

Jegyzőkönyv
Adatbázis rendszerek I.
Féléves feladat

Készítette: **Kaló Dániel**
Neptunkód: **BBKHXF**
Gyak: **Csütörtök** 8-10
Vezér: *Bednarik László*

A feladat leírása:

A feladatban egy E-R adatbázist hozok létre, az alábbi jellemzőkkel:

Autócentrum felépítését fogom lemodellezni. Hét egyeddel fog rendelkezni, melyek a következők: autó, megrendelő, kereskedés, dolgozó, beszállító, számla, rendelés.

Az autó egyed rendelkezik egy összetett „motor” tulajdonsággal, melyhez teljesítmény mindenképpen tartozik, valamint opcionálisan „hengerűrtartalom” nem elektromos autó esetén. Van többértékű „extrák” és „hajtás” tulajdonság, hiszen mind hajtásból, mind extrákból többféle lehet, ezek mellett „szín”, „ár”, „évjárat”, egyedi autó azonosító. Tartalmaz még egy kor származtatott tulajdonságot.

Van maga a kereskedés, mely rendelkezik összefoglalóan az autók jellemzőivel, a kereskedés azonosítójával és hogy tartalmazza-e jelenleg az autót vagy sem.

A következő a megrendelő, akinek van egy összetett típusó „címe”, mely az „irányítószámból”, a „településből”, az „utcából” és a „házzszámból” épül fel. Tartozik még hozzá egy „név”, valamint egy egyedi azonosító.

A dolgozó táblához tartozik egy többértékű beosztás tulajdonság, ugyanis egy dolgozó több területen is dolgozhat felváltva. E mellett van a fizetés, valamint egy összetett cím tulajdonság,

Minden vásárláshoz tartozik egy számla, mely az „összeget”, a „dátumot” és magát a számla azonosítót foglalja magában.

A készlet fenntartása érdekében rendelni is kell autókat, melyhez a Rendelés és a Beszállító elem tartozik.

A rendeléshez tartozik egy egyedi azonosító, egy „dátum”, valamint egy „darabszám”.

A beszállítónak szintén van egyedi azonosítója, egy „márkanéve”, valamint egy összetett „cím” tulajdonsága, mely az „irányítószámból”, a „településből”, az „utcából” és a „házzszámból” épül fel.

Az autó és a megrendelő között fennáll egy 1:N típusú kapcsolat, mivel az autóhoz csak 1 megrendelő tartozik, ezért az autó oldaláról egyes, viszont egy megrendelő több autót is vásárolhat, ezért erről az oldalról többszörös. Magához a vásárlás kapcsolathoz tartozik a fizetési mód és a darabszám.

