

Online Quiz System – The Ultimate DBMS Project



by Alish Akadil

WHERE:

1.

Query Query History

```
1 v SELECT *
2   FROM quiz
3 WHERE category = 'Математика';
4
```

Data Output Messages Notifications

Showing rows: 1 to 1 | | Page No: 1 of 1 |

	quiz_id	creator_id	title	description	category	difficulty_level	created_at	time_limit	max_si
1	13	3	Числа и чувства	Математика с элементами драмы.	Математика	hard	2025-04-07 00:18:15.468764	18	

2.

Query Query History

```
1 v SELECT *
2   FROM question_easy
3 WHERE points > 1;
4
```

Data Output Messages Notifications

Showing rows: 1 to 4 | | Page No: 1 of 1 |

	question_id	quiz_id	question_text	created_at	points	explanation
1	8	2	Why was the math book sad?	2025-04-07 00:38:47.1288	2	A timeless meme among students
2	9	3	What's the deal with airplane food?	2025-04-07 00:38:47.1288	2	This one's been used in many situations
3	10	4	I asked the librarian if she had any books on paranoia. She whispered, "They're right behind you."	2025-04-07 00:38:47.1288	2	Perfect for library lovers
4	11	5	W had to listen to you?	2025-04-07 00:38:47.1288	2	Talk to them in memes

3.

Query Query History

```
1 ✓ SELECT *
2   FROM feedback
3 WHERE feedback_category = 'General' AND rating >= 4;
4
```

Data Output Messages Notifications

	feedback_id [PK] integer	user_id integer	quiz_id integer	rating integer	comment text	submitted_at timestamp without time zone	response_from_admin text	feedback_category character varying (50)
1	1	1	1	5	Great quiz, very informative and fun!	2025-04-07 00:58:27.258974	[null]	General
2	2	2	2	4	It was a good experience, but I would prefer more challenging questions.	2025-04-07 00:58:27.258974	[null]	General
3	4	4	4	5	Loved the quiz, especially the questions about memes!	2025-04-07 00:58:27.258974	[null]	General
4	7	7	7	5	I enjoyed it thoroughly, would recommend to my friends!	2025-04-07 00:58:27.258974	[null]	General
5	10	10	10	5	Excellent experience, definitely coming back for more!	2025-04-07 00:58:27.258974	[null]	General
6	13	13	13	5	Absolutely loved it! Very engaging and fun!	2025-04-07 00:58:27.258974	[null]	General

4.

Query Query History

```
1 ✓ SELECT *
2   FROM quiz
3 WHERE created_at > '2025-01-01' AND max_score > 100;
4
```

Data Output Messages Notifications

	quiz_id [PK] integer	creator_id integer	title character varying (255)	description text	category character varying (100)	difficulty_level character varying (50)	created_at timestamp without time zone	time_limit integer
1	3	3	Философия Пельменей	Глубокий квиз о смысле начинки.	Философия	medium	2025-04-07 00:18:15.468764	15
2	5	4	Кошачьи загадки	Только для тех, кто был выбран кошками.	Животные	hard	2025-04-07 00:18:15.468764	20
3	9	6	Квиз по квизам	Метаквиз для квизмастеров.	Метаквинг	hard	2025-04-07 00:18:15.468764	25
4	13	3	Числа и чувства	Математика с элементами драмы.	Математика	hard	2025-04-07 00:18:15.468764	18

5.

Query Query History

```
1 ▾ SELECT r.user_id, r.attempt_date, r.highest_streak, r.score
2   FROM result r
3   WHERE r.attempt_date BETWEEN '2025-01-01' AND '2025-12-31'
4   AND r.highest_streak >= 5
5   AND r.score > 0;
6
```

Data Output Messages Notifications

	user_id integer	attempt_date timestamp without time zone	highest_streak integer	score integer
1	2	2025-04-07 00:55:30.909853	5	90
2	4	2025-04-07 00:55:30.909853	5	85
3	5	2025-04-07 00:55:30.909853	6	95
4	7	2025-04-07 00:55:30.909853	5	92
5	10	2025-04-07 00:55:30.909853	5	88
6	12	2025-04-07 00:55:30.909853	6	90
7	14	2025-04-07 00:55:30.909853	6	91
8	15	2025-04-07 00:55:30.909853	5	79

CONCAT, UPPER, LOWER, SUBSTRING, TRIM:

1.

Query Query History

```
1 ▾ SELECT CONCAT(u.username, ' (' , u.email, ')') AS user_info
2   FROM user    u;
3
```

Data Output Messages Notifications

SQL

	user_info	lock
1	alish_akadil (alish@example.com)	
2	beka_user (beka@example.com)	
3	akbope_22 (akbope@example.com)	
4	nuris_88 (nuris@example.com)	
5	teacher01 (teach1@example.com)	
6	guest_user1 (guest1@example.com)	
7	admin2 (admin2@example.com)	
8	zhanibek_kz (zhanibek@example.co...)	
9	samira mn (samira@example.com)	

2.

