

Tables

```
create table product (
    productKey int primary key,
    productName varchar(255),
    price int,
    instant_delivery bit,
    seller varchar(255),
    brand varchar(255),
    available int,
    numOfView int,
    numOfSales int,
    score int,
    promotion_status int,
    outstanding_offer bit,
    discount int
);

create table userTable (
    mobile char(10) primary key,
    email varchar(255),
    username varchar(255),
    credit_card_number int,
    user_address varchar(255)
);

create table form (
    code int primary key identity(1,1),
    topic varchar(255),
    positive_points varchar(255),
    negative_points varchar(255),
    context varchar(255),
    includes_picture bit,
    suggest_to_others bit,
    productKey int foreign key references product(productKey) on delete cascade,
    user_mobile char(10) foreign key references userTable(mobile) on delete cascade
);

create table digitalProduct (
    code int primary key,
    guaranted bit,
    size int,
    special_functions varchar(255),
    model varchar(255),
    weightValue int,
    productKey int foreign key references product(productKey) on delete cascade
);

create table mobileAccessory (
    code int primary key,
    usableFor varchar(255),
    productKey int foreign key references digitalProduct(code) on delete cascade
);
```

```
create table mobilePhone (  
    model varchar(255),  
    networkConnection varchar(255),  
    OStype varchar(255),  
    OSversion varchar(255),  
    resolution int,  
    technology varchar(255),  
    RAM int,  
    internalMemory int,  
    productKey int foreign key references digitalProduct(code) on delete cascade  
);
```

```
create table computerAccessory (  
    code int primary key,  
    hasUSBport bit,  
    hasHTMLport bit,  
    hasDVIport bit,  
    productKey int foreign key references digitalProduct(code) on delete cascade  
);
```

```
create table laptop (  
    RAM int,  
    graphic int,  
    internalMemory int,  
    displaySize int,  
    resolution int,  
    OS varchar(255),  
    GPUproducer varchar(255),  
    CPU varchar(255),  
    RAMtype varchar(255),  
    touchScreen bit,  
    productKey int foreign key references digitalProduct(code) on delete cascade  
);
```

```
create table powerBank (  
    numOfPort int,  
    capacity int,  
    material varchar(255),  
    outputCurrency int,  
    powerDelivery bit,  
    productKey int foreign key references mobileAccessory(code) on delete cascade  
);
```

```
create table data_storage_hardware (  
    code int primary key,  
    capacity int,  
    water_proof bit,  
    shock_proof bit,  
    productKey int foreign key references computerAccessory(code) on delete cascade  
);
```

```

create table keyboard (
    connectionType varchar(255),
    portType varchar(255),
    hasMouse bit,
    has_mic_handsfreePort bit,
    water_proof bit,
    productKey int foreign key references computerAccessory(code) on delete cascade
);

create table external_hard_disk (
    capacity int,
    water_proof bit,
    shock_proof bit,
    productKey int foreign key references data_storage_hardware(code) on delete cascade
);

create table mobileStand (
    chargable bit,
    wirelessCharge bit,
    rotation360 int,
    productKey int foreign key references mobileAccessory(code) on delete cascade
);

create table monitor (
    hasSpeaker bit,
    panelType varchar(255),
    background_lighting varchar(255),
    display varchar(255),
    resolution int,
    response_rate int,
    productKey int foreign key references computerAccessory(code) on delete cascade
);

create table bagAndCover (
    material varchar(255),
    structure varchar(255),
    productKey int foreign key references mobileAccessory(code) on delete cascade
);

create table assembledCase (
    internal_memory int,
    RAM int,
    memoryType varchar(255),
    CPUproducer varchar(255),
    productKey int foreign key references computerAccessory(code) on delete cascade
);

```

GetPhone

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
use digiDB;

-- Query 1: GetPhone -> RAM > A, price < B
create procedure GetPhone (@A int, @B int) as
begin
    select *
    from product p
    inner join digitalProduct d
    on p.productKey = d.productKey
    inner join mobilePhone m
    on d.code = m.productKey
    where p.price < @B and m.RAM > @A;
end

execute GetPhone @A = 4, @B = 10000;
```

100 %

Results Messages

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status
1	17	Samsung Galaxy A51	5400	1	NULL	Samsung	126	NULL	742	NULL	0
2	20	Redmi Note 8 Pro m1906g7G	6980	0	NULL	Xiaomi	326	NULL	54	NULL	0

Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 2 rows

GetLaptop

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 2: GetLaptop -> ASUS or lenovo, graphic >= A, weight < B
create procedure GetLaptop (@A int, @B int) as
begin
    select *
    from product p
    inner join digitalProduct d
    on p.productKey = d.productKey
    inner join laptop l
    on d.code = l.productKey
    where (p.brand = 'asus' or p.brand = 'Lenovo') and
           weightValue < @B and l.graphic >= @A;
end

execute GetLaptop @A = 2, @B = 2;
```

