

Онлайн магазин за техника

Проект по База данни 2020

Заглавна страница

Екип:

Име	Фак. №
Божидар Навчев	62261
Йордан Йорданов	62265
Борислав Христов	62317
Петър Павлов	62319

Разпределение на задачите:

Задача	Фак. №
1. Създаване на схемите и релациите	62265 62317 62319 62261
2. Добавяне на съдържание	62261 62265
3. Примерни прости заявки върху две и повече релации	62265
4. Примери с подзаявки	62317
5. Примери със съединения	62317
6. Примери с групиране и агрегация	62317 62319
7. Примери с ограничения	62261
8. Примери с изгледи и индекси	62319
9. Примери с тригери	62319
10. Подготовка на презентацията	62265 62317 62319 62261

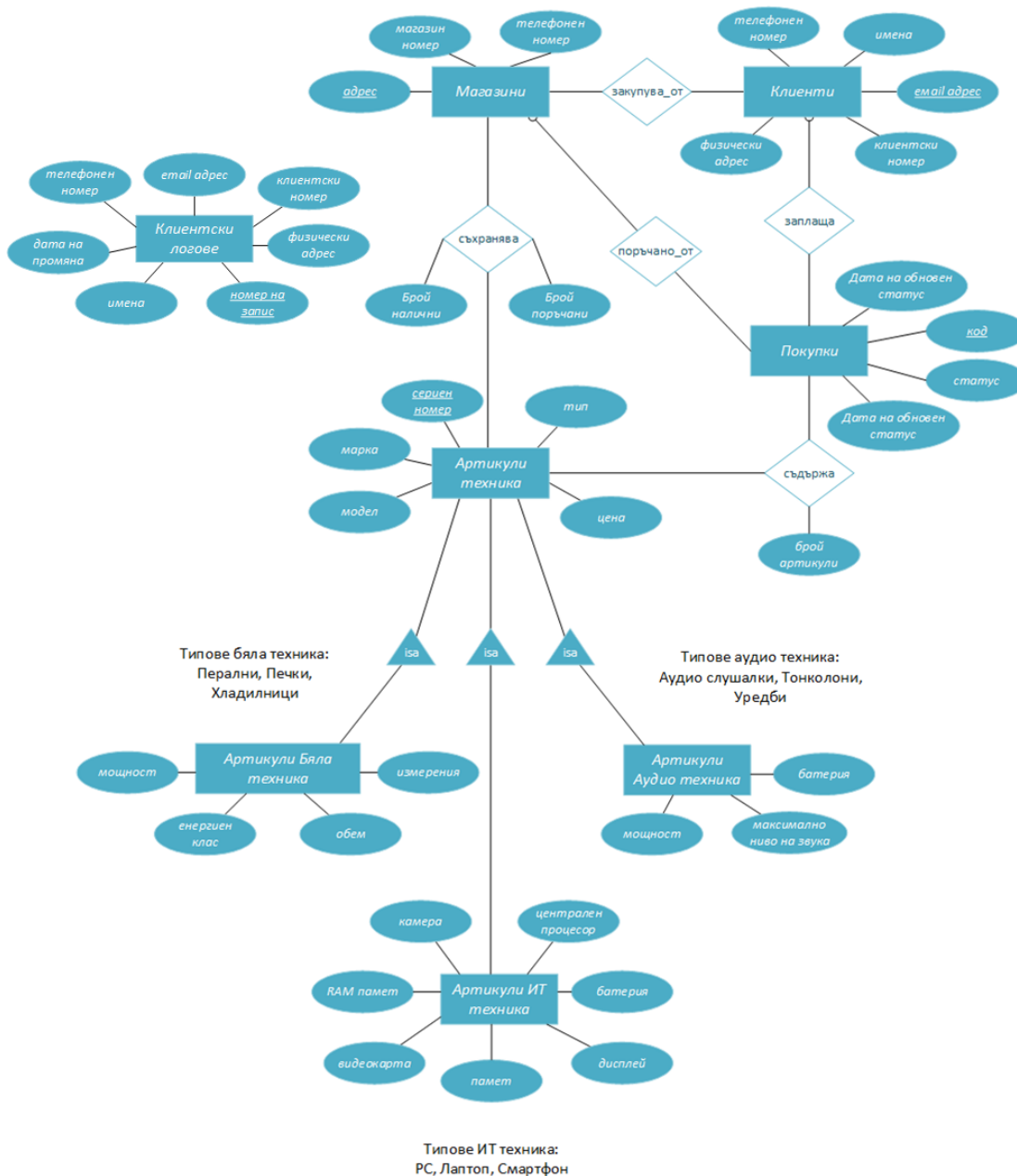
Съдържание

1. Описание:

Проектът е насочен към разработването на база данни, която е свързана с покупко-продажба и менажиране на огромен асортимент от бяла, черна, офис и ИТ техника.

