

Проект по Бази от данни 2019/2020

RESTAURANTS

Васил Василев, 62348

Владислав Стефанов, 62349

Живко Георгиев, 62304

Задача	Фак. №
1. Създаване на схемите и релациите	62348
2. Добавяне на съдържание	62348
3. Примерни прости заявки и заявки върху две и повече релации	62304
4. Примери с подзаявки	62304
5. Примери със съединения	62304
6. Примери с групиране и агрегация	62349
7. Примери с ограничения	62348
8. Примери с изгледи и индекси	62349
9. Примери с тригери	62349
10. Подготовка на презентацията	62348

ОПИСАНИЕ НА ПРОЕКТА

Описание на предметната област на заданието

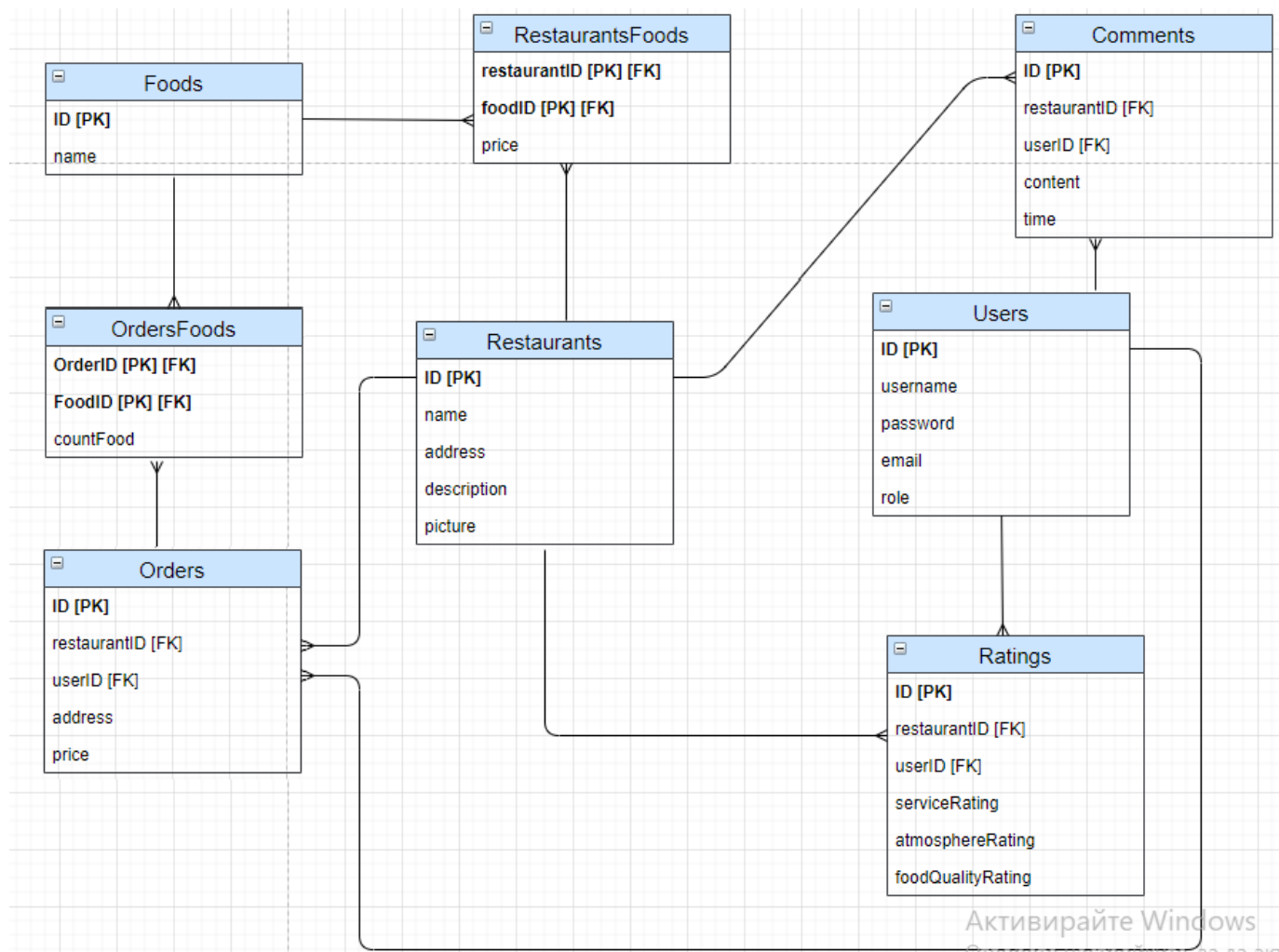
Разработената база от данни е част от по-голяма система IFDb(Internet Food Database). Основната функционалност на системата е да позволява на потребителите да се регистрират и да търсят в системата ресторанти по дадена храна. Също така те могат да правят поръчки, да оценяват ресторантите по някои основни качества (обслужване, качество на храната и атмосфера), а също така и да пишат коментари за всеки ресторант.

Всеки ресторант в системата се характеризира с име, адрес, описание и снимка, като последните две са по избор. Тъй като името не е достатъчно, за да ни осигури уникалност на всеки ресторант, добавихме ID като уникален ключ за всеки ресторант. Типовете потребители в системата са няколко - Admin, Moderator, Restaurant Page Manager, User. Всеки потребител влиза в системата чрез username, password и email. При бъдещи модификации на системата е възможно тя да поддържа повече от един потребител, използващи един и същ мейл или повече от един потребител с един и същ username, за това решихме отново да добавим ID като уникален ключ за всеки потребител.

