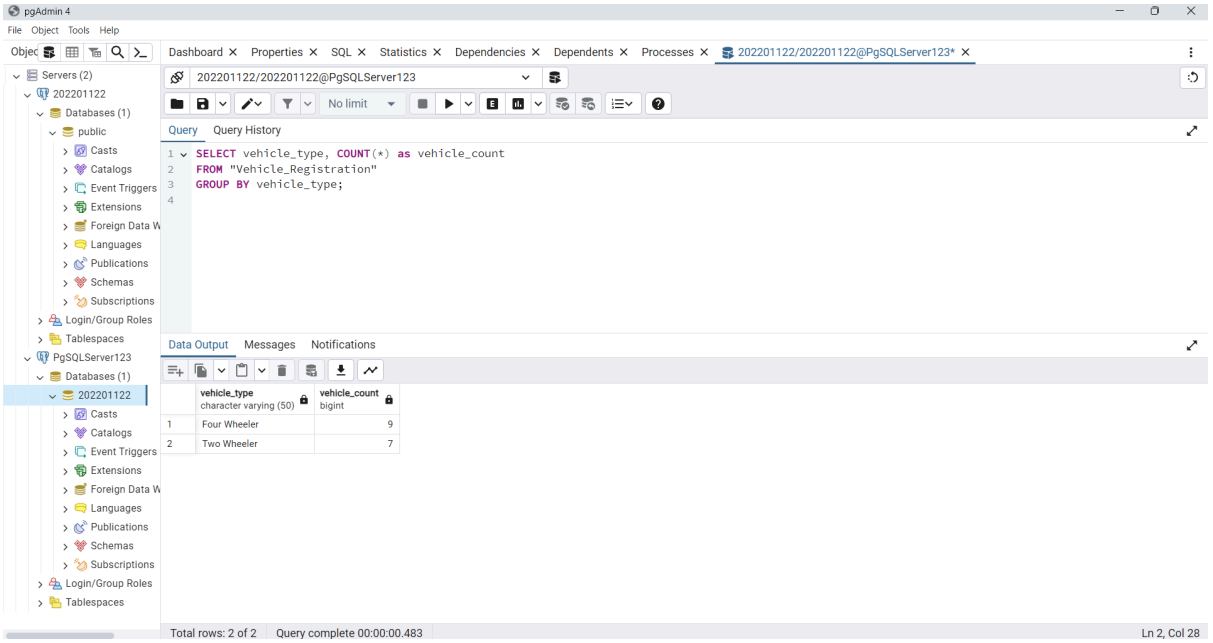
31. Find all tests that have a score above a threshold of 50



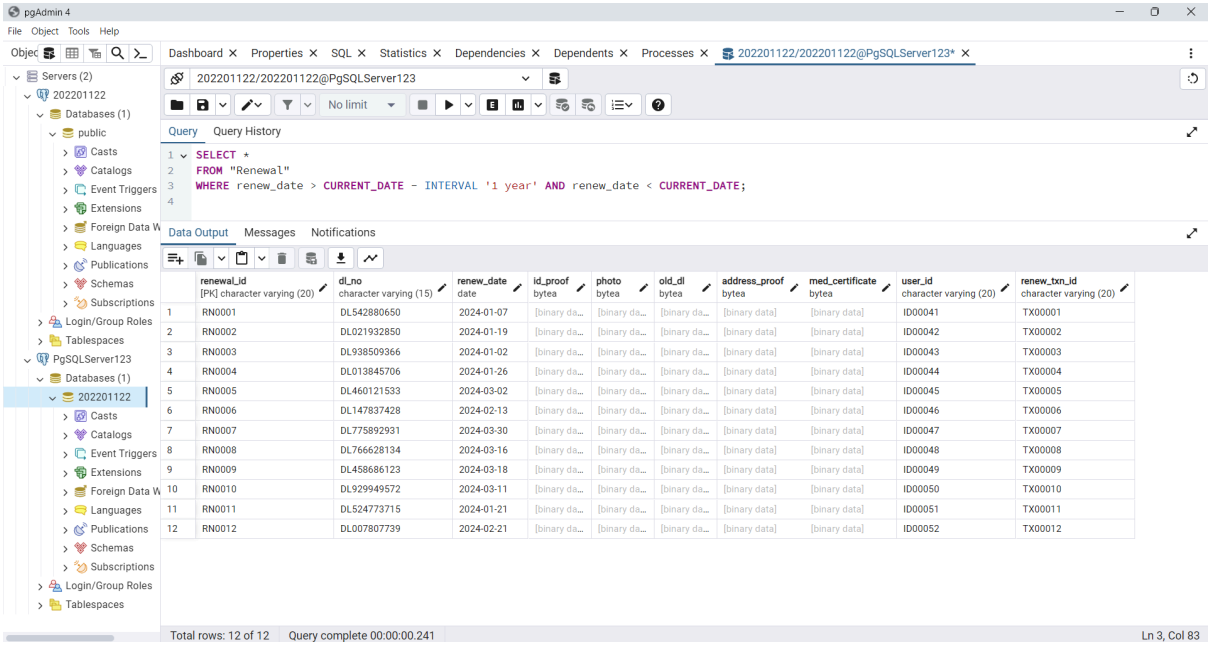
32. Show all the appointments scheduled for a particular date(2025-04-02)



33. Count the number of vehicles registered under each type



34. Fetch all renewal applications that have been made within the last year



35. Provide a report showing the performance metrics of each counter in our system, including the total number of appointments processed by each counter and the average test scores of the applicants they have supervised

Query Query History

```sql
SELECT
    c.counter_no,
    c.counter_type,
    c.employee_name,
    COUNT(DISTINCT a.appointment_no) AS total_appointments,
    AVG(t.test_score) AS avg_test_score
FROM
    "Counter" c
LEFT JOIN
    "Appointment" a ON c.counter_no = a.counter_no
LEFT JOIN
    "Applied_To" ato ON a.appointment_no = ato.appointment_no
LEFT JOIN
    "Test" t ON ato.roll_no = t.roll_no
GROUP BY
    c.counter_no, c.counter_type, c.employee_name;
```

Data Output Messages Notifications

	counter_no [PK] character varying (5)	counter_type character varying (5)	employee_name character varying (20)	total_appointments bigint	avg_test_score numeric
1	C1	C	Amit Patel	0	[null]
2	C10	D	Lakshmi Goyal	19	53.6000000000000000
3	C11	B	Neha Chauhan	0	[null]
4	C12	D	Sanjay Mehta	0	[null]
5	C13	D	Deepa Pillai	0	[null]
6	C14	B	Rohan Gupta	0	[null]
7	C15	A	Shreya Iyer	0	[null]
8	C16	C	Vikas Dhawan	0	[null]
9	C17	D	Ritu Bajaj	0	[null]
10	C18	C	Suresh Prasad	0	[null]
11	C19	C	Anjali Verma	0	[null]
12	C2	D	Sunita Rao	14	39.0000000000000000
13	C20	A	Harish Nair	0	[null]
14	C3	D	Rajesh Kumar	5	[null]
15	C4	A	Meena Sharma	4	85.0000000000000000
16	C5	D	Karan Singh	0	[null]
17	C6	C	Priya Desai	4	[null]
18	C7	B	Vijay Joshi	4	[null]
19	C8	B	Anita Reddy	18	57.0000000000000000
20	C9	C	Mohit Agarwal	8	43.0000000000000000