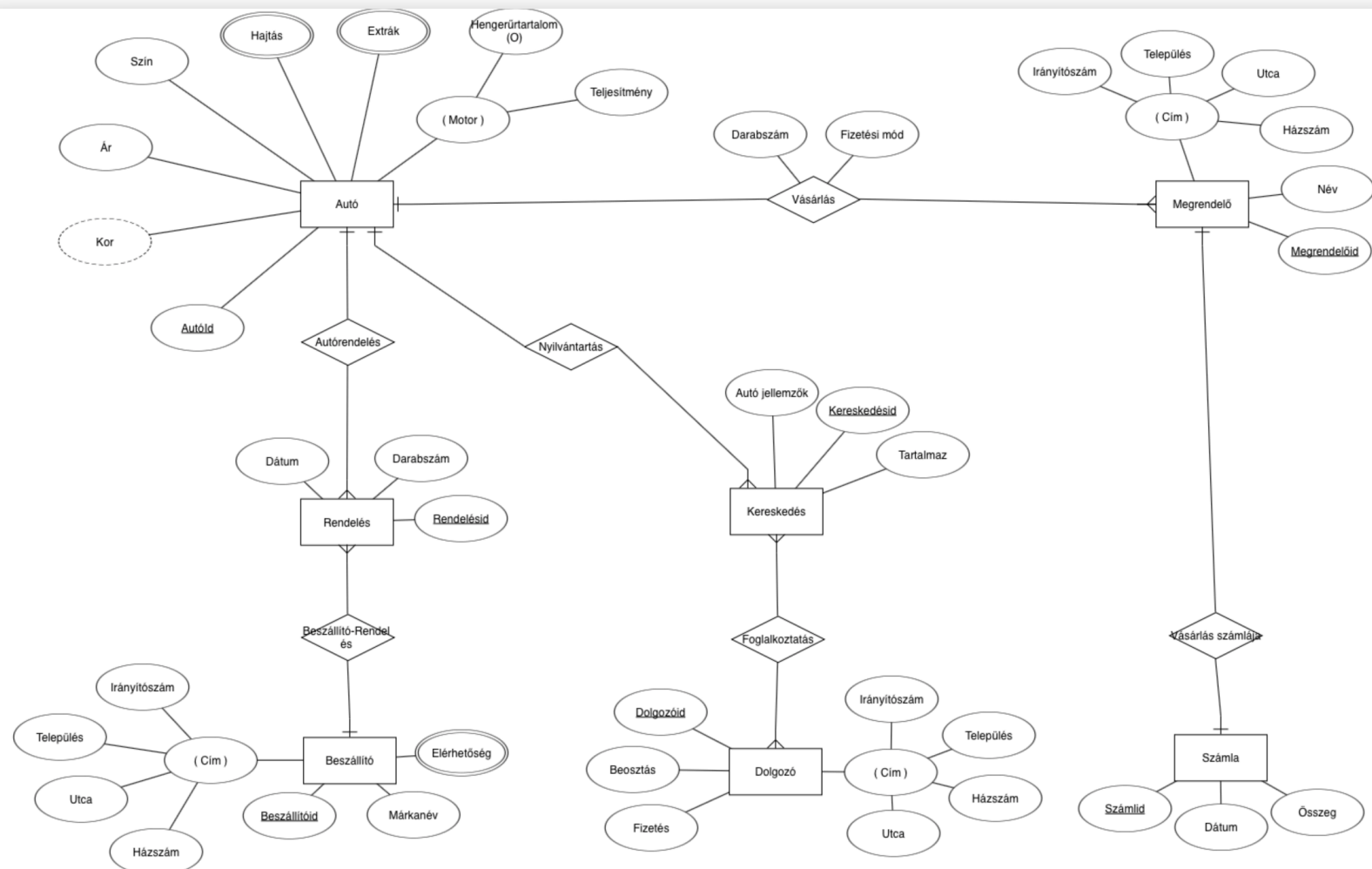
A dolgozó és a kereskedés között található egy N:M kapcsolat, mivel egy kereskedésben több dolgozó is van, valamint 1 dolgozó több kereskedésben is dolgozhat.

A megrendelő és a számla között található 1:1-es kapcsolat, mivel egy megrendeléshez 1 számla tartozik.

Az autó és a kereskedés között van a nyilvántartás kapcsolat, az adott autó a kereskedésben csak egyszer található, viszont egyféle autóból több is lehet a kereskedésben, ezért ez egy 1:N típusú kapcsolat, valamint a kereskedésben minimum 1 autó mindig van.

Van még kapcsolat az autó és a rendelés között, méghozzá egy 1:N kapcsolat, egy autótípust többször is rendelhetnek.

Fennáll egy 1:N kapcsolat a beszállító és a rendelés között, mivel egy adott rendelést csak egy beszállító tud kiszolgáltatni az adott márkából.



Az adatbázis relációs modelljének leírása:

A megrendelő táblából kiesik a cím, azonban a címhez tartozó 4 tulajdonság (irányítószám, település, utca, házszám) megmarad, valamint a nevet, a számlaid-t és az autóid-t fogja tartalmazni (idegen kulcs) az 1:N kapcsolat miatt. Ezek mellett elsődleges kulcsként jelen van benne a megrendelő azonosítója. A dolgozó tábla az azonosítóból, a címet felépítő tulajdonságokból, valamint a kereskedésid-ből és a beosztásból fog állni.

A dolgozó és a kereskedés N:M kapcsolata miatt egy külön tábla is létrejön, Foglalkoztatás néven, mely a Kereskedésid-t és Dolgozóid-t tartalmazza.

Az autó tábla tartalmazni fogja az azonosítóját, mely az elsődleges kulcs, a hengerűrtartalmat, a teljesítményt, a színt, az évjáratot, valamint egy rendelésid-t (idegen kulcs, 1:N kapcsolat miatt), az extrák és a hajtás külön táblába kerül, melyek az autóid-vel kapcsolódnak az autó táblához.