За храните пазим името им и една междинна таблица между храни и ресторанти, за да може потребителите да търсят коя храна в кои ресторанти се предлага.

Всеки потребител може да прави поръчка за храна от даден ресторант. Ще използваме външна услуга и за това няма да

пазим много информация относно поръчките. Всяка поръчка има цена, която включва цената на храната от ресторанта плюс други услуги.



Дефиниране на схемата на релациите

на таблиците RestaurantsFoods и OrdersFoods сме задали Primary Key като ограничения (виж подобрения с добавяне на ограничения на ниво таблица)

Restaurants – съдържа информация за ресторанта(*ID, name, address, description, picture*)

Users – съдържа информация за потребителите (ID, username, password, email, role)

Ratings - показва информация за оценките (от кой потребител за кой ресторант и оценките им – serviceRating, atmosphereRating, foodQualityRating)

Orders – съдържа информация за поръчките (от кой потребител до кой ресторант, address и price)

Comments – показва информация за коментарите (от кой потребител за кой ресторант, content и дата на последна промяна)

Foods – показва различните храни, които предлагат ресторантите (ID, name)

OrdersFoods – дава информация за поръчаните храни в дадена поръчка (orderId, foodID, countFood)

RestaurantsFoods – предоставя информация за това кой ресторант какви ястия предлага (restaurantID, foodID, price)

```
CREATE TABLE Restaurants (  
    ID INT PRIMARY KEY NOT NULL,  
    name VARCHAR(256) NOT NULL,  
    address VARCHAR(256) NOT NULL,  
    description VARCHAR(256),  
    picture VARCHAR(256)  
)
```

```
CREATE TABLE Users (  
    ID INT PRIMARY KEY NOT NULL,  
    username VARCHAR(256) NOT NULL,  
    password VARCHAR(256) NOT NULL,  
    email VARCHAR(256) NOT NULL,  
    role VARCHAR(256) NOT NULL  
)
```

```
CREATE TABLE Ratings (  
    ID INT PRIMARY KEY NOT NULL,  
    restaurantID INT NOT NULL,  
    userID INT NOT NULL,  
    serviceRating INT,  
    atmosphereRating INT,  
    foodQualityRating INT  
)
```

```
CREATE TABLE Orders (  
    ID INT PRIMARY KEY NOT NULL,  
    restaurantID INT NOT NULL,  
    userID INT NOT NULL,  
    address VARCHAR(256) NOT NULL,  
    price DECIMAL(6,2) NOT NULL DEFAULT 0  
)
```

```
CREATE TABLE Comments (  
    ID INT PRIMARY KEY NOT NULL,  
    restaurantID INT NOT NULL,  
    userID INT NOT NULL,  
    content VARCHAR(256),  
    time DATE  
)
```

```
CREATE TABLE Foods (  
    ID INT PRIMARY KEY NOT NULL,  
    name VARCHAR(256) NOT NULL  
)
```

```
CREATE TABLE OrdersFoods (  
    orderID INT NOT NULL,  
    foodID INT NOT NULL,  
    countFood INT NOT NULL DEFAULT 0  
)
```

```
CREATE TABLE RestaurantsFoods (  
    restaurantID INT NOT NULL,  
    foodID INT NOT NULL,  
    price DECIMAL(6,2) NOT NULL DEFAULT 0  
)
```