100 %

Results Messages

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status	outstanding_off
1	21	laptop	10000	NULL	NULL	asus	97	NULL	100	NULL	2	1

Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 1 rows

GetSortByPrice

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 3: GetSortByPrice -> sort laptops based on number of sales
create procedure GetSortByPrice as
begin
    select p.numOfSales, p.brand, p.price
    from product p
    inner join digitalProduct d
    on p.productKey = d.productKey
    inner join laptop l
    on d.code = l.productKey
    order by p.numOfSales;
end

execute GetSortByPrice;
```

100 %

Results Messages

	numOfSales	brand	price
1	33	hp	20000
2	45	apple	60000
3	75	acer	15000
4	80	asus	23000
5	100	asus	10000

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 5 rows

delAssembleCase

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 4: delAssembleCase -> delete where RAM < A, memory < B
create procedure delAssembleCase (@A int, @B int) as
begin
    delete
    from product
    where product.productKey in (select p1.productKey
                                from product p1, digitalProduct d, computerAccessory c, assembledCase a
                                where p1.productKey = d.productKey and d.code = c.productKey and
                                      a.productKey = c.code and a.internal_memory < @B and RAM < @A);
end

select * from assembledCase;
execute delAssembleCase @A = 16, @B = 256;
select * from assembledCase;
```

83 %

Results Messages

	internal_memory	RAM	memoryType	CPUproducer	productKey
1	256	64	NULL	NULL	11
2	64	16	NULL	NULL	12
3	128	4	NULL	NULL	13
4	1024	32	NULL	NULL	14
5	256	64	NULL	NULL	15

	internal_memory	RAM	memoryType	CPUproducer	productKey
1	256	64	NULL	NULL	11
2	64	16	NULL	NULL	12
3	1024	32	NULL	NULL	14
4	256	64	NULL	NULL	15

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 9 rows

GetBestOffer

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s....JHLHRJG\sana (53))

```
-- Query 5: GetBestOffer -> best offer, 2 <= price <= 5
create procedure GetBestOffer as
begin
    select *
    from product p
    where outstanding_offer = 1 and price >= 2000 and price <= 5000;
end

execute GetBestOffer;
```

100 %

Results Messages

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status	outstandir
1	26	keyboard	5000	NULL	NULL	green	0	NULL	11	NULL	2	1
2	29	keyboard	3200	NULL	NULL	TSCO	5	NULL	80	NULL	2	1
3	30	keyboard	2600	NULL	NULL	VERITY	6	NULL	120	NULL	2	1
4	32	monitor	3000	NULL	NULL	LG	1	NULL	23	NULL	0	1
5	33	monitor	5000	NULL	NULL	x-vision	2	NULL	40	NULL	1	1
6	35	monitor	2500	NULL	NULL	LG	1	NULL	47	NULL	1	1
7	37	assembledCase	4000	NULL	NULL	GREEN	1	NULL	8	NULL	2	1
8	44	external_hard_disk	2500	NULL	NULL	GREEN	14	NULL	500	NULL	2	1

< >

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 8 rows

CountByCpu

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 6: CountByCPU -> number of laptops that are not Acer, CPU = core i7
create procedure CountByCPU as
begin
    select count(*)
    from product p
    inner join digitalProduct d
    on p.productKey = d.productKey
    inner join laptop l
    on d.code = l.productKey
    where l.CPU = 'core i7' and p.brand != 'acer';
end

execute CountByCPU;

select p.productKey, p.productName, p.price, p.brand, l.cpu
from product p
inner join digitalProduct d
on p.productKey = d.productKey
inner join laptop l
on d.code = l.productKey
```

83 %

Results Messages

	(No column name)
1	1

	productKey	productName	price	brand	cpu
1	21	laptop	10000	asus	celeron
2	22	laptop	15000	acer	core i7
3	23	laptop	23000	asus	core i7
4	24	laptop	60000	apple	core i3
5	25	laptop	20000	hp	core i5

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 6 rows

UpdatePhonesOffer

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRIG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRIG\sana (53))

DigiKala_Project.s...-JHLHRIG\sana (53))

