

Adatbázisok (BMEGEMIBMAB)

Félévközi feladat

WEBSHOP ADATBÁZIS

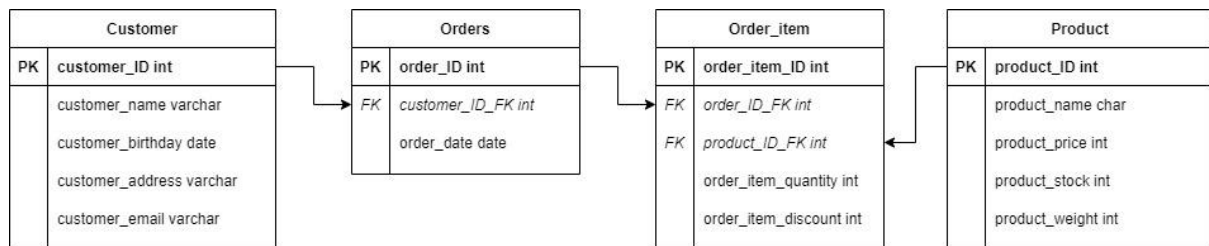
Körtvélyessy Csongor - SI0S1A

Veisz Andor - XF6KU2

Gajdos Marcell András - QVUBJU

2023. december 8.

ETK Modell:



A customer, order és product táblák rendre a vevők, a rendelések és a termékek adatait tartalmazzák, viszont a 3. Normál forma eléréséhez szükség volt egy order_item táblára is. Ebben a táblában tárolódik el minden megrendelt tétel, így lehetséges, hogy egy megrendeléshez több terméket tudjunk kapcsolni úgy, hogy egy terméket akár több megrendeléshez is kapcsolunk. Gyakorlatilag ez a tábla az adatbázis lelke.

2. heti feladat: táblák és adatok

- Létrehozzuk a táblákat a megfelelő tárolt oszlopokkal és kapcsolatokkal
- Generatív AI segítségével előállított adatokkal feltöltjük a táblákat
- Hozzáadunk egy telefonszám oszlopot a user táblához
- Mivel a feladat kért egy utólagosan hozzáadott kapcsolatot, az utolsó szükséges táblakapcsolást utólag oldjuk meg

3. heti feladat: lekérdezések

- Írunk egy egytáblás lekérdezést, ami azokat a vásárlókat választja ki, akik elmúltak már 18 évesek, majd a születésnapjuk alapján sorba rendezzük őket
- Ezután írunk még 5 különböző, egyszerre több táblát felhasználó lekérdezést. Ezek sorra:
 1. Egy-egy rendelés melyik vásárlóhoz kapcsolódik a rendelések szerint rendezve.
 2. Hány darabot rendeltek az egyes árukból, különböző leárazással és így mennyi a bevétel belőlük. Az áruk neve és a rajtuk lévő akció szerint rendeztük.
 3. Mennyit fizettek az egyes vásárlók összesen a vásárlásaik során. Ezt a eddig legtöbbet fizető vásárlótól csökkenő sorrendbe rendeztük.
 4. Az egyes termékekből mennyi marad a raktárban minden egyes vásárlást figyelembe véve. Itt a termékek neve alapján rendeztük.
 5. Az egyes rendelések tömege mennyi grammal a szállításhoz. Ezt a rendelések szerint rendeztük.

4. heti feladat: outer join, subselect

- Az outer join kulcsszóval az alábbi lekérdezéseket írtuk:
 - Ki rendelt több, mint 3 éve, és mikor?
 - A vásárlók melyik napokon rendeltek?
 - Melyik termékből, hány darabot rendeltek, egy-egy rendelés során?
- A következő lekérdezések subselecteket tartalmaznak:

- Termékek, amelyek többbe kerülnek, mint a termékek átlagos ára.
- A legdrágább termékből egy adott rendelésen belül hány darabot adtak el?
- Melyik termékből adtak el a legtöbbet, és mikor?

