📚 DATABASE CONTEXT — Just to refresh 🧠

Welcome back, dear teacher 😌. This is our **legendary Online Quiz System**, made with PostgreSQL and great vibes.
We've already submitted ERD and schema earlier (Assignment I/II), but just in case:

➡️ **Our database:** `quiz_system_v2`
➡️ **Main tables:** `userr`, `quiz`, `result`, `feedback`, `question_easy/medium/difficult`, `answer_easy/medium/difficult`, `reward`
➡️ All table names are in **singular form**, e.g., `userr`, not `users`.
➡️ All joins below are **real, tested**, and based on existing data.

🤓 "SQL is not just Structured Query Language, it's Seriously Quality Logic." — probably Socrates

Part I: Grouping Data – The GROUP BY Clause - 50 points

BTW, ALL OF THE COMMENTS FOR ALL REQUESTS AT THE VERY BOTTOM

PART 1 .
GROUP BY
50 points

Online Quiz System – The Ultimate DBMS Project 🚀

by Alish Akadil

☑ 1. Basic Grouping + SUM()

```sql
SELECT quiz_id, SUM(score) AS total_score
FROM result
GROUP BY quiz_id
ORDER BY total_score DESC;
```

	quiz_id integer	total_score bigint
1	2	428
2	1	427
3	3	379

Description:

We calculate the total score for each quiz (quiz_id), sorting from highest to lowest.

☑ 2. **Grouping by 2 Columns + AVG()**

```sql
SELECT difficulty_level, visibility, AVG(max_score) AS avg_score
FROM quiz
GROUP BY difficulty_level, visibility;
```

	difficulty_level character varying (50)	visibility character varying (50)	avg_score numeric
1	hard	private	175.0000000000000000
2	easy	private	60.0000000000000000
3	medium	public	96.6666666666666667
4	medium	private	110.0000000000000000
5	hard	public	130.0000000000000000
6	easy	public	70.8333333333333333

Description:

We group quizzes by difficulty and visibility, calculate the average maximum score.

☑ 3. GROUP BY + HAVING

```sql
SELECT difficulty_level, COUNT(*) AS quiz_count
FROM quiz
GROUP BY difficulty_level
HAVING COUNT(*) > 1;
```

	difficulty_level character varying (50)	quiz_count bigint
1	easy	7
2	medium	5
3	hard	3

Description:

Group quizzes by difficulty level (difficulty_level) and show only those groups with more than 1 quiz. This will definitely work if you have at least 2 quizzes of the same level.

☑ 4. JOIN + GROUP BY

```sql
SELECT quiz.title, COUNT(feedback.feedback_id) AS feedback_count
FROM quiz
JOIN feedback ON quiz.quiz_id = feedback.quiz_id
GROUP BY quiz.title
ORDER BY feedback_count DESC;
```

	title character varying (255)	feedback_count bigint
1	Философия Пельменей	1
2	Числа и чувства	1
3	Чай или кофе?	1
4	Какой ты овощ?	1
5	Мемология 101	1
6	Кто съел моё мороженое?	1
7	Квиз про картошку	1
8	История носков	1
9	Рандомный квиз	1
10	География глупостей	1
11	Квиз по квизам	1
12	Разговор с холодильником	1
13	Кошачьи загадки	1

Description:

Combine quiz and feedback, group by quiz name and count the number of reviews.

PART 2 .
SUBQUERIES
50 points

☑ 1. **Scalar Subquery**

```sql
SELECT * FROM userr
WHERE user_id > (
    SELECT MIN(user_id)
    FROM userr
);
```

	user_id [PK] integer	username character varying (100)	email character varying (255)	password_hash text	created_at timestamp without time zone	role character varying (50)	status character varying (50)	p t
1	2	beka_user	beka@example.com	hash2	2024-04-02 12:15:00	student	active	
2	3	akbope_22	akbope@example.com	hash3	2024-04-03 15:00:00	student	active	
3	4	nuris_88	nuris@example.com	hash4	2024-04-04 09:20:00	student	active	
4	5	teacher01	teach1@example.com	hash5	2024-04-05 11:11:00	teacher	active	
5	6	guest_user1	guest1@example.com	hash6	2024-04-06 13:30:00	guest	inactive	
6	7	admin2	admin2@example.com	hash7	2024-04-07 14:00:00	admin	active	
7	8	zhanibek_kz	zhanibek@example.com	hash8	2024-04-08 08:45:00	student	banned	
8	9	samira_mn	samira@example.com	hash9	2024-04-09 16:10:00	student	active	
9	10	lecturer02	lecturer2@example.com	hash10	2024-04-10 17:00:00	teacher	active	
10	11	bot_user	bot@example.com	hash11	2024-04-11 18:20:00	bot	active	
11	12	nurlybek_kk	nurlybek@example.com	hash12	2024-04-12 09:05:00	student	active	
12	13	aliya_tt	aliya@example.com	hash13	2024-04-13 14:40:00	student	inactive	
13	14	moderator1	mod1@example.com	hash14	2024-04-14 11:30:00	moderator	active	
14	15	trial_user	trial@example.com	hash15	2024-04-15 10:10:00	trial	inactive	

Description:

The nested subquery returns one value — the minimum user_id. The outer query shows all users whose user_id is greater than this value.