Тя ще дава възможност на служителите лесно да сортират артикулите и да проверяват наличността им. Самите данни ще помогнат за създаването на онлайн каталог, който ще бъде достъпен за потребителите.

2. Дефиниране на схемата на релациите



В тази точка са извлечени примери за дефинирани релации в базата данни:

a. **“tech”** (ser_num, brand, model, type_tech, price), където:

ser_num е низ от точно 8 символа;
brand е низ от най-много 20 символа;
model е низ от най-много 30 символа;
type_tech е низ от най-много 20 символа;
price е от тип float и е неотрицателно;

```
CREATE TABLE tech (  
  ser_num CHAR(8) NOT NULL,  
  brand VARCHAR(20) NOT NULL,  
  model VARCHAR(30) NOT NULL,  
  type_tech VARCHAR(20) NOT NULL,  
  price float NOT NULL  
);
```

b. **“audio_tech”** (ser_num, sound_hz, pow_watts, battery) като наследено множество на **“tech”**:

ser_num е низ от точно 8 символа;
sound_hz е низ от най-много 20 символа;
pow_watts е низ от най-много 10 символа;
battery е низ от най-много 20 символа;

```
CREATE TABLE audio_tech (  
  ser_num CHAR(8) NOT NULL,  
  sound_hz VARCHAR(20) NOT NULL,  
  pow_watts VARCHAR(10) NOT NULL,  
  battery VARCHAR(20)  
);
```

c. **“purchased_from”** (client_email, shop_add), където:

client_email е низ от най-много 50 символа;
shop_add е низ от най-много 50 символа;

```
CREATE TABLE purchased_from (  
client_email VARCHAR(50) NOT NULL,  
shop_add VARCHAR(50) NOT NULL  
);
```

3. Добавяне на примерно съдържание

В тази точка са извлечени примерни заявки за въвеждане на ново съдържание (кортежи) в базата данни:

```
/* TABLE: tech */
```

```
INSERT INTO tech (ser_num, brand, model, type_tech, price) VALUES  
( '10000042', 'JAMO', 'J-10 Sub Black', 'Subwoofer', 359),  
( '10000043', 'JAMO', 'S 807', 'Speakers', 599),  
( '10000044', 'JAMO', 'S-626 Black Ash', 'Three-Strip Column', 599),  
( '10000045', 'CANYON', 'CNE-CBTSP6', 'Bluetooth speaker', 34.9),  
( '10000046', 'TRACER', 'GUNMAN BLUE', 'Headset', 49),  
( '10000047', 'HYPER X', 'CL STINGER CORE HX-HSCSC-BK', 'Headset', 69),  
( '10000048', 'ASTRO', 'A50 WS+Base PS4 BK/BL 939-0015', 'Headset', 599.99);  
GO
```

```
/* TABLE: audio_tech */
```

```
INSERT INTO audio_tech (ser_num, sound_hz, pow_watts, battery) VALUES  
( '10000042', '40-120Hz', '150 W', 'cable'),  
( '10000043', '42Hz-26Khz', '200 W', 'cable'),  
( '10000044', '42-20000Hz', '210 W', 'cable'),  
( '10000045', '40-20000Hz', '10 W', '4h'),  
( '10000046', '30-16000Hz', '0 W', 'cable'),  
( '10000047', '20-20000Hz', '0 W', 'cable'),  
( '10000048', '20-20000Hz', '0 W', 'cable');  
GO
```

```
/* TABLE: purchased_from */
```

```
INSERT INTO purchased_from (client_email, shop_add) VALUES  
( 'alex69@gmail.com', 'Burgas, Slaveikov 94'),  
( 'eb\_ach@gmail.com', 'Plovdiv, Georgi Stranski 3'),
```

```
('gergana_1999@abv.bg', 'Burgas, Slaveikov 94'),
('jesus_with_u@gmail.com', 'Varna, Republika 49'),
('patriot_bg@abv.bg', 'Varna, Slivnitsa 120'),
('polikai@abv.bg', 'Ruse, Hristo Botev 38'),
('puss_slayer@gmail.com', 'Sofia, Alabin 35'),
('samodiva_69@abv.bg', 'Varna, Slivnitsa 120'),
('tarator@gmail.com', 'Sofia, Pancho Vladigerov 75'),
('xX_kote_Xx@abv.bg', 'Sofia, Pancho Vladigerov 75');
GO
```