```
-- Query 7: UpdatePhonesOffer -> update sale between 5 and 10 to no sale
create procedure UpdatePhonesOffer as
begin
    update product set promotion_status = 0, discount = 0
    where discount >= 5 and discount <= 10 and promotion_status = 1 and
    (product.productKey in (select p.productKey from product p
        inner join digitalProduct d
        on p.productKey = d.productKey
        inner join mobilePhone m
        on d.code = m.productKey) or
    product.productKey in (select p.productKey from product p
        inner join digitalProduct d
        on p.productKey = d.productKey
        inner join mobileAccessory m
        on d.code = m.productKey))
end

select * from product where discount >= 5 and discount <= 10 and promotion_status = 1;
execute UpdatePhonesOffer;
select * from product where productKey = 4 or productKey = 13;
```

100 %

Results Messages

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status	outstanding_offer	discount
1	4	bag&cover	170	NULL	NULL	Other	53	NULL	10	NULL	1	0	10
2	13	mobileStand	10	NULL	NULL	Other	93	NULL	104	NULL	1	0	7

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status	outstanding_offer	discount
1	4	bag&cover	170	NULL	NULL	Other	53	NULL	10	NULL	0	0	0
2	13	mobileStand	10	NULL	NULL	Other	93	NULL	104	NULL	0	0	0

Query executed successfully.

DESKTOP-JHLHRIG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRIG\sana (53)) digiDB 00:00:00 2 rows

AveragePriceOfMonitor

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 8: AveragePriceOfMonitor -> screen size > A, Brand = B
create procedure AveragePriceOfMonitor (@A int, @B varchar(255)) as
begin
    select avg(p.price)
    from product p
    inner join digitalProduct d
    on p.productKey = d.productKey
    inner join computerAccessory c
    on d.code = c.productKey
    inner join monitor m
    on m.productKey = c.code
    where d.size > @A and p.brand = @B;
end

execute AveragePriceOfMonitor @A = 27, @B = 'LG';

select p.productKey, p.price, p.brand, d.size
from product p
inner join digitalProduct d
on p.productKey = d.productKey
inner join computerAccessory c
on d.code = c.productKey
inner join monitor m
on m.productKey = c.code
```

83 %

Results Messages

	(No column name)
1	2750

	productKey	price	brand	size
1	31	2000	LG	19
2	32	3000	LG	28
3	33	5000	x-vision	26
4	34	6000	asus	27
5	35	2500	LG	30

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 6 rows

CountPhoneByCameraAndRam

Solution1 - DigiKala_Project.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Project.s...-JHLHRJG\sana (53))

```
-- Query 9: CountPhoneByCameraAndRam -> resolution > A or B <= internal memory <= C
create procedure CountPhoneByCameraAndRam (@A int, @B int, @C int) as
begin
    select count(*)
    from mobilePhone m
    where m.resolution > @A or (m.internalMemory >= @B and m.internalMemory <= @C);
end

execute CountPhoneByCameraAndRam @A = 12, @B = 32, @C = 128;

select * from mobilePhone;
```

100 %

Results Messages

(No column name)

1	4
---	---

	model	networkConnection	OSType	OSVersion	resolution	technology	RAM	internalMemory	productKey
1	NULL	NULL	Android	8	13	NULL	8	128	16
2	NULL	NULL	Android	8	48	NULL	8	64	17
3	NULL	NULL	iOS	14	12	NULL	6	256	18
4	NULL	NULL	iOS	14	12	NULL	4	128	19
5	NULL	NULL	Android	8	64	NULL	6	64	20

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 6 rows

FullPcPrice

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Queries.s...JHLHRJG\sana (53))

```
-- Query 10: FullPCPrice -> monitor + case + keyboard < 30
create view product_computerAccessory as (
    select c.code, p.price, p.brand
    from product p
    inner join digitalProduct d on p.productKey = d.productKey
    inner join computerAccessory c on d.code = c.productKey
);

create procedure FullPCPrice (@A int) as
begin
    select m1.price + c1.price + k1.price as price,
           m1.brand as monitorBrand, c1.brand as caseBrand, k1.brand as keyboardBrand
    from product_computerAccessory m1
    inner join monitor m on m1.code = m.productKey,
    product_computerAccessory c1
    inner join assembledCase c on c1.code = c.productKey,
    product_computerAccessory k1
    inner join keyboard k on k1.code = k.productKey
    where m1.price + k1.price + c1.price < @A;
end

execute FullPCPrice @A = 30000;
```