Query History

Data Output Messages Notifications



	upper_username	text	lock
1	ALISH_AKADIL		
2	BEKA_USER		
3	AKBOPE_22		
4	NURIS_88		
5	TEACHER01		
6	GUEST_USER1		
7	ADMIN2		
8	ZHANIBEK_KZ		
9	SAMIRA_MN		

3.

Query Query History

```
1 ▼ SELECT LOWER(u.email) AS lower_email
2   FROM user    u;
3
```

Data Output Messages Notifications

≡+ ↻ ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ ⏻ ⏽ SQL

	lower_email	🔒
	text	
1	alish@example.com	
2	beka@example.com	
3	akbope@example.com	
4	nuris@example.com	
5	teach1@example.com	
6	guest1@example.com	
7	admin2@example.com	
8	zhanibek@example.co...	
9	samira@example.com	

4.

Query Query History

```
1 ▾  SELECT SUBSTRING(u.username, 1, 5) AS username_substring
2   FROM user    u;
3
```

Data Output Messages Notifications

≡+ ↻ ⏴ 📁 ⏴ 🗑️ ⏴ 🔍 🔒 SQL

	username_substring	text
1	alish	
2	beka_	
3	akbop	
4	nuris	
5	teach	
6	guest	
7	admin	
8	zhani	
9	samir	

5.

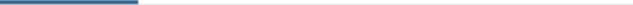
Query History

1 ✓ SELECT TRIM(u.

```
1 ▾ SELECT TRIM(u.username) AS trimmed_username  
2   FROM user    u;  
3
```

3

Data Output Messages Notifications



	trimmed_username	text
1	alish_akadil	
2	beka_user	
3	akbope_22	
4	nuris_88	
5	teacher01	
6	guest_user1	
7	admin2	
8	zhanibek_kz	
9	samira_mp	

NOW, DATEADD, DATEDIFF, YEAR, MONTH:

1.

Query Query History

```
1 ✓ SELECT r.result_id, r.user_id, r.quiz_id, r.attempt_date, NOW() AS current_timestamp
2 FROM result r;
3
```

Data Output Messages Notifications

Showing rows 1-9 of 9

	result_id [PK] integer	user_id integer	quiz_id integer	attempt_date timestamp without time zone	current_timestamp timestamp with time zone
1	1	1	1	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
2	2	2	2	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
3	3	3	3	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
4	4	4	1	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
5	5	5	2	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
6	6	6	3	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
7	7	7	1	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
8	8	8	2	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05
9	9	9	3	2025-04-07 00:55:30.909853	2025-04-07 01:29:56.829311+05

2.

```
1 ▾ SELECT r.result_id, r.attempt_date, r.attempt_date + INTERVAL '7 days' AS due_date
2   FROM result r;
3
```

Data Output Messages Notifications

The screenshot shows a database interface with a table named 'result'. The table has three columns: 'result_id' (PK), 'attempt_date', and 'due_date'. The 'attempt_date' column contains the value '2025-04-07 00:55:30.909853' for all rows. The 'due_date' column contains the value '2025-04-14 00:55:30.909853' for all rows. There are 9 rows in total.

	result_id [PK] integer	attempt_date timestamp without time zone	due_date timestamp without time zone
1	1	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
2	2	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
3	3	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
4	4	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
5	5	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
6	6	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
7	7	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
8	8	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853
9	9	2025-04-07 00:55:30.909853	2025-04-14 00:55:30.909853

3.

Query Query History

```
1 ✓ SELECT r.result_id, r.attempt_date, r.attempt_date + INTERVAL '1 month' AS next_month_attempt
2   FROM result r;
3
```

Data Output Messages Notifications

Showing rows: 1 to 1

	result_id [PK] integer	attempt_date timestamp without time zone	next_month_attempt timestamp without time zone
1	1	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
2	2	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
3	3	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
4	4	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
5	5	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
6	6	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
7	7	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
8	8	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853
9	9	2025-04-07 00:55:30.909853	2025-05-07 00:55:30.909853

✓ Successfully run

4.

Query Query History

```
1 ✓ SELECT r.result_id, r.attempt_date, r.attempt_date + INTERVAL '1 year' AS next_year_attempt
2   FROM result| r;
3
```

Data Output Messages Notifications

Showing rows: 1 to 1

	result_id [PK] integer	attempt_date timestamp without time zone	next_year_attempt timestamp without time zone
1	1	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
2	2	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
3	3	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
4	4	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
5	5	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
6	6	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
7	7	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
8	8	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853
9	9	2025-04-07 00:55:30.909853	2026-04-07 00:55:30.909853

✓ Successfully run

5.

