

General Technologies

recWtXKsmj0qxaMrs

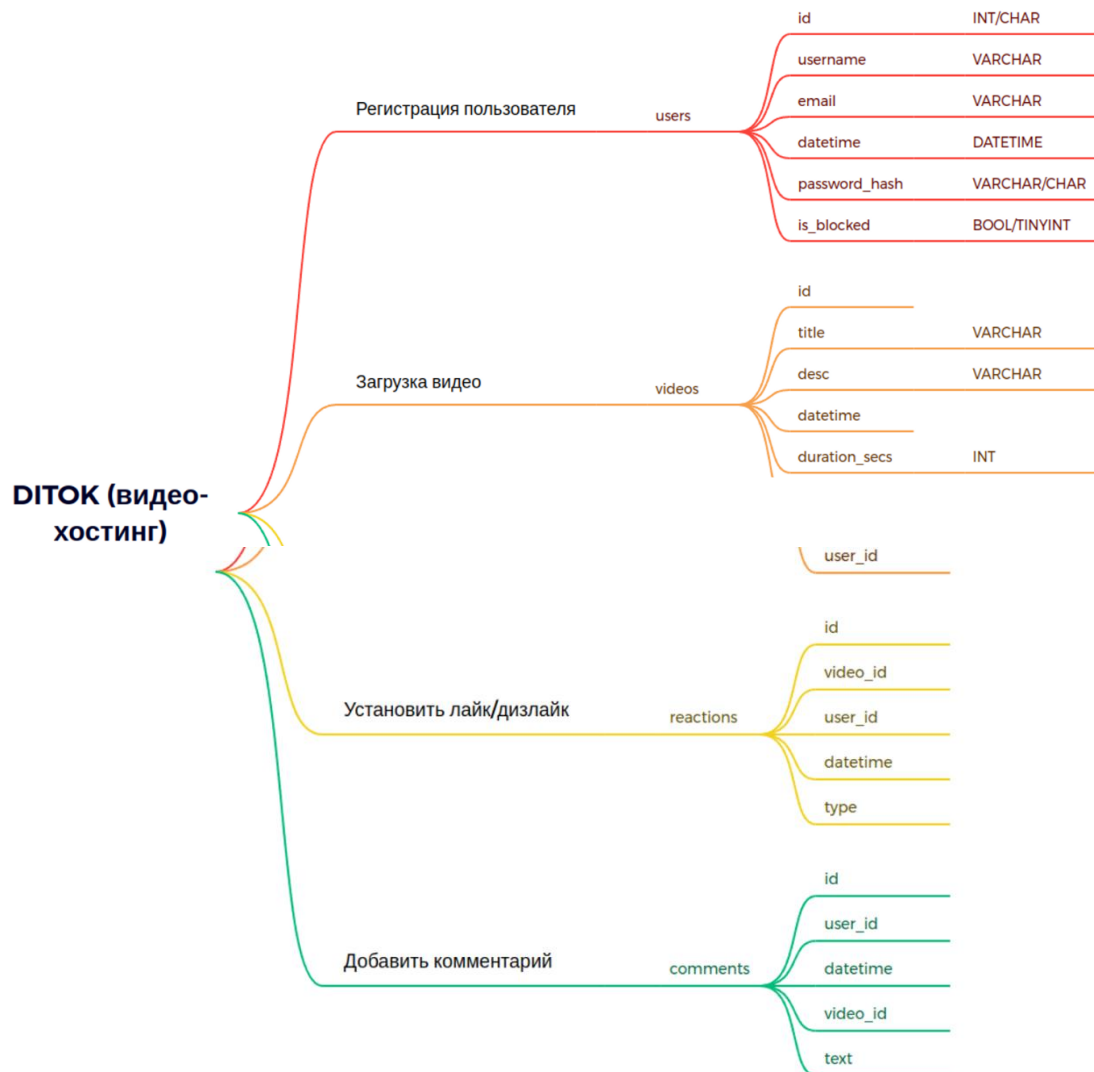
First level: С помощью SQL создать четыре таблицы согласно предложенной схеме (см. конспект).

Конспект занятия <https://western-appeal-39b.notion.site/GenTech-Feb-2-2023-a74ddaf68855443e9f4e6737a4a0cbf6>

Second level:

ITech / Feb 2, 2023

REGEX



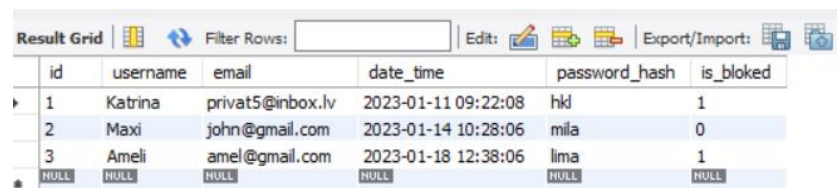
```
CREATE DATABASE DITOK;
```

```
USE DITOK;
```

```
CREATE TABLE IF NOT EXISTS Users (  
  id INTEGER primary key auto_increment,  
  username VARCHAR (128) NOT NULL,  
  email VARCHAR (128) NOT NULL,  
  created_at DATETIME default current_timestamp, //устанавливается автоматически время  
  password_hash VARCHAR (128) NOT NULL, --SHA  
  is_blocked BOOL default false //устанавливаем сразу значение по умолчанию  
);
```

```
INSERT INTO Users (username, email, date_time, password_hash, is_bloked)  
VALUES ('Katrina', 'privat5@inbox.lv', '2023-01-11 09:22:08', 'hkl', true),  
('Maxi', 'john@gmail.com', '2023-01-14 10:28:06', 'mila', false),  
('Ameli', 'amel@gmail.com', '2023-01-18 12:38:06', 'lima', true);
```

```
SELECT * FROM Users;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a query, showing columns: id, username, email, date_time, password_hash, and is_bloked. There are three rows of data, and a fourth row with all NULL values. The interface includes a 'Filter Rows' search bar, an 'Edit' button, and an 'Export/Import' button.

	id	username	email	date_time	password_hash	is_bloked
▶	1	Katrina	privat5@inbox.lv	2023-01-11 09:22:08	hkl	1
	2	Maxi	john@gmail.com	2023-01-14 10:28:06	mila	0
	3	Ameli	amel@gmail.com	2023-01-18 12:38:06	lima	1
*	NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Videos (  
  id INTEGER primary key auto_increment,  
  title VARCHAR (128) NOT NULL,  
  description VARCHAR (128) NOT NULL,  
  date_time DATETIME default current_timestamp,  
  user_id INTEGER NOT NULL,  
  FOREIGN KEY (user_id) REFERENCES Users (user_id) //привязка к ID из табл.Users  
  duration_secs INTEGER  
);
```

```
INSERT INTO Videos (title, descrip, date_time, user_id, duration_secs)
VALUES ('Panorama', 'News', '2023-01-12 07:26:08', 2, 2508),
('Beet', 'Game', '2023-02-09 15:32:58', 1, 4409),
('Kids', 'Funny', '2023-01-22 10:44:06', 3, 1004);
```

```
SELECT * FROM Videos;
```

Result Grid						
		Filter Rows:		Edit:		Export/Im
	id	title	descrip	date_time	user_id	duration_secs
▶	1	Panorama	news	2023-01-12 07:26:08	2	2508
	2	Beet	game	2023-02-09 15:32:58	1	4409
	3	Kids	Funny	2023-01-22 10:44:06	3	1004

```
CREATE TABLE Reactions (
id INTEGER primary key auto_increment,
video_id INTEGER,
FOREIGN KEY (video_id) REFERENCES Videos (video_id) //привязка к ID из табл. Videos
user_id INTEGER,
FOREIGN KEY (user_id) REFERENCES Users (user_id) //привязка к ID из табл. Users
date_time DATETIME default current_timestamp,
react_type VARCHAR (128) NOT NULL,
);
```

```
INSERT INTO Reactions (video_id, user_id, date_time, react_type)
VALUES (2, 8, '2023-01-12 07:48:07', 'like'),
(1, 5, '2023-02-12 16:12:07', 'like'),
(3, 4, '2023-01-22 10:52:24', 'like');
```

```
SELECT * FROM Reactions;
```

Result Grid					
		Filter Rows:		Edit:	
	id	video_id	user_id	date_time	react_type
▶	1	2	8	2023-01-12 07:48:07	like
	2	1	5	2023-02-12 16:12:07	like
	3	3	4	2023-01-22 10:52:24	like
	NULL	NULL	NULL	NULL	NULL

```

CREATE TABLE Comments (
id INTEGER primary key auto_increment,
user_id INTEGER,
FOREIGN KEY (user_id) REFERENCES Users (user_id) //привязка к ID из табл. Users
video_id INTEGER,
FOREIGN KEY (video_id) REFERENCES Videos (video_id) //привязка к ID из табл. Videos
date_time DATETIME default current_timestamp,
text VARCHAR (128) NOT NULL
);

```

```

INSERT INTO Comments (user_id, video_id, date_time, text_)
VALUES (2, 4, '2023-01-12 07:48:07', 'super'),
(3, 2, '2023-02-12 16:12:07', 'toll'),
(5, 1, '2023-01-22 10:52:24', 'like it');

```

```

SELECT * FROM Comments;

```

Result Grid					
Filter Rows:					
	id	user_id	video_id	date_time	text_
▶	1	2	4	2023-01-12 07:48:07	super
	2	3	2	2023-02-12 16:12:07	toll
	3	5	1	2023-01-22 10:52:24	like it
*	NULL	NULL	NULL	NULL	NULL