Добавяне на примерно съдържание, за доказване чрез
примери на работоспособността на системата (на база на
материала от 8-то упр.)

```
use restaurants  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (1, 'vasilnv', 'alabalaa',  
'vasil@gmail.com', 'User')  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (2, 'jivkomg', 'alabalal',  
'jivko@gmail.com', 'User')  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (3, 'vladislav', 'alablaal',  
'vladi@gmail.com', 'User')  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (4, 'gogo', 'alabalaa', 'gogo@abv.bg',  
'Admin')  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (5, 'jojo', 'alabalaa',  
'jojo@gmail.com', 'User')  
INSERT INTO USERS(ID, username, password, email,  
role) VALUES (6, 'pesho', 'alabalala',  
'pesho@gmail.com', 'User')  
  
INSERT INTO Restaurants(ID, name, address,  
description, picture) VALUES (1, 'Nacional',  
'Studentski kompleks 42', 'top food', NULL)  
INSERT INTO Restaurants(ID, name, address) VALUES  
(2, 'Nacional', 'Tsarigradsko')
```

```
INSERT INTO Restaurants(ID, name, address) VALUES
(3, 'Talent's Restaurant', 'Boulevard "Evlogi i
Hristo Georgiev"')
INSERT INTO Restaurants(ID, name, address) VALUES
(4, 'Shtastlivetsa', 'bul. "Vitosha" 27')
INSERT INTO Restaurants(ID, name, address) VALUES
(5, 'Made in Home', 'ul. "Angel Kanchev" 30')
INSERT INTO Restaurants(ID, name, address) VALUES
(6, 'Cosmos Restaurant', 'Lavele St 19')
INSERT INTO Restaurants(ID, name, address) VALUES
(7, 'Chef's Restaurant', 'ul. "Lyubata"')
```

```
INSERT INTO FOODS(ID, name) VALUES (1, 'Musaka')
INSERT INTO FOODS(ID, name) VALUES (2, 'Tarator')
INSERT INTO FOODS(ID, name) VALUES (3, 'Meat
rolls')
INSERT INTO FOODS(ID, name) VALUES (4, 'Spaghetti
Bolognese')
INSERT INTO FOODS(ID, name) VALUES (5, 'Soup')
INSERT INTO FOODS(ID, name) VALUES (6, 'Bob')
INSERT INTO FOODS(ID, name) VALUES (7, 'Cesar
salad')
```

```
INSERT INTO ORDERS(ID, restaurantID, userID,
address, price) VALUES (1, 1, 1, 'Studenski kompleks
№8', 10)
INSERT INTO ORDERS(ID, restaurantID, userID,
address, price) VALUES (2, 2, 3, 'Studenski kompleks
№8', 12)
INSERT INTO ORDERS(ID, restaurantID, userID,
address, price) VALUES (3, 3, 2, 'Studenski kompleks
№50', 8)
```

```
INSERT INTO Comments(ID, restaurantID, userID,
content, time) VALUES (1, 1, 1, 'fantastic food and
staff', '2019-09-27')
```

```
INSERT INTO Comments(ID, restaurantID, userID,  
content, time) VALUES (2, 1, 2, 'fantastic food and  
staff', '2019-09-12')  
INSERT INTO Comments(ID, restaurantID, userID,  
content, time) VALUES (3, 2, 3, 'fantastic food and  
staff', '2019-09-01')  
INSERT INTO Comments(ID, restaurantID, userID,  
content, time) VALUES (4, 3, 1, 'fantastic food and  
staff', '2019-12-05')  
INSERT INTO Comments(ID, restaurantID, userID,  
content, time) VALUES (5, 4, 2, 'fantastic food and  
staff', '2019-06-15')
```

```
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (1, 1, 1, 5, 5, 5)  
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (2, 1, 1, 5, 5, 5)  
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (3, 2, 3, 3, 5, 5)  
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (4, 3, 3, 3, 2, 5)  
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (5, 4, 4, 4, 5, 5)  
INSERT INTO Ratings(ID, restaurantID, userID,  
serviceRating, atmosphereRating, foodQualityRating)  
VALUES (6, 1, 2, 1, 1, 5)
```

```
INSERT INTO OrdersFoods(orderID, foodID, countFood)  
VALUES (1, 1, 1), (2, 2, 2), (3, 1, 1)
```

```
INSERT INTO RestaurantsFoods(restaurantID, foodID,  
price) VALUES (3, 1, 10)
```



```

INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (1, 1, 12)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (2, 1, 13)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (4, 1, 20)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (1, 2, 5)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (1, 3, 6)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (2, 4, 7)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (3, 6, 2)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (5, 5, 3)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (4, 4, 4)
INSERT INTO RestaurantsFoods(restaurantID, foodID,
price) VALUES (3, 2, 5)

```

Примерни заявки с резултата им разделени (по подобие на материала на упр. 1 – 5) на:

1. Прости заявки

Напишете заявка, която изкарва информацията за тези поръчки, чиято цена е по-голяма от 10 лева.

```
SELECT * FROM Orders WHERE price > 10
```

	ID	restaurantID	userID	address	price
1	2	2	3	Studenski kompleks №8	12.00

Напишете заявка, която изкарва username и email за всеки един от потребителите

```
SELECT username as Name, email from Users
```

	Name	email
1	vasilnv	vasil@gmail.com
2	jivkomg	jivko@gmail.com
3	vladislav	vladi@gmail.com
4	gogo	gogo@abv.bg
5	jojo	jojo@gmail.com
6	pesho	pesho@gmail....

Напишете заявка, която изкарва имената на храните, чието име не започва с M

SELECT name FROM Foods WHERE name NOT LIKE 'M%'

	name
1	Tarator
2	Spaghetti Bolognese
3	Soup
4	Bob
5	Cesar salad

Напишете заявка, която изкарва всичката информация за потребителите, които имат профил с email в gmail

SELECT * FROM Users WHERE email LIKE '%gmail.com'

	ID	username	password	email	role
1	1	vasilnv	alabalaa	vasil@gmail.com	User
2	2	jivkomg	alabalal	jivko@gmail.com	User
3	3	vladislav	alablaal	vladi@gmail.com	User
4	5	jojo	alabalaa	jojo@gmail.com	User
5	6	pesho	alabalala	pesho@gmail.com	User

Напишете заявка, която изкарва всичката информация за рейтингите, където рейтинга на обслужването и атмосферата е повече от 3, подредено възходящо по рейтинг на обслужването

SELECT * FROM Ratings WHERE servicerating > 3 AND atmosphererating > 3 ORDER BY servicerating ASC

	ID	restaurantID	userID	serviceRating	atmosphereRating	foodQualityRating
1	5	4	4	4	5	5
2	1	1	1	5	5	5
3	2	1	1	5	5	5

2. Заявки върху две и повече релации

Напишете заявка, която изкарва различните имена и рейтинги на атмосферата, където рейтинга на атмосферата е 5.

```
SELECT DISTINCT name, atmosphererating FROM  
Ratings, Restaurants WHERE atmosphererating = 5 AND  
Ratings.restaurantID = Restaurants.id
```

	name	atmosphererating
1	Nacional	5
2	Shtastlivetsa	5

Напишете заявка, която изкарва цените на храните в ресторантите и сумите на поръчките

```
SELECT RestaurantsFoods.price, Orders.price FROM  
RestaurantsFoods, Orders WHERE Orders.restaurantID  
= RestaurantsFoods.restaurantID
```

	price	price
1	10.00	8.00
2	12.00	10.00
3	13.00	12.00
4	5.00	10.00
5	6.00	10.00
6	7.00	12.00
7	2.00	8.00
8	5.00	8.00