Query Query History

```
1 | SELECT r.result_id, r.attempt_date, EXTRACT(YEAR FROM r.attempt_date) AS year
2 | FROM result r;
3 |
```

Data Output Messages Notifications

Sh

	result_id [PK] integer	attempt_date timestamp without time zone	year numeric
1	1	2025-04-07 00:55:30.909853	2025
2	2	2025-04-07 00:55:30.909853	2025
3	3	2025-04-07 00:55:30.909853	2025
4	4	2025-04-07 00:55:30.909853	2025
5	5	2025-04-07 00:55:30.909853	2025
6	6	2025-04-07 00:55:30.909853	2025
7	7	2025-04-07 00:55:30.909853	2025
8	8	2025-04-07 00:55:30.909853	2025
9	9	2025-04-07 00:55:30.909853	2025

6.

Query Query History

```
1 ▾ SELECT r.result_id, r.attempt_date, EXTRACT(DAY FROM r.attempt_date) AS day
2 FROM result r;
3
```

Data Output Messages Notifications



	result_id [PK] integer	attempt_date timestamp without time zone	day numeric
1	1	2025-04-07 00:55:30.909853	7
2	2	2025-04-07 00:55:30.909853	7
3	3	2025-04-07 00:55:30.909853	7
4	4	2025-04-07 00:55:30.909853	7
5	5	2025-04-07 00:55:30.909853	7
6	6	2025-04-07 00:55:30.909853	7
7	7	2025-04-07 00:55:30.909853	7
8	8	2025-04-07 00:55:30.909853	7
9	9	2025-04-07 00:55:30.909853	7

UPDATE:

1.

Query Query History

```
1 UPDATE result
2 SET score = 85
3 WHERE user_id = 1 AND attempt_date = '2025-03-15';
4
```

Data Output [Messages](#) Notifications

UPDATE 0

Query returned successfully in 110 msec.

2.

Query Query History

```
1 ✓ UPDATE user
2   SET status = 'active'
3 WHERE user_id = 2 AND status = 'inactive';
4
```

Data Output Messages Notifications

UPDATE 0

Query returned successfully in 70 msec.

3.

[Query](#) [Query History](#)

```
1 ✓ UPDATE feedback
2   SET comment = 'Great quiz! Loved it.'
3 WHERE user_id = 4 AND quiz_id = 2;
4 |
```

[Data Output](#) [Messages](#) [Notifications](#)

UPDATE 0

Query returned successfully in 136 msec.

4.

Query Query History

```
1 ✓ UPDATE reward
2   SET reward_name = 'Winner of March'
3 WHERE user_id = 5;
4
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 68 msec.

5.

Query Query History

```
1 ✓ UPDATE user
2   SET profile_picture = 'profile_7.jpg'
3 WHERE user_id = 7 AND profile_picture = 'default.jpg';
4
```

Data Output Messages Notifications

UPDATE 0

Query returned successfully in 72 msec.

DELETE:

1.

Query Query History

```
1 ▾ DELETE FROM result
2 WHERE attempt_date < '2025-01-01';
3
```

Data Output Messages Notifications

DELETE 0

Query returned successfully in 71 msec.

2.

Query Query History

```
1 ▾ DELETE FROM results
2 WHERE user_id = 1 AND quiz_id = 2;
3 |
```

Data Output Messages Notifications

DELETE 0

Query returned successfully in 71 msec.

3.

Query [Query History](#)

```
1 ✓ DELETE FROM feedback  
2 WHERE rating < 3;  
3
```

[Data Output](#) [Messages](#) [Notifications](#)

DELETE 2

Query returned successfully in 108 msec.

4.

Query Query History

```
1 ▼ DELETE FROM rewards
2 WHERE reward_name = 'First Place' AND user_id = 3;
3
```

Data Output Messages Notifications

DELETE 2

Query returned successfully in 108 msec.

5.

Query Query History

```
1 ✓ DELETE FROM user
2 WHERE last_login < NOW() - INTERVAL '6 months';
3
```

Data Output [Messages](#) Notifications

DELETE 0

Query returned successfully in 93 msec.

In this assignment, I explored and applied various SQL DQL and DML commands using the database structure I built in Assignment II. I performed data filtration using the `WHERE` clause to retrieve specific records based on conditions. Then, I used multiple string functions such as `CONCAT`, `UPPER`, `LOWER`, `SUBSTRING`, and `TRIM` to manipulate and display text data more effectively.

Additionally, I experimented with date functions like `NOW()`, `AGE()`, `EXTRACT`, and `DATE_TRUNC` to analyze and format time-related data across different tables. I also updated existing records using the `UPDATE` command, ensuring changes were based on meaningful conditions, and I used the `DELETE` command to remove selected rows without affecting the structure of the database.

All the queries were tested on real data within my extended quiz system database, which includes users, quizzes, results, feedback, and rewards. Through this hands-on practice, I improved my understanding of SQL syntax and logic, as well as how to query and manage relational data efficiently.

P.S. SAGACIOUS Akadil  