5. heti feladat: nézettábla, index

- Készítünk egy view-t egy előző lekérdezést felhasználva, amely azt mutatja, mennyi maradt a készleten adott termékekből
- A view-t felhasználjuk egy lekérdezésben, azokat a termékeket kiírva, amelyekből 30 alatti darabszám van készleten
- Készítünk egy indexet a customer tábla email oszlopából
- Egy lekérdezésben az egy bizonyos e-mail címmel rendelkező vásárló által rendelt termékeket listázzuk

6. heti feladat: update utasítások, tranzakció

- Elindítunk egy tranzakciót, majd elkezdjük változtatni a táblákat.
- Első változtatás: Átírjuk az összes beérkezett rendelést, ami 2015. nyarán érkezett be, mivel ekkor karbantartás volt. Így ezek a rendelések 2015. szeptember 1-jén érkeztek be.
- Második változtatás: Akik Laptop-ot rendeltek kapnak plusz 5% kedvezményt a rendelésükre.
- Harmadik változtatás: Növeljük azoknak a termékeknek az árát 10%-kal, amelyekből már több, mint 20 db-ot rendeltek.
- Ezután pedig befejezzük a tranzakciót.

7. heti feladat: függvény

- Az egyik függvény töröl egy rendelést, a hozzá tartozó rendelés tételekkel együtt
- A másik függvény a rendelésben kicseréli a rendelő személyét

SQL kód

-- dropolunk minden tablat, hogy tiszta lappal induljunk

drop table if exists order_items;

drop table if exists products;

drop table if exists orders;

drop table if exists customers;

drop view if exists maradek_stock;

-- legeneraljuk a tablakat

create table customers(

customer_ID int auto_increment primary key not null,

customer_name varchar(50) not null,

customer_email varchar(50) not null,

customer_birthday date not null,

customer_address varchar(250) not null);

create table products(

product_ID int auto_increment primary key not null,

product_name varchar(200) not null,

product_price int not null,

product_stock int not null,

product_weight double not null);

create table orders(

order_ID int auto_increment primary key not null, -- kell egy változot gyartani, ami FK lesz

customer_ID_FK int,

order_date date not null,

foreign key (customer_ID_FK) references customers(customer_ID) on delete set null); -- a változot amit gyartottunk osszekotjuk

a másik tablával, "on delete set null" annyit csinál, hogyha torlik a fő tablából a rekordot, amit itt hivatkozunk, akkor null-ra

állítja itt a mezőt

create table order_items(

order_item_ID int auto_increment primary key not null,

order_ID_FK int,

product_ID_FK int,

order_item_quantity int not null,

order_item_discount int not null,

foreign key (order_ID_FK) references orders(order_ID) on delete set null);

-- feltöltjük a tablakat

insert into customers (customer_name, customer_email, customer_birthday, customer_address)

values

('Sarah Johnson', 'sarah.johnson@example.com', '1978-06-11', '1234 Elm Street, Apt 101, Anytown, CA 98765'),

('John Smith', 'john.smith@example.com', '1985-04-23', '5678 Oak Avenue, Suite 205, Smalltown, NY 12345'),

('Jennifer Davis', 'jennifer.davis@example.com', '1990-09-15', '789 Maple Drive, Unit 301, Suburbia, TX 54321'),

('Michael Anderson', 'michael.anderson@example.com', '1982-11-30', '456 Pine Lane, Apt 202, Metro City, FL 67890'),

('Jessica Brown', 'jessica.brown@example.com', '1975-08-20', '8769 Cedar Street, Apt 3B, Nearbyville, OH 23456'),

('Brian Wilson', 'brian.wilson@example.com', '1988-02-14', '987 Elmwood Boulevard, Suite 102, Urbantown, IL 34567'),

('Amanda Miller', 'amanda.miller@example.com', '1983-07-03', '1010 Birch Court, Apt 501, Villageland, GA 87654'),

('David Davis', 'david.davis@example.com', '1979-01-17', '222 Pine Avenue, Unit 12A, Downtown, WA 65432'),

('Karen Johnson', 'karen.johnson@example.com', '1987-12-25', '3333 Oak Lane, Apt 401, Countryside, AZ 43210'),

('Matthew Thompson', 'matthew.thompson@example.com', '1995-03-08', '4444 Maple Road, Suite 10, Cityville, NM 21098'),