Напишете заявка, която изкарва съдържанието на всички коментари и техните ID-та, username, написани от потребител с username vasilnv

```
SELECT Comments.ID AS commentID, username, content  
FROM Users, Comments where username = 'vasilnv' AND  
Users.ID = Comments.userID
```

	commentID	username	content
1	1	vasilnv	fantastic food and staff
2	4	vasilnv	fantastic food and staff

Напишете заявка, която изкарва ID на поръчките, адреса на доставка на всяка от тях и сумите за ресторантите с име Nacional

```
SELECT Orders.ID, Orders.address, price, name FROM  
Orders, Restaurants WHERE name = 'Nacional' AND  
Orders.restaurantID = Restaurants.ID
```

	ID	address	price	name
1	1	Studenski kompleks №8	10.00	Nacional
2	2	Studenski kompleks №8	12.00	Nacional

Напишете заявка, която изкарва името на храните и техните цени, които предлагат ресторантите

```
SELECT Foods.name, price, Restaurants.name FROM
Foods, Restaurants, RestaurantsFoods WHERE
restaurantid = Restaurants.id and Foods.id =
RestaurantsFoods.foodID
```

	food	price	restaurant
1	Musaka	10.00	Talent's Restaurant
2	Musaka	12.00	Nacional
3	Musaka	13.00	Nacional
4	Musaka	20.00	Shtastlivetsa
5	Tarator	5.00	Nacional
6	Meat rolls	6.00	Nacional
7	Spaghetti Bolognese	7.00	Nacional
8	Bob	2.00	Talent's Restaurant
9	Soup	3.00	Made in Home
10	Spaghetti Bolognese	4.00	Shtastlivetsa
11	Tarator	5.00	Talent's Restaurant

3. Подзаявки

Напишете заявка, която изкарва информацията за коментарите на потребител с username jivkomg

```
SELECT * FROM Comments WHERE userid = (SELECT id
FROM Users WHERE username='jivkomg')
```

	ID	restaurantID	userID	content	time
1	2	1	2	fantastic food and staff	2019-09-12
2	5	4	2	fantastic food and staff	2019-06-15

Напишете заявка, която изкарва ID на храната, ресторанта и цената, на която храната се предлага в ресторантите, чиито имена започват с N

```
SELECT foodID, restaurantID, price FROM
RestaurantsFoods WHERE restaurantid IN (SELECT id
FROM Restaurants WHERE name LIKE 'N%')
```

	foodID	restaurantID	price
1	1	1	12.00
2	1	2	13.00
3	2	1	5.00
4	3	1	6.00
5	4	2	7.00

Напишете заявка, която изкарва имената на храните, чиято цена в ресторантите е по-малка от 5.

```
SELECT name FROM Foods WHERE EXISTS (SELECT * FROM RestaurantsFoods WHERE price < 5 AND Foods.id = foodID)
```

	name
1	Spaghetti Bolognese
2	Soup
3	Bob

Напишете заявка, която изкарва най-евтината храна във всички ресторанти и нейното ID

```
SELECT foodID, price FROM RestaurantsFoods WHERE price <= ALL (SELECT price FROM RestaurantsFoods)
```

	foodID	price
1	6	2.00

Напишете заявка, която изкарва съдържанието на коментарите и ID на потребителя, който го е написал за ресторантите с ID = 1

```
SELECT DISTINCT Users.username, c.content FROM Users, ( SELECT Comments.userID, Comments.content FROM Comments WHERE restaurantID = 1 ) c WHERE c.userID = Users.ID
```

	username	content
1	jivkomg	fantastic food and staff
2	vasilnv	fantastic food and staff

4. Съединения

Напишете заявка, която изкарва информацията за потребителите и коментарите, които са написали

```
SELECT * FROM Comments c JOIN Users u ON c.userid = u.id
```

	ID	restaurantID	userID	content	time	ID	username	password	email	role
1	1	1	1	fantastic food and staff	2019-09-27	1	vasilnv	alabalaa	vasil@gmail.com	User
2	2	1	2	fantastic food and staff	2019-09-12	2	jivkomg	alabalal	jivko@gmail.com	User
3	3	2	3	fantastic food and staff	2019-09-01	3	vladislav	alablaal	vladi@gmail.com	User
4	4	3	1	fantastic food and staff	2019-12-05	1	vasilnv	alabalaa	vasil@gmail.com	User
5	5	4	2	fantastic food and staff	2019-06-15	2	jivkomg	alabalal	jivko@gmail.com	User

Напишете заявка, която изкарва всичката информация за всички потребители както и всичката информация за коментарите на тези, които са писали

```
SELECT * FROM Users LEFT JOIN Comments ON Users.ID = Comments.userID
```

	ID	username	password	email	role	ID	restaurantID	userID	content	time
1	1	vasilnv	alabalaa	vasil@gmail.com	User	1	1	1	fantastic food and staff	2019-09-27
2	1	vasilnv	alabalaa	vasil@gmail.com	User	4	3	1	fantastic food and staff	2019-12-05
3	2	jivkomg	alabalal	jivko@gmail.com	User	2	1	2	fantastic food and staff	2019-09-12
4	2	jivkomg	alabalal	jivko@gmail.com	User	5	4	2	fantastic food and staff	2019-06-15
5	3	vladislav	alablaal	vladi@gmail.com	User	3	2	3	fantastic food and staff	2019-09-01
6	4	gogo	alabalaa	gogo@abv.bg	Ad...	N...	NULL	NULL	NULL	NULL
7	5	jojo	alabalaa	jojo@gmail.com	User	N...	NULL	NULL	NULL	NULL
8	6	peshe	alabalala	peshe@gmail....	User	N...	NULL	NULL	NULL	NULL