4. Примерни заявки и техните резултати

а. Прости заявки

- Изкарайте всички заплащания със статус “заплатено”, принтирайте всички атрибути:

```
SELECT *
FROM payment
WHERE pay_status = 'payed'
```

	code	client_email	pay_status
1	1006	puss_slayer@gmail.com	paid
2	1007	alex69@gmail.com	paid
3	1008	samodiva_69@abv.bg	paid
4	1010	xX_kote_Xx@abv.bg	paid
5	1011	patriot_bg@abv.bg	paid
6	1013	polikai@abv.bg	paid
7	1014	gergana_1999@abv.bg	paid

- Изкарайте всички клиенти с регистриран имейл в gmail.com и телефонен номер, започващ с цифрите “088”, принтирайте всички атрибути:

```
SELECT *
FROM client
WHERE email like '%gmail.com' and phone like '088%'
```

	num	names	email	phone	client_add
1	638424	Alexandar Todorov	alex69@gmail.com	088 9123635	Mladost 104
2	935723	Radomir Ivanov	eb_ach@gmail.com	088 7523463	Elenovo 87
3	123074	Hristo Dimitrov	jesus_with_u@gmail.com	088 6672737	General Racovski 89

b. Заявки върху две и повече релации

- Изкарайте всички IT продукти с марка “Asus” или “Acer”, принтирайте техния сериен номер, марка и модел:

```
SELECT tech.ser_num as Serial, brand as Brand, model as Model
FROM it_tech, tech
WHERE tech.ser_num = it_tech.ser_num and (brand = 'ASUS' or brand = 'ACER')
```

	Serial	Brand	Model
1	10000000	ACER	ASPIRE TC-885 DT.BAPEX.013
2	10000001	ACER	ASPIRE C22-860 AIO
3	10000002	ACER	Nitro N50-110 DG.E1FEX.007
4	10000005	ACER	NITRO N50-600
5	10000007	ASUS	UX362FA-EL207T
6	10000008	ASUS	G531GT-AL010
7	10000009	ASUS	G731GU-EV011
8	10000010	ACER	NITRO 5 AN515-54-7064
9	10000011	ACER	HELIOS 300 PH317-53-71U2
10	10000012	ACER	HELIOS 300 PH315-52-707V
11	10000013	ASUS	UX434FL-A6009T

- Изкарайте всички клиенти, които са закупили настолен компютър (PC), принтирайте имената им, както и сериен номер, марка и модел на всеки продукт:

```
SELECT names as Names, tech.ser_num as Product_Serial, tech.brand as Brand, tech.model as Model
FROM tech, client, payment, included_in
WHERE type_tech = 'PC' and tech.ser_num = included_in.ser_num and included_in.pay_code = payment.code and client.email = payment.client_email
```

	Names	Product_Serial	Brand	Model
1	Gergana Jordanova	10000000	ACER	ASPIRE TC-885 DT.BAPEX.013
2	Hristo Dimitrov	10000005	ACER	NITRO N50-600
3	Martin Hristov	10000003	LEN...	T540-15ICK 90LW004SRM

c. Подзаявки

- Намерете и принтирайте каква обща парична сума уеб сайтът е успял да спечели от всички покупки:

```
SELECT SUM(Price_tag.price) as Money
FROM included_in,
(SELECT ser_num as num , price
FROM tech) as Price_tag
WHERE included_in.ser_num=Price_tag.num
```

	Money
1	27574.89

- Намерете всички продукти, които са поръчани (ordered) и принтирайте тяхната марка, модел, тип и цена, подредени по марка:

```
SELECT tech.brand as Brand, tech.model as Model, tech.type_tech as Type, tech.price as Price
FROM tech,
(SELECT included_in.ser_num
FROM included_in,
(SELECT code
FROM payment
WHERE pay_status='ordered') as Ordered
WHERE included_in.pay_code=Ordered.code) as Product
WHERE Product.ser_num=tech.ser_num
ORDER BY Brand
```