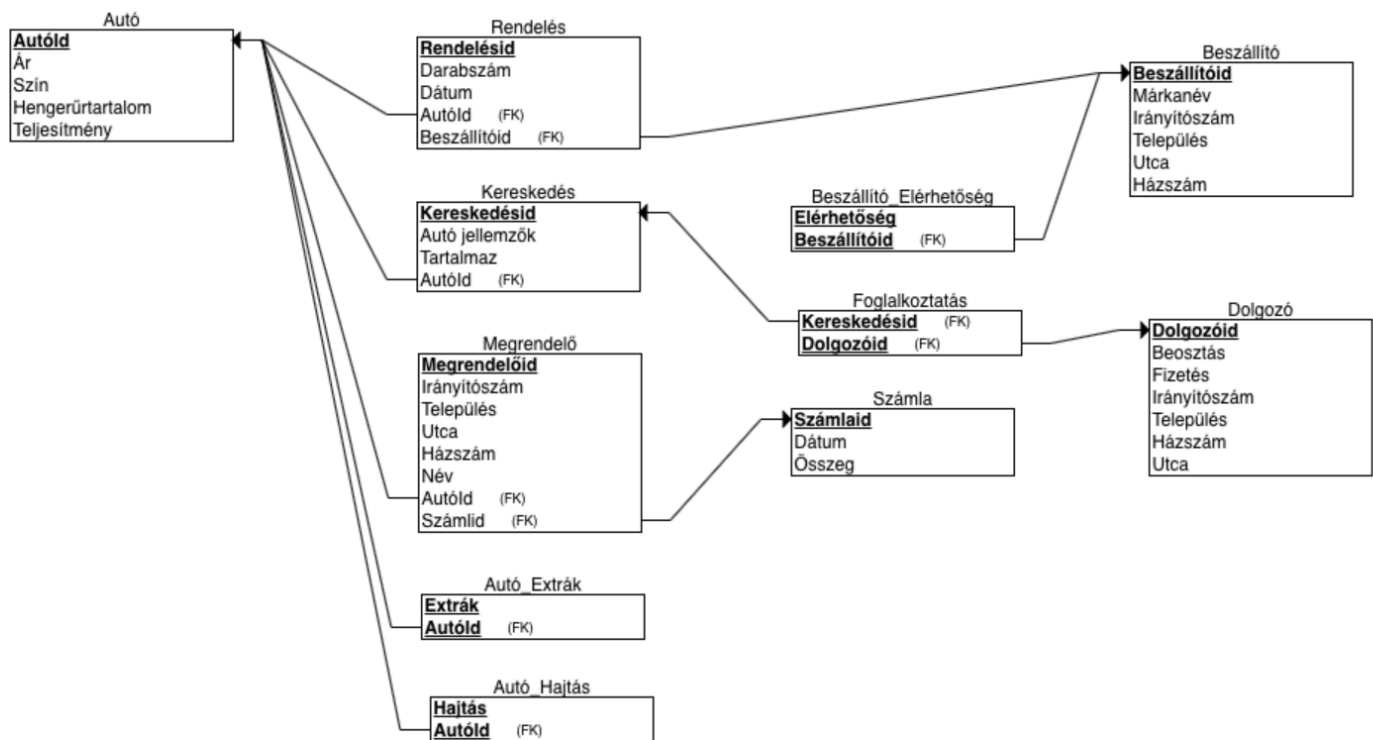
A kereskedés táblába kerül a kereskedésid (elsődleges kulcs), a tartalmaz és az autó jellemzők tulajdonság, az autóid-n keresztül kapcsolódik az autó táblához a fennálló 1:N kapcsolat miatt (1 oldalról kötelező).

A rendelés tábla tartalmazza a rendelés azonosítóját (elsődleges kulcs), a megrendelt darabszámot, a dátumot, mely automatikusan a megrendelés rögzítésekor fennálló rendszeridő, valamint a beszállítóid, melyen keresztül kapcsolódik a beszállító táblához.

A beszállító tábla tulajdonságai közül eltűnik a cím mező, ugyanis ez az irányítószám, település, utca és házszám tulajdonságokból tevődik össze. Tartalmazza még ezen felül a márkanevet és a tábla egyedi azonosítóját (elsődleges kulcs), az elérhetőség egy külön táblába kerül és a beszállító azonosítójával kapcsolódik az eredeti elemhez.