Напишете заявка, която изкарва всичката информация за всички ресторанти, както и всичката информация за коментарите на тези, които имат коментари

```
SELECT * FROM Comments RIGHT JOIN Restaurants ON Restaurants.ID = Comments.restaurantID
```

	ID	restaurantID	userID	content	time	ID	name	address	description	picture
1	1	1	1	fantastic food and staff	2019-09-27	1	Nacional	Studentski kompleks 42	top food	NULL
2	2	1	2	fantastic food and staff	2019-09-12	1	Nacional	Studentski kompleks 42	top food	NULL
3	3	2	3	fantastic food and staff	2019-09-01	2	Nacional	Tsarigradsko	NULL	NULL
4	4	3	1	fantastic food and staff	2019-12-05	3	Talent'...	Boulevard "Evlogi i ...	NULL	NULL
5	5	4	2	fantastic food and staff	2019-06-15	4	Shtastli...	bul. "Vitosha" 27	NULL	NULL
6	N...	NULL	NULL	NULL	NULL	5	Made i...	ul. "Angel Kanchev"...	NULL	NULL
7	N...	NULL	NULL	NULL	NULL	6	Cosmo...	Lavele St 19	NULL	NULL
8	N...	NULL	NULL	NULL	NULL	7	Chef's ...	ul. "Lyubata"	NULL	NULL

Напишете заявка, която изкарва информация за името и адреса на ресторантите и username на потребителите, оценили ги със средна оценка по-висока от 4

```
SELECT username, Restaurants.name,
Restaurants.address FROM Ratings JOIN USERS ON
Users.ID = Ratings.userID JOIN Restaurants ON
Ratings.restaurantID = Restaurants.ID WHERE
(serviceRating + atmosphereRating +
foodQualityRating) / 3 >= 4
```

	username	name	address
1	vasilnv	Nacional	Studentski komplex 42
2	vasilnv	Nacional	Studentski komplex 42
3	vladislav	Nacional	Tsarigradsko
4	gogo	Shtastli...	bul. "Vitosha" 27

Напишете заявка, която изкарва всичката информация за потребителите, които са гласували за даден ресторант, както и всичката информация за ресторантите и рейтинга им (включително и тези без рейтинг)

```
SELECT * FROM Users JOIN Ratings ON Users.ID =
Ratings.userID RIGHT JOIN Restaurants ON
restaurantID = Restaurants.ID
```

	ID	username	password	email	role	ID	restaurantID	userID	serviceRating	atmosphere...	foodQualityRating	ID	name	address	description	picture
1	1	vasilnv	alabalaa	vasil@gmail.com	User	1	1	1	5	5	5	1	Nacional	Studentski komplex 42	top food	NULL
2	1	vasilnv	alabalaa	vasil@gmail.com	User	2	1	1	5	5	5	1	Nacional	Studentski komplex 42	top food	NULL
3	2	jivkomg	alabalal	jivko@gmail.com	User	6	1	2	1	1	5	1	Nacional	Studentski komplex 42	top food	NULL
4	3	vladislav	alablaal	vladi@gmail.com	User	3	2	3	3	5	5	2	Nacional	Tsarigradsko	NULL	NULL
5	3	vladislav	alablaal	vladi@gmail.com	User	4	3	3	3	2	5	3	Talent'...	Boulevard "Evlogi i ...	NULL	NULL
6	4	gogo	alabalaa	gogo@abv.bg	Admin	5	4	4	4	5	5	4	Shtastli...	bul. "Vitosha" 27	NULL	NULL
7	N...	NULL	NULL	NULL	NULL	N...	NULL	NULL	NULL	NULL	NULL	5	Made i...	ul. "Angel Kanchev"...	NULL	NULL
8	N...	NULL	NULL	NULL	NULL	N...	NULL	NULL	NULL	NULL	NULL	6	Cosmo...	Lavele St.19	NULL	NULL
9	N...	NULL	NULL	NULL	NULL	N...	NULL	NULL	NULL	NULL	NULL	7	Chef's...	ul. "Lyubata"	NULL	NULL

5. Групиране и агрегация

Напишете заявка, която изкарва информация за името на дадена храна и броят на ресторантите, които я предлагат, като записите са подредени низходящо по броя на ресторантите.

```
SELECT COUNT(rf.restaurantID) AS restaurantsCount,  
f.name AS foodName  
FROM RestaurantsFoods AS rf  
JOIN Foods AS f ON f.ID = rf.foodID  
GROUP BY f.name  
ORDER BY restaurantsCount DESC
```

	restaurantsCount	foodName
1	4	Musaka
2	2	Spaghetti Bolognese
3	2	Tarator
4	1	Soup
5	1	Bob
6	1	Meat rolls

Напишете заявка, която изкарва информация за името на дадена храна и броят на ресторантите, които я предлагат, като записите са подредени по името на храната.

```
SELECT COUNT(rf.restaurantID) AS restaurantsCount,  
f.name AS foodName  
FROM RestaurantsFoods AS rf  
JOIN Foods AS f ON f.ID = rf.foodID  
GROUP BY f.name  
ORDER BY f.name
```

	restaurantsCount	foodName
1	1	Bob
2	1	Meat rolls
3	4	Musaka
4	1	Soup
5	2	Spaghetti Bolognese
6	2	Tarator

Напишете заявка, която изкарва броя на потребителите в системата

```
SELECT COUNT(ID) AS usersCount  
FROM Users
```

	usersCount
1	6

Напишете заявка, която изкарва броят на ресторантите в системата

```
SELECT COUNT(ID) AS restaurantsCount  
FROM Restaurants
```

	restaurantsCount
1	7

Напишете заявка, която изкарва броя на поръчките за всеки ресторант(име и адрес), като резултата е подреден по име на ресторанта и адреса

```
SELECT COUNT(o.ID) AS ordersCount, r.name AS  
restaurantName, r.address AS restaurantAddress  
FROM Orders AS o  
JOIN Restaurants AS r ON r.ID = o.restaurantID  
GROUP BY r.name, r.address  
ORDER BY restaurantName, restaurantAddress
```

	ordersCount	restaurantName	restaurantAddress
1	1	Nacional	Studentski kompleks 42
2	1	Nacional	Tsarigradsko
3	1	Talent's Rest...	Boulevard "Evlogi i ...

