1. select v.USERNEV,v.NEV,v.EMAIL

from Vendeg v join Foglalas f on v.USERNEV=f.UGYFEL\_FK

group by v.USERNEV,v.NEV,v.EMAIL

HAVING COUNT(\*) > 1

select v.\*

1. from Vendeg v join Foglalas f on v.USERNEV=f.UGYFEL\_FK

    join Szoba sz on sz.SZOBA\_ID=f.SZOBA\_FK

where FEROHELY = (

    select MAX(FEROHELY)

    from szoba

) and KLIMAS = 'i'

1. select RANK() over (partition by COUNT(f.FOGLALAS\_PK) order by COUNT(f.FOGLALAS\_PK)),v.USERNEV, v.NEV, COUNT(f.FOGLALAS\_PK) as 'Foglalások száma'

from Vendeg v join Foglalas f on v.USERNEV=f.UGYFEL\_FK

group by v.USERNEV, v.NEV

select sz.SZOBA\_ID, COUNT(\*) over (PARTITION by sz.szoba\_id) as 'foglalások szobánként',

  f.mettol,

  COUNT(\*) over (PARTITION by day(f.mettol)) as 'Foglalások naponta',

    COUNT(\*) over (partition by sz.szoba\_id, f.mettol) as 'Foglalások szobánként, naponta'

from Szoba sz join Foglalas f on sz.szoba\_id=f.SZOBA\_FK

group by sz.SZOBA\_ID, f.mettol

1. CREATE TABLE #MASKED\_GUEST

(

    [USERNEV] [nvarchar](20) NOT NULL,

    [NEV] [nvarchar](50) MASKED WITH (FUNCTION = 'partial(2, "XXXX", 3)'),

    [EMAIL] [nvarchar](60) NOT NULL,

    [SZAML\_CIM] [nvarchar](100) NULL,

    [SZUL\_DAT] [date] NULL,

 CONSTRAINT [PK\_Vendeg] PRIMARY KEY CLUSTERED

)

1. SET konyvtar "Könyvtár neve"

HSET konyv:1 cim "Könyv címe" iro "Könyv Írója" kiadas\_ev "2000"

SADD mufajok "Műfaj1" "Műfaj2" "Műfaj3"

RPUSH kolcsonzott "Könyv1" "Könyv2" "Könyv3"

ZADD kolcson\_szam 5 "Könyv1" 3 "Könyv2" 4 "Könyv3"

CREATE TABLE DIAK (

ID int PRIMARY KEY,

Nev text,

Teljesitett\_felevek list<int>,

Eredmenyek map<int, float>,

Nyelvtudas set<text>

);

INSERT INTO DIAK (ID, Nev, Teljesitett\_felevek, Eredmenyek, Nyelvtudas)

VALUES (1, 'Anna', [1, 2, 3], {1: 3.00, 2: 4.55, 3: 4.10}, {'Angol', 'Francia'});

INSERT INTO DIAK (ID, Nev, Teljesitett\_felevek, Eredmenyek, Nyelvtudas)

VALUES (2, 'Pista', [1, 2, 3, 4], {1: 3.50, 2: 4.00, 3: 3.75, 4: 3.80}, {'Nemet', 'Spanyol', 'Angol'});

match (p:Person {name: 'Tom Hanks'}) - [:ACTED\_IN] -> (m:Movie) <- [:ACTED\_IN] - (pe:Person)

where pe.name <> 'Tom Hanks'

return distinct pe.name