	Brand	Model	Type	Price
1	ACER	NITRO N50-600	PC	1599.00
2	APPLE	IPHONE 11 BLACK	Smartphone	1590.00
3	ASTRO	A50 WS+Base P...	Headset	599.99
4	ASUS	G731GU-EV011	Laptop	2449.00
5	ASUS	UX434FL-A6009T	Laptop	2199.00
6	ASUS	G531GT-AL010	Laptop	1999.00
7	BEKO	Rdsa 240 K20W	Refrigerator	359.00
8	CANYON	CNE-CBTSP6	Bluetooth ...	34.90
9	GORENJE	Gorenje E5121Wh	Cooker	419.00
10	HUAWEI	NOVA 5T BLUE	Smartphone	649.00
11	JAMO	S-626 Black Ash	Three-Stri...	599.00
12	LENOVO	T540-15ICK 90L...	PC	1399.00
13	NEO	E-6065Mf-Tr	Cooker	399.00
14	SAMSU...	GALAXY S10 128...	Smartphone	1150.00

d. Съединения

- Намерете имейлите на всички клиенти, които са поръчали или заплатили бяла техника с мощност 0 W, както и адресите на магазините, от които са ги взели:

```
SELECT DISTINCT Clients.shop_add, payment.client_email
FROM (SELECT Address.ser_num, Address.shop_add, included_in.pay_code
FROM
(SELECT DISTINCT stored_in.shop_add, Product.ser_num
FROM
(SELECT ser_num
FROM white_tech
WHERE power_watts='0 W') as Product
JOIN stored_in ON stored_in.ser_num=Product.ser_num) as Address
JOIN included_in ON Address.ser_num=included_in.ser_num) as Clients
JOIN payment ON Clients.pay_code=payment.code
```

	shop_add	client_email
1	Burgas, Slaveikov 94	alex69@gmail.com
2	Ruse, Hristo Botev 38	policaibg@abv.bg
3	Sofia, Pancho Vladi...	xX_kote_Xx@abv.bg

- Принтирайте цялата информация на всички клиенти, които са закупували продукти от магазини в София:

```
SELECT *
FROM (SELECT Address.num, Address.phone, Address.shop_add, purchased_from.client_email
FROM
(SELECT *
FROM shop
WHERE shop.shop_add='Sofia, Alabin 35'
OR shop.shop_add='Sofia, Pancho Vladigerov 75') AS Address
JOIN purchased_from ON Address.shop_add=purchased_from.shop_add) AS Email
JOIN client ON Email.client_email=client.email
```

	num	phone	shop_add	client_email	num	names	email	phone	client_add
1	6	02 5791201	Sofia, Alabin 35	puss_slayer@gmail.com	538423	Milen Cvetanov	puss_slayer@gmail.com	089 9123588	Vasil Levski 70
2	4	02 8170102	Sofia, Pancho Vladigerov 75	tarator@gmail.com	738428	Todor Georgiev	tarator@gmail.com	089 7672727	Tsveten konets 27
3	4	02 8170102	Sofia, Pancho Vladigerov 75	xX_kote_Xx@abv.bg	738426	Antoaneta Ve...	xX_kote_Xx@abv.bg	088 1263574	Svoboda 14

e. Групиране и агрегация

- Намерете каква сума трябва да заплати всеки клиент:

```
SELECT payment.client_email as Email, SUM(Money.price) AS Money
FROM (SELECT included_in.pay_code, SUM(tech.price) AS price
FROM included_in JOIN tech ON (tech.ser_num=included_in.ser_num)
GROUP BY pay_code) AS Money
JOIN payment ON payment.code=Money.pay_code
GROUP BY payment.client_email
```

	Email	Money
1	alex69@gmail.com	2818.00
2	eb_ach@gmail.com	5348.00
3	gergana_1999@abv...	2789.00
4	jesus_with_u@gmai...	2052.90
5	patriot_bg@abv.bg	1317.99
6	polica@abv.bg	9996.00
7	puss_slayer@gmail....	958.00
8	samodiva_69@abv....	49.00
9	tarator@gmail.com	599.00
10	xX_kote_Xx@abv.bg	1647.00

- Намерете колко IT продукти е закупил всеки клиент:

```
SELECT NewProducts.client_email as Email, SUM(NewProducts.NumberProducts) AS Products
FROM
(SELECT payment.client_email, Numbers.NumberProducts
FROM
(SELECT Products.pay_code, COUNT(Products.ser_num) AS NumberProducts
FROM(SELECT included_in.pay_code, it_tech.ser_num
FROM it_tech
RIGHT JOIN included_in ON included_in.ser_num=it_tech.ser_num) AS Products
GROUP BY Products.pay_code) AS Numbers
RIGHT JOIN payment ON payment.code=Numbers.pay_code) AS NewProducts
GROUP BY NewProducts.client_email
```