Напишете заявка, която изкарва броя на храните, които един ресторант предлага (име и адрес), подредени по име и адрес на ресторанта

```
SELECT COUNT(rf.foodID) AS foodsCount, r.name AS  
restaurantName, r.address AS restaurantAddress  
FROM RestaurantsFoods AS rf  
JOIN Restaurants AS r ON rf.restaurantID = r.ID  
GROUP BY r.name, r.address  
ORDER BY restaurantName, restaurantAddress
```

	foodsCount	restaurantName	restaurantAddress
1	1	Made in Home	ul. "Angel Kanchev" 30
2	3	Nacional	Studentski kompleks 42
3	2	Nacional	Tsarigradsko
4	2	Shtastlivetsa	bul. "Vitosha" 27
5	3	Talent's Rest...	Boulevard "Evlogi i Hr...

Напишете заявка, която изкарва име и адрес на ресторантите, както и осреднения рейтинг от всички рейтинги, които е получил, подредени по осреднения рейтинг низходящо

```
SELECT AVG((serviceRating + atmosphereRating +
foodQualityRating) / 3) AS averageOverallRating,
r.name AS restaurantName, r.address AS
restaurantAddress
FROM Ratings AS rat
JOIN Restaurants AS r ON restaurantID = r.ID
GROUP BY r.name, r.address
ORDER BY averageOverallRating DESC
```

	averageOverallRating	restaurantName	restaurantAddress
1	4	Shtastlivetsa	bul. "Vitosha" 27
2	4	Nacional	Studentski kompleks 42
3	4	Nacional	Tsarigradsko
4	3	Talent's Rest...	Boulevard "Evlogi i ...

Напишете заявка, която изкарва максималния среден цялостен рейтинг, който ресторант е успял да получи

```
SELECT MAX(averageRating) AS
maxAverageOverallRating
FROM (SELECT AVG((serviceRating + atmosphereRating
+ foodQualityRating) / 3) AS averageRating,
restaurantID FROM Ratings GROUP BY restaurantID) AS
avrg
INNER JOIN Restaurants AS r ON avrg.restaurantID =
r.ID
```

	maxAverageOverallRating
1	4

Напишете заявка, която изкарва средния рейтинг за обслужване на всички ресторанти, които са получили рейтинг, подредени низходящо по среден рейтинг за обслужването

```
SELECT AVG(serviceRating) AS averageServiceRating,
r.name AS restaurantName, r.address AS
restaurantAddress
FROM Ratings AS rat
JOIN Restaurants AS r ON restaurantID = r.ID
GROUP BY r.name, r.address
ORDER BY averageServiceRating DESC
```

	averageServiceRating	restaurantName	restaurantAddress
1	4	Shtastlivetsa	bul. "Vitosha" 27
2	3	Nacional	Studentski kompleks 42
3	3	Nacional	Tsarigradsko
4	3	Talent's Rest...	Boulevard "Evlogi i Hristo Georgiev"

Напишете заявка, която изкарва средния рейтинг за атмосфера на всички ресторанти, които са получили рейтинг, подредени низходящо по среден рейтинг за атмосферата

```
SELECT AVG(atmosphereRating) AS
averageAtmosphereRating, r.name AS restaurantName,
r.address AS restaurantAddress
FROM Ratings AS rat
JOIN Restaurants AS r ON restaurantID = r.ID
GROUP BY r.name, r.address
ORDER BY averageAtmosphereRating DESC
```

	averageAtmosphereRating	restaurantName	restaurantAddress
1	5	Shtastlivetsa	bul. "Vitosha" 27
2	5	Nacional	Tsarigradsko
3	3	Nacional	Studentski kom...
4	2	Talent's Rest...	Boulevard "Evlo...

Напишете заявка, която изкарва средния рейтинг за качеството на храната на всички ресторанти, които са получили рейтинг, подредени низходящо по среден рейтинг за качеството на храната

```
SELECT AVG(foodQualityRating) AS  
averageFoodQualityRating, r.name AS restaurantName,  
r.address AS restaurantAddress  
FROM Ratings AS rat  
JOIN Restaurants AS r ON restaurantID = r.ID  
GROUP BY r.name, r.address  
ORDER BY averageFoodQualityRating DESC
```

	averageFoodQualityRating	restaurantName	restaurantAddress
1	5	Talent's Restaurant	Boulevard "Evlogi i Hristo Georgiev"
2	5	Shtastlivetsa	bul. "Vitosha" 27
3	5	Nacional	Studentski kompleks 42
4	5	Nacional	Tsarigradsko

Заклучения, които се описват с особен фокус върху:

1) подобрения с добавяне на ограничения на ниво таблица (на база на материала от 10-то упр.)