A számla tábla tartalmazza az 1:1 kapcsolatból adódó megrendelőid-t, a dátumot, mely a vásárlás dátuma, valamint a számla saját azonosítóját.

Az adatbázis relációs modellje:



Az adatbázis relációs sémái:

Megrendelő[Megrendelőid, Autóid, Irányítószám, Település, Utca, Házsám, Név]

Autó[Autóid, Rendelésid, Hengerűrtartalom, Teljesítmény, Szín, Ár, Típus]

AutóExtrák[Autóid, Extrák]

AutóHajtás[Autóid, Hajtás]

Kereskedés[Kereskedésid, Autóid, Tartalmaz]

KereskedésAutóJellemzők[Kereskedésid, AutóJellemzők]

Beszárllító[Beszárllítóid, Márkanév, Házsám, Település, Irányítószám, Utca]

BeszárllítóElérhetőség[Beszárllítóid, Elérhetőség]

Rendelés[Rendelésid, Beszárllítóid, Darabszám, Dátum]

Számla[Megrendelőid, Számlaid, Dátum]

Dolgozó[Dolgozóid, Kereskedésid, Beosztásid,

Fizetés, Irányítószám, Település, Utca, Házsám] Foglalkoztatás[Kereskedésid,

Dolgozóid]

DolgozóBeosztás[Beosztásid, Beosztás]

(Megjegyzés: a jegyzőkönyvet apple pages-ben szerkesztem, ahol nem találok szaggatott aláhúzást, ezért dőlttel jelöltem az idegen kulcsokat)

A táblák létrehozása:

A létrehozásnál ügyelni kell a sorrendre, először azokat a táblákat kell létrehozni, amelyekben nincs idegen kulcs, és ezután azokat, amelyekben van, hiszen az idegen

kulcsnak a már létrehozott táblára kell mutatnia. Az idegen kulcsot tartalmazó mezők

típusának meg kell egyeznie a referenciaként szolgáló, másik táblában található kulcsmező

típusával, ezért a Számla tábla MegrendelőId-je és a Megrendelő tábla MegrendelőId-je is

INT típusú. A Dolgozó tábla KereskedésId-je és Beosztása is INT, megegyezik a többi tábla ugyanazon Id-jével. A Megrendelő AutóId-je és az Autó t. azonosítója is szintén INT típusú,

hasonlóan a Kereskedés tábla AutóId-hez, valamint az Autó t. és a Rendelés t. RendelésID- hez, melyek szintén INT típusúak. A Rendelés t. BeszállítóId-je és a Beszállító t. azonosítója is egyezik, azaz INT típusú. Mivel figyelni kell a létrehozáskor arra, hogy milyen sorrendben

hozzuk létre a táblát ezért én az alábbi módon fogom létrehozni a táblákat: Autó, Beszállító, Kereskedés, Számla, Rendelés, Dolgozó, Foglalkoztatás, Autó_Hajtás, Autó_Extrák, Beszállító_Elérhetőség, Kereskedés_Autó_Jellemzők, Dolgozó_Beosztás, Megrendelő megakadályozva azt hogy olyan kulcsra lenne szüksége amely még nem létezik.

```
CREATE TABLE Autó( Hengerűrtartalom NUMERIC(5), Teljestmény  
NUMERIC(4) NOT NULL, Szín VARCHAR(15) NOT NULL, Ár NUMERIC(10)  
NOT NULL, AutóId INT NOT NULL, Típus VARCHAR(40) NOT NULL, PRIMARY  
KEY (AutóId));
```

```
CREATE TABLE Beszállító( Márkanév VARCHAR(10) NOT NULL, BeszállítóId  
INT NOT NULL, Házszám NUMERIC(5) NOT NULL, Település VARCHAR(15)  
NOT NULL, Irányítószám NUMERIC(5) CHECK(Irányítószám>0), Utca  
VARCHAR(20) NOT NULL, PRIMARY KEY (BeszállítóId));
```

```
CREATE TABLE Kereskedés( KereskedésId INT NOT NULL, Tartalmaz  
NUMERIC(1) NOT NULL, AutóId INT NOT NULL, PRIMARY KEY  
(KereskedésId), FOREIGN KEY (AutóId) REFERENCES Autó(AutóId));
```

```
CREATE TABLE Számla( Összeg NUMERIC(12) NOT NULL, Dátum DATE NOT  
NULL, SzámlaId INT NOT NULL, PRIMARY KEY (SzámlaId));
```

```
CREATE TABLE Rendelés( Darabszám NUMERIC(3) NOT NULL, RendelésId  
INT NOT NULL, Dátum DATE NOT NULL, BeszállítóId INT NOT NULL, AutóId  
INT NOT NULL, PRIMARY KEY (RendelésId), FOREIGN KEY (BeszállítóId)
```