	Email	Products
1	alex69@gmail.com	1
2	eb_ach@gmail.com	3
3	gergana_1999@abv...	2
4	patriot_bg@abv.bg	1
5	polica_i@abv.bg	3
6	puss_slayer@gmail....	0
7	samodiva_69@abv....	0
8	tarator@gmail.com	0

5. Важни заключения върху проекта

а. Ограничения

- Към tech:

```
ALTER TABLE tech ADD CONSTRAINT PK_tech PRIMARY KEY (ser_num);
```

```
ALTER TABLE tech ADD CONSTRAINT chk_ser_tech CHECK (LEN(ser_num) = 8);
```

```
ALTER TABLE tech ADD CONSTRAINT df_type_tech DEFAULT 'undefined' FOR type_tech;
```

```
ALTER TABLE tech ADD CONSTRAINT chk_price_tech CHECK (price >= 0);
```

- Към audio_tech:

```
ALTER TABLE audio_tech ADD CONSTRAINT PK_audiotech PRIMARY KEY (ser_num);
```

```
ALTER TABLE audio_tech ADD CONSTRAINT FK_audiotech_tech FOREIGN KEY (ser_num)
REFERENCES tech (ser_num) ON DELETE Cascade ON UPDATE Cascade;
```

```
ALTER TABLE audio_tech ADD CONSTRAINT chk_ser_audiotech CHECK (LEN(ser_num) = 8);
```

- Към purchased_from:

```
ALTER TABLE purchased_from ADD CONSTRAINT FK_purfrom_client FOREIGN KEY
(client_email) REFERENCES client (email) ON DELETE Cascade ON UPDATE Cascade;
```

```
ALTER TABLE purchased_from ADD CONSTRAINT FK_purfrom_shop FOREIGN KEY (shop_add)
REFERENCES shop (shop_add) ON DELETE Cascade ON UPDATE Cascade;
```

```
ALTER TABLE purchased_from ADD CONSTRAINT chk_email_purfrom CHECK (client_email like
'@%.%' and LEN(client_email) >= 6);
```

b. Тригери

- Тригер за въвеждане в таблицата stored_in и обновяване на броя налични продукти (available items) :

```
IF EXISTS (SELECT * FROM sys.objects WHERE [name] = N'TR_insert_storedin' and [type] = 'TR')
BEGIN
```

```
DROP TRIGGER TR_insert_storedin;
```

```
END
```

```
GO
```

```
CREATE TRIGGER TR_insert_storedin ON stored_in INSTEAD OF INSERT AS
IF TRIGGER_NESTLEVEL() <= 1
BEGIN
```

```
IF EXISTS (
(
SELECT shop_add, ser_num
FROM stored_in
)
INTERSECT
(
SELECT shop_add, ser_num
FROM inserted
)
)
BEGIN
```

```
UPDATE stored_in
SET stored_in.count_available = stored_in.count_available + inter.Total
FROM (
SELECT inserted.shop_add, inserted.ser_num, SUM(inserted.count_available) AS Total
FROM inserted
```

```
GROUP BY inserted.shop_add, inserted.ser_num
) AS inter
WHERE stored_in.shop_add = inter.shop_add AND stored_in.ser_num = inter.ser_num
```

```
END
```

```
IF EXISTS (
SELECT shop_add, ser_num
FROM inserted
EXCEPT
SELECT shop_add, ser_num
FROM stored_in
)
BEGIN
```

```
INSERT INTO stored_in (shop_add, ser_num, count_available)
SELECT ShopAddress, SerialNumber, CountAvailable
FROM (
SELECT inserted.shop_add AS ShopAddress, inserted.ser_num AS SerialNumber,
SUM(inserted.count_available) AS CountAvailable
FROM (
SELECT shop_add, ser_num
FROM inserted
EXCEPT
SELECT shop_add, ser_num
FROM stored_in
) AS diff, inserted
WHERE diff.ser_num = inserted.ser_num AND diff.shop_add = inserted.shop_add
GROUP BY inserted.shop_add, inserted.ser_num
) AS random
```

```
END
```

```
END
GO
```

с. Изгледи (за целите на този проект не бяха от особена необходимост):

```
CREATE VIEW V_join_techittech AS  
SELECT it_tech.ser_num  
FROM it_tech JOIN tech  
ON tech.ser_num = it_tech.ser_num;  
GO
```

d. Индекси

```
CREATE INDEX idx_tech_model ON tech (model);
```