примери за добавяне на primary keys в две от таблиците. Решихме това да са първични ключове, защото:

- в една поръчка ще се поръчва един вид храна, като ако се поръча повече от една порция няма да се прави нов запис, а ще се увеличава полето countFood
- няма логика един ресторант да предлага храна с едно и също име повече от веднъж в менюто си

```
ALTER TABLE OrdersFoods  
ADD CONSTRAINT pk_orders_foods PRIMARY KEY(orderID,  
foodID);
```

```
ALTER TABLE RestaurantsFoods  
ADD CONSTRAINT pk_restaurants_foods PRIMARY  
KEY(restaurantID, foodID)
```

Както вече споменахме в началото, един User може да влезе в системата като Admin, Moderator, Restaurant Page Manager или User

```
ALTER TABLE Users  
ADD CONSTRAINT role  
CHECK (role IN ('Admin', 'Moderator', 'Restaurant  
Page Manager', 'User'))
```

Ограничения за паролата и email-а при влизане и регистрация на един потребител. Паролата трябва да бъде поне осем символа, докато е-mail-а трябва да съдържа @

```
ALTER TABLE Users  
ADD CONSTRAINT password  
CHECK (LEN(password) >= 8)
```

```
ALTER TABLE Users  
ADD CONSTRAINT email  
CHECK (email LIKE '%@%')
```

Добавяне на ограничения за референтна цялостност (виж схемата на базата от данни)

```
ALTER TABLE Ratings  
ADD CONSTRAINT fk_to_users FOREIGN KEY(userID)  
REFERENCES Users(ID)  
ON DELETE CASCADE
```

```
ALTER TABLE Comments  
ADD CONSTRAINT fk_from_comments_to_users FOREIGN  
KEY(userID)  
REFERENCES Users(ID)  
ON DELETE CASCADE
```

```
ALTER TABLE Comments
ADD CONSTRAINT fk_from_comments_to_restaurants
FOREIGN KEY(restaurantID)
REFERENCES Restaurants(ID)
ON DELETE CASCADE
```

```
ALTER TABLE Ratings
ADD CONSTRAINT fk_from_ratings_to_restaurants
FOREIGN KEY(restaurantID)
REFERENCES Restaurants(ID)
ON DELETE CASCADE
```

```
ALTER TABLE Orders
ADD CONSTRAINT fk_from_orders_to_users FOREIGN
KEY(userID)
REFERENCES Users(ID)
ON DELETE CASCADE
```

```
ALTER TABLE Orders
ADD CONSTRAINT fk_from_orders_to_restaurants
FOREIGN KEY(restaurantID)
REFERENCES Restaurants(ID)
ON DELETE CASCADE
```

```
ALTER TABLE OrdersFoods
ADD CONSTRAINT fk_from_ordersFoods_to_orders
FOREIGN KEY(orderID)
REFERENCES Orders(ID)
ON DELETE CASCADE
```

```
ALTER TABLE OrdersFoods
ADD CONSTRAINT fk_from_ordersFoods_to_foods FOREIGN
KEY(foodID)
REFERENCES Foods(ID)
ON DELETE CASCADE
```

```
ALTER TABLE RestaurantsFoods
ADD CONSTRAINT fk_from_restaurantsFoods_to_foods
FOREIGN KEY(foodID)
REFERENCES Foods(ID)
ON DELETE CASCADE
```

```
ALTER TABLE RestaurantsFoods
ADD CONSTRAINT
fk_from_restaurantsFoods_to_restaurants FOREIGN
KEY(restaurantID)
REFERENCES Restaurants(ID)
ON DELETE CASCADE
```

ограничения от тип UNIQUE

Всеки ресторант се характеризира с неговото име и адрес – те трябва да бъдат уникални.

```
ALTER TABLE Restaurants
ADD CONSTRAINT uq_name_address UNIQUE(name,
address)
```

На този етап от развиването на системата смятаме, че всеки user влиза в нея с уникални username и email. Това може да се промени в бъдеще

```
ALTER TABLE Users
ADD CONSTRAINT uq_username UNIQUE(username);
```

```
ALTER TABLE Users
ADD CONSTRAINT uq_email UNIQUE(email);
```

ограничения, използващи CHECK

За всеки запис в таблиците RestaurantsFoods и Orders сме добавили проверка дали цената е по-голяма или равна на 0. По този начин минимизираме възможността за грешки при въвеждането на данни. Същото важи и за таблицата OrdersFoods и countFood

```
ALTER TABLE RestaurantsFoods
ADD CONSTRAINT checkPriceRestaurantsFoods
CHECK (price >= 0)
```

```
ALTER TABLE Orders
ADD CONSTRAINT checkPriceOrders
CHECK (price >= 0)
```

```
ALTER TABLE OrdersFoods
ADD CONSTRAINT checkCountFood
CHECK (countFood >= 0)
```

доразвиване на системата с добавяне на изгледи –
индекси и тригери (на база на материала от от 11-то и 12-
то упр. съответно).