REFERENCES Beszállító(Beszállítóld), FOREIGN KEY (Autóld) REFERENCES Autó(Autóld));

CREATE TABLE Dolgozó(Házszám NUMERIC(4) NOT NULL, Irányítószám NUMERIC(5) NOT NULL, Utca VARCHAR(20) NOT NULL, Település VARCHAR(15) NOT NULL, Fizetés NUMERIC(7) NOT NULL, Dolgozóld INT NOT NULL, Beosztásld INT NOT NULL, PRIMARY KEY (Dolgozóld), FOREIGN KEY (Beosztásld) REFERENCES Dolgozó_Beosztás(Beosztásld));

CREATE TABLE Foglalkoztatás(Kereskedésld INT NOT NULL, Dolgozóld INT NOT NULL, PRIMARY KEY (Kereskedésld), FOREIGN KEY (Kereskedésld) REFERENCES Kereskedés(Kereskedésld), FOREIGN KEY (Dolgozóld) REFERENCES Dolgozó(Dolgozóld));

CREATE TABLE Autó_Hajtás(Hajtás VARCHAR(100) NOT NULL, Autóld INT NOT NULL, PRIMARY KEY (Autóld), FOREIGN KEY (Autóld) REFERENCES Autó(Autóld));

CREATE TABLE Autó_Extrák(Extrák VARCHAR(100) NOT NULL, Autóld INT NOT NULL, PRIMARY KEY (Autóld), FOREIGN KEY (Autóld) REFERENCES Autó(Autóld));

CREATE TABLE Beszállító_Elérhetőség(Elérhetőség VARCHAR(100) NOT NULL, Beszállítóld INT NOT NULL, PRIMARY KEY (Beszállítóld), FOREIGN KEY (Beszállítóld) REFERENCES Beszállító(Beszállítóld));

CREATE TABLE Kereskedés_Autó_jellemzők(Autó_jellemzők VARCHAR(100) NOT NULL, Kereskedésld INT NOT NULL, PRIMARY KEY (Kereskedésld), FOREIGN KEY (Kereskedésld) REFERENCES Kereskedés(Kereskedésld));

CREATE TABLE Dolgozó_Beosztás(Beosztásld INT NOT NULL, Beosztás VARCHAR(20) NOT NULL, PRIMARY KEY (Beosztásld));

CREATE TABLE Megrendelő(Irányítószám NUMERIC(6) CHECK(Irányítószám>0), Település VARCHAR(20) NOT NULL, utca VARCHAR(50) NOT NULL, Házszám NUMERIC(4) NOT NULL, Név VARCHAR(20) NOT NULL, Megrendelőld INT NOT NULL, Autóld INT NOT NULL, Számlald INT NOT NULL, PRIMARY KEY (Megrendelőld), FOREIGN KEY (Autóld) REFERENCES Autó(Autóld), FOREIGN KEY (Számlald) REFERENCES Számla(Számlald));

Adatszerkezetet érintő módosító parancsok:

ALTER TABLE Autó modify Hengerűrtartalom char(50) NOT NULL;

ALTER TABLE Dolgozó MODIFY Házszám char(10) NOT NULL;

ALTER TABLE Dolgozó MODIFY Irányítószám char(10) NOT NULL;

ALTER TABLE Dolgozó ADD Név VARCHAR(10);

A táblák feltöltése:

Autó:

```
insert into Autó values(1200,400,'Piros',1000000,01,'SkodaY3GTL');
```

```
insert into Autó values(1500,420,'Kék',1100000,02,'SkodaY5GHL');
```

```
insert into Autó values(1000,100,'Sárga',1204000,03,'AudiXY');
```

```
insert into Autó values(3000,150,'Zöld',1530700,04,'BMWl6');
```

```
insert into Autó values(6000,130,'Szürke',1707040,05,'Lada2.1');
```