91 %

Results Messages

	price	monitorBrand	caseBrand	keyboardBrand
1	11000	LG	GREEN	green
2	14000	LG	DELL	green
3	16500	LG	TechZone	green
4	15000	LG	HP	green
5	12000	LG	GREEN	green
6	15000	LG	DELL	green
7	17500	LG	TechZone	green
8	16000	LG	HP	green
9	14000	x-vision	GREEN	green
10	17000	x-vision	DELL	green
11	19500	x-vision	TechZone	green
12	18000	x-vision	HP	green
13	15000	asus	GREEN	green

Query executed... DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG\sana (53)) | digiDB | 00:00:00 | 100 rows

GetCaseAndCoverEndOffer

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Queries.s...JHLHRJG\sana (53))

```
-- Query 11: GetCaseAndCoverEndOffer -> promotiom status = 0, os of mobile = android8
create procedure GetCaseAndCoverEndOffer as
begin
    select p.productKey, p.productName, p.promotion_status, m.usableFor,
           mobile.OSType, mobile.OSversion
    from product p
    inner join digitalproduct d on p.productKey = d.productKey
    inner join mobileAccessory m on m.productKey = d.code
    inner join bagAndCover b on b.productKey = m.code,
    product p1
    inner join digitalproduct d1 on p1.productKey = d1.productKey
    inner join mobilePhone mobile on d1.code = mobile.productKey
    where mobile.OSType = 'Android' and mobile.OSversion = 8 and
           m.usableFor = p1.productName and p.promotion_status = 0;
end

execute GetCaseAndCoverEndOffer;
```

100 %

Results Messages

	productKey	productName	promotion_status	usableFor	OSType	OSversion
1	1	bag&cover	0	Samsung Galaxy A51	Android	8
2	3	bag&cover	0	Samsung Galaxy S5	Android	8
3	5	bag&cover	0	Samsung Galaxy S5	Android	8

✓ Query executed su... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (53) | digiDB | 00:00:00 | 3 rows

GetPowerBank

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG...

DigiKala_Queries.s...JHLHRJG\sana (52))*

```
-- Query 12: GetPowerBank -> delete powerBank 5<=capacity<=10, (numOfPort < 3 or price > 200)
create procedure GetPowerBank as
begin
    delete from product
    where product.productKey in (select p. productKey from product p
                                inner join digitalProduct d
                                on p.productKey = d.productKey
                                inner join mobileAccessory m
                                on d.code = m.productKey
                                inner join powerBank pow
                                on m.code = pow.productKey
                                where pow.capacity >= 5 and pow.capacity <= 10
                                and (pow.numOfPort < 3 or p.price > 200));
end

select p.productKey, pow.capacity, pow.numOfPort, p.price
from product p
inner join digitalProduct d on p.productKey = d.productKey
inner join mobileAccessory m on d.code = m.productKey
inner join powerBank pow on m.code = pow.productKey;

execute GetPowerBank;

select p.productKey, pow.capacity, pow.numOfPort, p.price
from product p
inner join digitalProduct d on p.productKey = d.productKey
inner join mobileAccessory m on d.code = m.productKey
inner join powerBank pow on m.code = pow.productKey;
```

75 %

Results Messages

	productKey	capacity	numOfPort	price
1	6	10	2	590
2	7	4	1	760
3	8	10	3	1200
4	9	7	2	850
5	10	4	2	1270

	productKey	capacity	numOfPort	price
1	7	4	1	760
2	10	4	2	1270

✓ Query executed success... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (52) | digiDB | 00:00:00 | 7 rows

SumPriceLaptopAndPhone

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG...