индекси

*Създадохме индекси, за да можем да търсим бързо в
базата от данни.*

```
CREATE INDEX idx_username
ON Users(username)
```

```
CREATE INDEX idx_restaurant__by_name
ON Restaurants(name, address)
```

```
CREATE INDEX idx_ratings_by_restaurants
ON Ratings(restaurantID, userID)
```

```
CREATE INDEX idx_ratings_by_users
ON Ratings(userID, restaurantID)
```

```
CREATE INDEX idx_comments_by_restaurants
ON Comments(restaurantID, userID)
```



```
CREATE INDEX idx_comments_by_users  
ON Comments(userID, restaurantID)
```

```
CREATE INDEX idx_restaurants_foods  
ON RestaurantsFoods(restaurantID)
```

```
CREATE INDEX idx_orders_by_restaurants  
ON Orders(restaurantID, userID)
```

```
CREATE INDEX idx_orders_by_users  
ON Orders(userID, restaurantID)
```

```
CREATE INDEX idx_orders_foods_by_orders  
ON OrdersFoods(orderID)
```

views

създадохме изглед за това кой потребител от къде си е поръчал храна, на какъв адрес иска да бъде доставена и каква е цената на поръчката

```
CREATE VIEW dbo.orders_view AS  
SELECT r.name AS restaurantName, r.address AS  
restaurantAddress, u.username AS clientName,  
o.address AS clientAddress, o.price AS price  
FROM Orders o  
INNER JOIN Restaurants AS r ON r.ID =  
o.restaurantID  
INNER JOIN Users AS u ON u.ID = o.userID
```

изглед на коментарите за всеки ресторант, от който и кога са написани

```
CREATE VIEW dbo.comments_view AS
SELECT r.name AS restaurantName, r.address AS
restaurantAddress, u.username AS clientName,
c.content AS content, c.time AS lastModified
FROM Comments c
INNER JOIN Restaurants AS r ON r.ID =
c.restaurantID
INNER JOIN Users AS u ON u.ID = c.userID
```

изглед на рейтинзите за всеки ресторант и от който са дадени

```
CREATE VIEW dbo.ratings_view AS
SELECT r.name AS restaurantName, r.address AS
restaurantAddress, u.username AS clientName,
serviceRating, atmosphereRating, foodQualityRating
FROM Ratings AS rat
INNER JOIN Restaurants AS r ON r.ID =
rat.restaurantID
INNER JOIN Users AS u ON u.ID = rat.userID
```

изглед на храната, която всеки ресторант предлага

```
CREATE VIEW dbo.restaurants_foods_view AS
SELECT r.name AS restaurantName, r.address AS
restaurantAddress, f.name AS foodName
FROM RestaurantsFoods AS rf
INNER JOIN Restaurants AS r ON r.ID =
rf.restaurantID
INNER JOIN Foods AS f ON f.ID = rf.foodID
```

Заявки към изгледите

Напишете заявка, която изкарва името и адреса на всички ресторанти, които предлагат Musaka като храна

```
SELECT restaurants_foods_view.restaurantName,  
restaurants_foods_view.restaurantAddress  
FROM restaurants_foods_view  
WHERE foodName LIKE 'Musaka'
```

	restaurantName	restaurantAddress
1	Talent's Restaurant	Boulevard "Evlogi i Hristo Georgiev"
2	Nacional	Studentski kompleks 42
3	Nacional	Tsarigradsko
4	Shtastlivetsa	bul. "Vitosha" 27

Напишете заявка, която изкарва имената на потребителите, както и рейтингите, които са дали за ресторант с име Nacional и адрес Tsarigradsko

```
SELECT ratings_view.clientName,  
ratings_view.foodQualityRating,  
ratings_view.serviceRating,  
ratings_view.atmosphereRating  
FROM ratings_view  
WHERE restaurantName LIKE 'Nacional' AND  
restaurantAddress LIKE 'Tsarigradsko'
```

	clientName	foodQualityRating	serviceRating	atmosphereRating
1	vladislav	5	3	5

Напишете заявка, която изкарва имената и адресите на ресторантите, съдържанието на коментарите и датата на последна промяна на коментарите от потребител с username vladislav

```
SELECT comments_view.restaurantName,  
comments_view.restaurantAddress,  
comments_view.content, comments_view.lastModified  
FROM comments_view  
WHERE comments_view.clientName LIKE 'vladislav'
```

	restaurantName	restaurantAddress	content	lastModified
1	Nacional	Tsarigradsko	fantastic food and staff	2019-09-01

Напишете заявка, която изкарва имената на потребителите, адресите за доставка и цените на поръчките, които са направени към ресторант с име Nacional и адрес Studentski kompleks 42

```
SELECT orders_view.clientName,  
orders_view.clientAddress, orders_view.price  
FROM orders_view  
WHERE orders_view.restaurantName LIKE 'Nacional'  
AND orders_view.restaurantAddress LIKE 'Studentski  
komplex 42'
```

	clientName	clientAddress	price
1	vasilnv	Studentski kompleks №8	10.00

triggers

тригер за автоматично задаване на датата при добавяне или промяна на коментар

```

CREATE TRIGGER comments_add_time
ON Comments
AFTER INSERT, UPDATE
AS
    UPDATE Comments
    SET time = GETDATE()
    WHERE ID IN (SELECT ID FROM inserted)

```

тригер за изчисляване на общата цена на дадена поръчка

```

CREATE TRIGGER orders_calculate_price
ON OrdersFoods
AFTER INSERT, DELETE, UPDATE
AS
    UPDATE o
    SET o.price = o.price + (
        SELECT COALESCE(SUM(rf.price *
i.countFood), 0)
        FROM RestaurantsFoods AS rf
        JOIN inserted AS i ON rf.foodID = i.foodID
        WHERE o.restaurantID = rf.restaurantID AND
o.ID = i.orderID
    ) - (
        SELECT COALESCE(SUM(rf.price *
d.countFood), 0)
        FROM RestaurantsFoods AS rf
        JOIN deleted AS d ON rf.foodID = d.foodID
        WHERE o.restaurantID = rf.restaurantID AND
o.ID = d.orderID
    )
    FROM Orders AS o
    WHERE o.ID IN (SELECT orderID FROM inserted
UNION SELECT orderID FROM deleted)

```