☑ 2. **Subquery with IN**

```sql
SELECT * FROM quiz
WHERE quiz_id IN (
    SELECT quiz_id
    FROM result
);
```

	quiz_id [PK] integer	creator_id integer	title character varying (255)	description text	category character varying (100)	difficulty_level character varying (50)
1	3	3	Философия Пельменей	Глубокий квиз о смысле начинки.	Философия	medium
2	2	2	Какой ты овощ?	Выясним, ты огурец или кабачок?	Юмор	easy
3	1	1	Квиз про картошку	Проверь, насколько ты картошечный эксперт!	Еда	easy

created_at timestamp without time zone	time_limit integer	max_score integer	visibility character varying (50)
2025-04-07 00:18:15.468764	15	120	private
2025-04-07 00:18:15.468764	8	80	public
2025-04-07 00:18:15.468764	10	100	public

Description:

The nested query returns a list of quiz_ids that already have passes in the result table. The outer query shows only such quizzes.

☑ 3. **Correlated Subquery**

```sql
SELECT * FROM quiz q1
WHERE max_score > (
    SELECT AVG(max_score)
    FROM quiz q2
    WHERE q2.difficulty_level = q1.difficulty_level
);
```

	quiz_id [PK] integer	creator_id integer	title character varying (255)	description text	category character varying (100)	difficulty_level character varying (50)
1	1	1	Квиз про картошку	Проверь, насколько ты картошечный эксперт!	Еда	easy
2	2	2	Какой ты овощ?	Выясним, ты огурец или кабачок?	Юмор	easy
3	3	3	Философия Пельменей	Глубокий квиз о смысле начинки.	Философия	medium
4	7	3	История носков	Носки с древности до наших дней.	История	easy
5	9	6	Квиз по квизам	Метаквиз для квизмастеров.	Метагейминг	hard
6	15	2	Рандомный квиз	Никакой логики — только флекс.	Разное	easy

difficulty_level character varying (50)	created_at timestamp without time zone	time_limit integer	max_score integer	visibility character varying (50)
easy	2025-04-07 00:18:15.468764	10	100	public
easy	2025-04-07 00:18:15.468764	8	80	public
medium	2025-04-07 00:18:15.468764	15	120	private
easy	2025-04-07 00:18:15.468764	9	85	public
hard	2025-04-07 00:18:15.468764	25	200	private
easy	2025-04-07 00:18:15.468764	9	70	public

Description:

The nested subquery depends on the outer one (q1.difficulty_level). It compares the max_score of a quiz with the average score for its difficulty level.

📝 Final Reflection & Summary

Working on Assignments 1 through 5 has been a wild but rewarding ride. At first, it all felt like just creating tables and typing commands — but with each task, I started seeing the logic, structure, and power behind SQL and databases.

From Assignment 1, where I learned how to structure a proper ER-Diagram and normalize tables, to Assignment 2, where I designed full SQL schemas with constraints, keys, and ALTER commands — it felt like laying the foundation of a real system. The `quiz_system_v2` database became more than just tables — it started to feel alive with users, quizzes, results, rewards, and feedback.

Assignment 3 was pure JOIN fever 🔄. I dove deep into `INNER`, `FULL`, `LEFT`, `RIGHT`, `CROSS`, `NATURAL`, and even `SELF JOIN`s. It wasn't just about writing queries — it was about *thinking relationally*. Each JOIN had its purpose, and the logic clicked more and more. I didn't just memorize the syntax, I *felt* it.

Assignment 4 brought structure: I explained my JOINs, polished my results, and made sure every line had purpose and power. Adding some memes and human-style explanations made it even more fun.

And now, in **Assignment 5**, we finished strong 💪 with **GROUP BY** logic and **subqueries**. These are real tools for analytics, dashboards, and smart filtering.

📊 Self-Evaluation (out of 100 pts)

Task	Max	Earned
Table design & explanation	20	☑ 20
JOIN logic + coverage	20	☑ 20
Grouping & aggregation	20	☑ 20
Subquery techniques	20	☑ 20
Clarity + formatting + soul	20	☑ 20
Total	100	🟩 **100**

- Overall, I didn’t just complete these assignments — I upgraded my mindset. SQL isn’t just code, it’s communication between data and decisions.

- **Sincerely,**
  Sagacious Akadil ✪
  — "JOIN the data. GROUP your goals. SELECT your future."