DigiKala_Queries.s...JHLHRJG\sana (52))

```
-- Query 13: SumPriceLaptopAndPhone -> price of laptop + mobile, not the same brand
--mobile instant delivery, laptop RAM > A
create procedure SumPriceLaptopAndPhone (@A int) as
begin
    select laptop.price + mobile.price as sumPrice
    from product laptop
    inner join digitalProduct d1 on laptop.productKey = d1.productKey
    inner join laptop l on d1.code = l.productKey,
    product mobile
    inner join digitalProduct d2 on mobile.productKey = d2.productKey
    inner join mobilePhone m on d2.code = m.productKey
    where laptop.brand != mobile.brand and mobile.instant_delivery = 1 and l.RAM > @A;
end

select laptop.price as laptopPrice, mobile.price as mobilePrice, mobile.instant_delivery,
    l.RAM, laptop.brand as laptopBrand, mobile.brand as mobileBrand
from product laptop
inner join digitalProduct d1 on laptop.productKey = d1.productKey
inner join laptop l on d1.code = l.productKey,
product mobile
inner join digitalProduct d2 on mobile.productKey = d2.productKey
inner join mobilePhone m on d2.code = m.productKey

execute SumPriceLaptopAndPhone @A = 6;
```

91 %

Results Messages

1	10000	15800	1	4	asus	Samsung
2	15000	15800	1	4	acer	Samsung
3	20700	15800	1	8	asus	Samsung
4	54000	15800	1	8	apple	Samsung
5	20000	15800	1	4	hp	Samsung
6	10000	5400	1	4	asus	Samsung
7	15000	5400	1	4	acer	Samsung
8	20700	5400	1	8	asus	Samsung

	sumPrice
1	36500
2	26100
3	70200
4	69800
5	59400

Query executed success... DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG\sana (52)) | digiDB | 00:00:00 | 5 rows

UpdateLaptopPrice

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG...

DigiKala_Queries.s...JHLHRJG\sana (52))*

```
-- Query 14: UpdateLaptopPrice -> A percent off on laptops more expensive than B
create procedure UpdateLaptopPrice (@A int, @B int) as
begin
    update product set price = (price * (100-@A))/100, promotion_status = 1, discount = @A
    where product.productKey in (select p.productKey
                                from product p
                                inner join digitalProduct d
                                on p.productKey = d.productKey
                                inner join laptop l
                                on d.code = l.productKey
                                where p.price > @B);
end

select p.productKey, p.promotion_status, p.price, p.discount from product p
inner join digitalProduct d on p.productKey = d.productKey
inner join laptop l on d.code = l.productKey

execute UpdateLaptopPrice @A = 10, @B = 20000;

select p.productKey, p.promotion_status, p.price, p.discount from product p
inner join digitalProduct d on p.productKey = d.productKey
inner join laptop l on d.code = l.productKey
```

100 %

Results Messages

	productKey	promotion_status	price	discount
1	21	2	10000	50
2	22	2	15000	30
3	23	2	23000	10
4	24	2	60000	0
5	25	2	20000	0

	productKey	promotion_status	price	discount
1	21	2	10000	50
2	22	2	15000	30
3	23	1	20700	10
4	24	1	54000	10
5	25	2	20000	0

Query executed succes... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (52) | digiDB | 00:00:00 | 10 rows

ShowCom

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG\sana (52))

DigiKala_Queries.s...JHLHRJG\sana (52))

```
-- Query 15: ShowCom -> product, less than 2 opinions or opinion includes pic
create procedure ShowCom as
begin
    select *
    from product p
    where p.productKey not in (select f.productKey from form f
                              where p.productKey = f.productKey and f.includes_picture = 0 and
                              (select count(*) from form f where p.productKey = f.productKey) >= 2);

end

select * from form;
execute ShowCom;
```

100 %

Results Messages

	code	topic	positive_points	negative_points	context	includes_picture	suggest_to_others	productKey	user_mobile
1	1	NULL	NULL	NULL	NULL	1	NULL	24	1234567890
2	2	NULL	NULL	NULL	NULL	1	NULL	7	2345678901
3	3	NULL	NULL	NULL	NULL	0	NULL	32	3456789012
4	4	NULL	NULL	NULL	NULL	0	NULL	32	4567890123
5	5	NULL	NULL	NULL	NULL	0	NULL	7	4567890123
6	6	NULL	NULL	NULL	NULL	0	NULL	7	1234567890
7	7	NULL	NULL	NULL	NULL	0	NULL	32	5678901234