('Laura Clark', 'laura.clark@example.com', '1981-10-05', '5555 Cedar Street, Apt 301, Suburbia, MI 56789'),

('Robert Martin', 'robert.martin@example.com', '1980-09-12', '666 Elmwood Drive, Apt 202, Townsville, NJ 87678'),

('Mary Davis', 'mary.davis@example.com', '1976-07-18', '777 Pine Avenue, Unit 15B, Hamlet, CT 32109'),

('Daniel Harris', 'daniel.harris@example.com', '1992-05-14', '888 Birch Court, Apt 102, Downtown, NV 10987'),

('Sandra Smith', 'sandra.smith@example.com', '1984-04-09', '999 Oak Lane, Suite 101, Metropolis, SC 54321'),

('Paula Taylor', 'paula.taylor@example.com', '1989-08-26', '123 Maple Road, Apt 601, Nearbytown, VA 89012'),

('Kevin Johnson', 'kevin.johnson@example.com', '1986-06-01', '456 Cedar Drive, Apt 402, Uptown, OR 65432'),

('Jason Anderson', 'jason.anderson@example.com', '1980-12-04', '789 Elm Street, Apt 501, Suburbia, CA 34567'),

('Lisa White', 'lisa.white@example.com', '1977-09-19', '321 Pine Avenue, Unit 204, Countryside, TX 54321'),

('Brian Davis', 'brian.davis@example.com', '1993-02-28', '111 Birch Court, Apt 303, Cityville, FL 21098'),

('Karen Martin', 'karen.martin@example.com', '1985-11-10', '222 Maple Drive, Apt 202, Villageland, IL 87654'),

```

('Mark Thompson', 'mark.thompson@example.com', '1979-06-23', '333 Cedar Lane, Suite 10, Urbantown, OH
10987'),
('Catherine Smith', 'catherine.smith@example.com', '1988-03-17', '444 Elmwood Boulevard, Apt 305, Downtown, WA
43210'),
('John Davis', 'john.davis@example.com', '1990-12-30', '555 Birch Court, Apt 101, Countryside, GA 65432'),
('Jessica Williams', 'jessica.williams@example.com', '1981-07-21', '666 Maple Drive, Apt 502, Cityville, MI 21098'),
('Michael Wilson', 'michael.wilson@example.com', '1987-10-14', '777 Cedar Lane, Apt 205, Uptown, NJ 56789'),
('Lisa Johnson', 'lisa.johnson@example.com', '1984-05-08', '888 Elm Street, Suite 101, Nearbytown, CT 89012'),
('David Martin', 'david.martin@example.com', '1976-08-03', '999 Oak Lane, Apt 403, Suburbia, NV 34567'),
('Amy Clark', 'amy.clark@example.com', '1982-02-27', '123 Maple Road, Apt 205, Villageland, SC 21098'),
('Steven Davis', 'steven.davis@example.com', '1978-11-11', '456 Cedar Drive, Apt 502, Metropolis, VA 87654');

```

```

insert into products (product_name, product_price, product_stock, product_weight)
values

```

```

('Laptop', 899.99, 50, 2.5),
('Smartphone', 599.99, 100, 0.3),
('Headphones', 79.99, 200, 0.1),
('Camera', 499.99, 30, 1.2),
('Tablet', 349.99, 75, 0.8),
('Keyboard', 49.99, 200, 1.0),
('Mouse', 19.99, 300, 0.1),
('Monitor', 199.99, 60, 5.0),
('Printer', 129.99, 30, 7.0),
('Speakers', 69.99, 100, 2.5),
('External Hard Drive', 89.99, 80, 0.6),
('USB Flash Drive', 9.99, 500, 0.02),
('Gaming Console', 399.99, 35, 3.5),
('E-reader', 129.99, 90, 0.4),
('Fitness Tracker', 59.99, 120, 0.05),
('Digital Watch', 49.99, 200, 0.1),
('Microphone', 39.99, 150, 0.3),
('Router', 79.99, 70, 0.5),
('SD Card', 14.99, 400, 0.01),
('Webcam', 29.99, 100, 0.15),
('Smart TV', 599.99, 45, 10.0),
('Tablet Stand', 9.99, 250, 0.2),
('Charging Cable', 4.99, 800, 0.02),
('Screen Protector', 7.99, 300, 0.01),
('Laptop Bag', 24.99, 180, 0.8),
('Graphics Tablet', 179.99, 60, 1.5),
('Wireless Earbuds', 69.99, 120, 0.1),
('Game Controller', 39.99, 100, 0.3),
('Digital Camera', 249.99, 50, 1.8),
('USB Hub', 12.99, 200, 0.1);

```

```

insert into orders (order_date, customer_ID_FK)
values

```

```

('2015-06-04', 1),
('2015-03-12', 15),
('2015-08-20', 7),
('2016-07-16', 22),
('2018-09-03', 8),
('2017-05-21', 18),
('2016-10-16', 6),
('2019-12-30', 12),
('2015-12-10', 25),
('2015-05-07', 9),
('2019-01-17', 14),
('2016-03-05', 2),
('2019-09-13', 27),
('2019-05-28', 5),
('2017-11-05', 4),
('2017-01-29', 10),
('2018-03-26', 26),
('2016-04-16', 11),
('2017-09-18', 16),

```

('2015-04-30', 3),
 ('2019-10-30', 13),
 ('2015-02-12', 19),
 ('2016-02-06', 23),
 ('2015-05-14', 28),
 ('2019-10-21', 29),
 ('2018-03-19', 24),
 ('2016-08-12', 17),
 ('2016-02-13', 21),
 ('2019-01-24', 30),
 ('2019-04-09', 20),
 ('2015-06-04', 1),
 ('2017-06-18', 15),
 ('2018-10-01', 7),
 ('2018-08-13', 22),
 ('2015-08-28', 8),
 ('2015-08-06', 18),
 ('2019-03-03', 6),
 ('2018-08-14', 12),
 ('2016-09-18', 25),
 ('2016-08-22', 9),
 ('2018-03-07', 14),
 ('2016-11-17', 2),
 ('2015-06-04', 27),
 ('2016-03-26', 5),
 ('2015-08-18', 4),
 ('2018-01-08', 10),
 ('2018-08-01', 26),
 ('2016-03-10', 11);

insert into order_items (order_ID_FK, product_ID_FK, order_item_quantity, order_item_discount)
 values

(1, 15, 7, 15),
 (1, 7, 3, 25),
 (2, 22, 8, 5),
 (2, 11, 5, 35),
 (3, 2, 4, 10),
 (3, 30, 9, 18),
 (4, 8, 7, 28),
 (4, 17, 2, 2),
 (5, 6, 5, 33),
 (5, 18, 6, 22),
 (6, 4, 3, 12),
 (6, 20, 8, 30),
 (7, 12, 9, 7),
 (7, 28, 5, 32),
 (8, 3, 2, 20),
 (8, 27, 4, 19),
 (9, 9, 6, 26),
 (9, 19, 4, 8),
 (10, 23, 5, 11),
 (10, 30, 2, 17),
 (11, 14, 3, 22),
 (11, 26, 6, 16),
 (12, 10, 5, 9),
 (12, 7, 8, 31),
 (13, 21, 4, 14),
 (13, 29, 7, 13),
 (14, 1, 2, 21),
 (14, 26, 5, 1),
 (15, 12, 8, 24),
 (15, 24, 7, 29),
 (16, 16, 6, 38),
 (16, 20, 2, 4),
 (17, 25, 9, 23),
 (17, 30, 6, 27),