	productKey	productName	price	instant_delivery	seller	brand	available	numOfView	numOfSales	score	promotion_status	outstanding_offer	discount
20	23	laptop	23...	NULL	NULL	asus	26	NULL	80	NULL	2	1	10
21	24	laptop	60...	NULL	NULL	apple	50	NULL	45	NULL	2	1	0
22	25	laptop	20...	NULL	NULL	hp	0	NULL	33	NULL	2	1	0
23	26	keyboard	5000	NULL	NULL	green	0	NULL	11	NULL	2	1	0
24	27	keyboard	800	NULL	NULL	green	1	NULL	22	NULL	1	1	20
25	28	keyboard	1200	NULL	NULL	TSCO	1	NULL	78	NULL	1	1	20
26	29	keyboard	3200	NULL	NULL	TSCO	5	NULL	80	NULL	2	1	15
27	30	keyboard	2600	NULL	NULL	VERITY	6	NULL	120	NULL	2	1	0
28	31	monitor	2000	NULL	NULL	LG	7	NULL	134	NULL	2	0	11
29	33	monitor	5000	NULL	NULL	x-vision	2	NULL	40	NULL	1	1	0
30	34	monitor	6000	NULL	NULL	asus	5	NULL	50	NULL	2	1	0
31	35	monitor	2000	NULL	NULL	LG	1	NULL	47	NULL	1	1	20
32	36	assembled...	55...	NULL	NULL	GREEN	4	NULL	17	NULL	1	0	0
33	37	assembled...	3200	NULL	NULL	GREEN	1	NULL	8	NULL	1	1	20

Query executed successfully. DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB DESKTOP-JHLHRJG\sana (52) | digiDB | 00:00:00 | 41 rows

Update20Percent

Solution1 - DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHL...

DigiKala_Queries.s...JHLHRJG\sana (52)

```
-- Query 16: Update20Percent -> 20 percent off on products with availability 1
create procedure Update20Percent as
begin
    update product set price *= 0.8, promotion_status = 1, discount = 20
    where product.available = 1;
end

select price, promotion_status, discount from product where available = 1;
execute Update20Percent;
select price, promotion_status, discount from product where available = 1;
```

100 %

Results Messages

	price	promotion_status	discount
1	590	1	15
2	850	0	0
3	15000	2	30
4	1000	2	5
5	1500	2	10
6	3000	0	0
7	2500	1	0
8	4000	2	0
9	9500	2	5
10	1500	1	0
11	3000	1	0

	price	promotion_status	discount
1	472	1	20
2	680	1	20
3	12000	1	20
4	800	1	20
5	1200	1	20
6	2400	1	20
7	2000	1	20
8	3200	1	20
9	7600	1	20
10	1200	1	20
11	2400	1	20

✓ Query executed s... | DESKTOP-JHLHRJG\CACTUSEXPRESS... | DESKTOP-JHLHRJG\sana (52) | digiDB | 00:00:00 | 22 rows

delS5Case

DigiKala_Queries.sql - DESKTOP-JHLHRJG\CACTUSEXPRESSNEW.digiDB (DESKTOP-JHLHRJG\sana (52)) - Microsoft SQL Serv

File Edit View Query Project Tools Window Help

digiDB Execute

product.sql - DESK...JHLHRJG\sana (53)) digitalProduct.sql...-JHLHRJG\sana (54)) mobileAcces
computerAccessory...HLHRJG\sana (57)) laptop.sql - DESKT...JHLHRJG\sana (58)) form.sql - DE

```
-- Query 17: delS5Case -> delete covers for Samsung S5
create procedure delS5Case as
begin
    delete from product
    where product.productKey in (select p.productKey from product p
                                inner join digitalProduct d
                                on p.productKey = d.productKey
                                inner join mobileAccessory m
                                on d.code = m.productKey
                                inner join bagAndCover b
                                on b.productKey = m.code
                                where m.usableFor = 'Samsung Galaxy S5')
end

select * from mobileAccessory m inner join bagAndCover b on b.productKey = m.code;
execute delS5Case;
select * from mobileAccessory m inner join bagAndCover b on b.productKey = m.code;
```

91 %

Results Messages

	code	usableFor	productKey	material	structure	productKey
1	1	Samsung Galaxy A51	1	NULL	NULL	1
2	2	iPhone 11 Pro Max	2	NULL	NULL	2
3	3	Samsung Galaxy S5	3	NULL	NULL	3
4	4	Xiaomi Redmi Note 8 Pro	4	NULL	NULL	4
5	5	Samsung Galaxy S5	5	NULL	NULL	5

	code	usableFor	productKey	material	structure	productKey
1	1	Samsung Galaxy A51	1	NULL	NULL	1
2	2	iPhone 11 Pro Max	2	NULL	NULL	2
3	4	Xiaomi Redmi Note...	4	NULL	NULL	4

Query executed successfully.