(18, 7, 7, 6),
(18, 18, 3, 34),
(19, 8, 6, 37),
(19, 15, 8, 39),
(20, 19, 7, 3),
(20, 28, 2, 36),
(21, 25, 5, 0),
(21, 4, 3, 40),
(22, 6, 6, 30),
(22, 27, 7, 7),
(23, 17, 4, 8),
(23, 11, 6, 25),
(24, 14, 5, 11),
(24, 16, 6, 25),
(25, 8, 7, 20),
(25, 13, 2, 10),
(26, 29, 4, 30),
(26, 5, 6, 28),
(27, 21, 8, 33),
(27, 14, 3, 17),
(28, 13, 9, 19),
(28, 25, 3, 21),
(29, 26, 6, 10),
(29, 2, 8, 14),
(30, 4, 7, 12),
(30, 5, 5, 35),
(31, 23, 6, 8),
(31, 19, 4, 15),
(32, 8, 3, 9),
(32, 30, 5, 32),
(33, 7, 8, 25),
(33, 17, 7, 16),
(34, 1, 4, 22),
(34, 5, 9, 18),
(35, 20, 5, 30),
(35, 4, 6, 12),
(36, 15, 7, 33),
(36, 6, 8, 21),
(37, 13, 2, 13),
(37, 25, 3, 19),
(38, 6, 5, 26),
(38, 10, 9, 14),
(39, 12, 4, 35),
(39, 15, 3, 2),
(40, 11, 6, 11),
(40, 18, 7, 9),
(41, 22, 2, 25),
(41, 29, 4, 1),
(42, 21, 3, 30),
(42, 4, 9, 6),
(43, 25, 5, 22),
(43, 9, 8, 15),
(44, 1, 4, 12),
(44, 16, 6, 37),
(45, 7, 7, 16),
(45, 22, 8, 20),
(46, 28, 2, 8),
(46, 9, 5, 28),
(47, 23, 4, 7),
(47, 20, 9, 9);

-- hozzáadunk egy telefonszám oszlopot a customer tablahoz

alter table customers

add column phone int;

-- összekötjük az order_item tablat a product tablával, az ETK szerint. Azert külön, mert a feladat kért egy változtatást

```
alter table order_items
add foreign key (product_ID_FK) references products(product_ID) on delete set null;
```

```
-- megnyitjuk a tablakat, hogy lassuk minden rendben
-- select * from customers;
-- select * from orders;
-- select * from products;
-- select * from order_items;
```

/*Lekerdezések, 3. hét*/

-- egytablasok (1db):

```
-- 1. lekerdezes: 18 ev feletti customerok, ha esetleg van korhataros termek
select customer_name, customer_birthday
from customers
where customer_birthday<=date_sub(curdate(), interval 18 year) -- current date minusz 18 ev
order by customer_birthday;
```

-- tobbtablasok (5db):

```
-- 2. lekerdezes: melyik order kihez, mikor, milyen emaillel
select order_ID, customer_name, order_date, customer_email
from orders, customers
where orders.customer_ID_FK=customers.customer_ID
order by 1;
```

-- 3. lekerdezes: mikbol mennyi rendelve, milyen discount rajtuk, bevetel az arukbol egyenkent

```
select product_name, concat(order_item_discount, '%') as discount, sum(order_item_quantity) as total_quantity,
sum(product_price*order_item_quantity*(100-order_item_discount) / 100) as income$
from products, order_items, orders
where orders.order_ID=order_items.order_ID_FK
and order_items.product_ID_FK=products.product_ID
group by product_name, order_item_discount
order by product_name, order_item_discount desc;
```

-- 4. lekerdezes: mennyit fizettek a vasarlok osszesen

```
select customer_name, round(SUM(product_price * order_item_quantity * (100 - order_item_discount)/ 100),2) as total_spent
from customers, orders, order_items, products
where customers.customer_ID = orders.customer_ID_FK
and orders.order_ID = order_items.order_ID_FK
and order_items.product_ID_FK = products.product_ID
group by customers.customer_name
order by total_spent desc;
```

-- 5. lekerdezes: mennyi marad a stock-ban

```
select products.product_name, products.product_stock-coalesce(sum(order_items.order_item_quantity), 0) as remaining_stock
from products, order_items
where products.product_ID = order_items.product_ID_FK
group by products.product_name, products.product_stock
order by products.product_name;
```

-- 6. lekerdezes: rendelések tomege

```
select order_ID, round(sum(product_weight*order_item_quantity), 2) as order_weight
from order_items, products, orders
where products.product_ID=order_items.product_ID_FK
and orders.order_ID=order_items.order_ID_FK
group by orders.order_ID
order by 1;
```

/*Outer join, 4. hét*/

-- #1 Ki rendelt tobb mint 5 eve, es mikor?

```
select customer_name,order_date from customers c
left join orders o on o.customer_ID_FK=c.customer_ID
having order_date<=date_sub(curdate(), interval 5 year)
order by order_date desc;
```



```

-- #2 A vasarlok melyik napokon rendeltek
select customer_name, order_date from customers c
left join orders o on c.customer_ID=o.customer_ID_FK
order by order_date desc;

-- #3 Melyik termekbol, hany darabot rendeltek, egy-egy rendeles soran
select product_name, order_item_quantity from products p
left join order_items oi on p.product_ID=oi.product_ID_FK
order by product_name asc;

/*Subselect lekerdezes, 4. het*/

-- #1 Termekek, amelyek dragabbak, mint a termekek atlagos ara
select product_name, product_price,(select avg(product_price) from products) as atlag, product_price-(select
avg(product_price) from products) as kulonbseg from products p
where product_price > (select avg(product_price) from products)
order by kulonbseg desc;

-- #2 A legdragabb termekbol egy adott rendelesben hanyat adtak el
select product_name, order_item_quantity from products p
join order_items oi on p.product_ID=oi.order_ID_FK
where product_price= (select max(product_price) from products);

-- #3 Melyik(ek)bol adtak el a legtobbet, es mikor?
select product_name,order_date, order_item_quantity from orders o
join order_items oi on o.order_ID=oi.order_ID_FK
join products p on p.product_ID=oi.order_ID_FK
where order_item_quantity=(select max(order_item_quantity) from order_items);

/*Nezettablas lekerdezes, indexes lekerdezes, 5. het*/

-- Mennyi marad a stockban lekerdezes view-va konvertalva
create view maradek_stock as
select products.product_name,
products.product_stock-coalesce(sum(order_items.order_item_quantity), 0) as remaining_stock
from products, order_items
where products.product_ID = order_items.product_ID_FK
group by products.product_name, products.product_stock
order by products.product_name;

-- Termekek, amikbol 30 alatt van keszleten
select * from maradek_stock
where maradek_stock.remaining_stock<30
order by maradek_stock.remaining_stock;

-- Email indexet keszitunk
create index email_index on customers(customer_email);

-- Email alapjan vasarlo által rendelt termeket listazunk
select order_ID, order_date, p.product_ID, p.product_name, p.product_price, c.customer_name from orders
join customers c on customer_ID=customer_ID_FK
join order_items oi on order_ID_FK=order_ID
join products p on product_ID=product_ID_FK
where customer_email="amanda.miller@example.com";

/*Tranzakcio 6. het*/

-- Tranzakcio kezdete, 2015. nyaran karbantartas volt, igy akik ott rendeltek csak 2015.09.01-en érkezett be a rendelesük
begin;
update orders
set order_date='2015-09-01'
where order_date between '2015-06-01' and '2015-08-31';

-- Adunk +5% kedvezmenyt azoknak, akik Laptopot rendelnek
update order_items
set order_item_discount=order_item_discount+5

```

```
where product_ID_FK=
(select product_ID
from products
where product_name='Laptop');
```

```
-- Növeljük az árat 10%-kal azoknak a termékeknek amikből már több, mint 20-at vásároltak
update products p
join (select product_ID_FK
from order_items
group by product_ID_FK
having sum(order_item_quantity) > 20) as feltetel
on p.product_ID = feltetel.product_ID_FK
set p.product_price = p.product_price * 1.1;
```

```
-- Ha nincs hiba a tranzakció vége, ha hiba van rollback-el visszavonjuk az update-eket
commit;
```

```
/*Fuggvény 7. hét*/
```

```
-- töröl egy rendelést --
drop function if exists deleteOrder;
DELIMITER //
create function deleteOrder(porderId int)
returns int
begin
    delete from orders where order_ID = porderId;
    delete from order_items where order_ID_FK=porderId;
    return 1;
end //
DELIMITER ;
```

```
-- kicseréli egy orderben a vásárlót
drop function if exists idcsere;
delimiter //
create function idcsere(porderId int, pNewCustID int)
returns int
begin
    declare valtozott int;
    update orders
    set customer_ID_FK = pNewCustID where order_ID = porderId;
    set valtozott= row_count();
    return valtozott;
end //
delimiter ;

select idcsere(4,6);
```