AutóExtrák:

```
Insert into Autó_Extrák values ('Ülésfűtés',1);
```

```
Insert into Autó_Extrák values ('Kormányállítás',2);
```

```
Insert into Autó_Extrák values ('Talatókamera',3);
```

```
Insert into Autó_Extrák values ('Dönthető ülés',4);
```

```
Insert into Autó_Extrák values ('HarmanKardonHangrendszer',5);
```

AutóHajtás

```
Insert into Autó_Hajtás values('4kerék',1);
```

```
Insert into Autó_Hajtás values('2kerék',2);
```

```
Insert into Autó_Hajtás values('2kerék',3);
```

```
Insert into Autó_Hajtás values('4kerék',4);
```

```
Insert into Autó_Hajtás values('állítható',5);
```

Megrendelő:

```
insert into Megrendelő values  
(3400,'Mezőkövesd','Mátyáskirály',77,'KisBéla',1,1,1);
```

```
insert into Megrendelő values (3521,'Miskolc','Aba',12,'NagyJános',2,2,2);
```

```
insert into Megrendelő values (3256,'Eger','Jósika',14,'HorváthÁdám',3,3,3);
```

```
insert into Megrendelő values  
(3600,'Nyíregyháza','KlapkaGyörgy',44,'KovácsJózsef',4,4,4);
```

```
insert into Megrendelő values (3721,'Győr','DózsaGyörgy',32,'LehelMária',5,5,5);
```


Számla:

Insert into Számla values(6845,'2011-10-14',1);

Insert into Számla values(68885,'2015-5-23',2);

Insert into Számla values(68789,'2019-3-11',3);

Insert into Számla values(6878,'2018-8-20',4);

Insert into Számla values(68585,'2017-12-4',5);

Beszállító:

Insert into Beszállító values ('Skoda','1','10','Prága','2351','Opletalova');

Insert into Beszállító values ('VW','2','17','Lichtenstein','5123','Gafeli');

Insert into Beszállító values ('BMW','3','15','Frankfurt','4511','GinnHeim');

Insert into Beszállító values ('Toyota','4','12','LoShuan','4521','Kasuo');

Insert into Beszállító values ('Mitsubishi','5','14','Kirikato','4671','Sekikawa');

BeszállítóElérhetőség:

Insert into Beszállító_Elérhetőség values ('pelda@example.hu',1);

Insert into Beszállító_Elérhetőség values ('+45 34 564443',2);

Insert into Beszállító_Elérhetőség values ('pelda@example-ext.hu',3);

Insert into Beszállító_Elérhetőség values ('pelda232@exam.hu',4);

Insert into Beszállító_Elérhetőség values ('+345 24 55645433',5);

Kereskedés:

Insert into Kereskedés values(1,0,1);

Insert into Kereskedés values(2,0,2);

Insert into Kereskedés values(3,0,3);

Insert into Kereskedés values(4,0,4);

Insert into Kereskedés values(5,0,5);

KereskedésAutóJellemzők:

Insert into Kereskedés_Autó_jellemzők values('jó állapotú',1);

Insert into Kereskedés_Autó_jellemzők values('nem jó állapotú',2);

Insert into Kereskedés_Autó_jellemzők values('jó állapotú',3);

Insert into Kereskedés_Autó_jellemzők values('nem jó állapotú',4);

Insert into Kereskedés_Autó_jellemzők values('jó állapotú',5);

Dolgozó:

Insert into Dolgozó values('12',2548,'Fütyülő','Badacsonytomaj',100000,1,1,'Kiss Béla');

Insert into Dolgozó values('45',2518,'Patakpart','Tapolca',250000,2,2,'Nagy Árpí');

Insert into Dolgozó values('32',2528,'Fasor','Salgótarján',1000000,3,3,'OK Péter');

Insert into Dolgozó values('21',2545,'Telep','Pécs',1000014,4,4,'Örs Balázs');

Insert into Dolgozó values('88',2541,'Kétút','Szeged',1000014,5,5,'Karakó Szilvia');

DolgozóBeosztás:

Insert into Dolgozó_Beosztás values(1,'Marketing manager');

Insert into Dolgozó_Beosztás values(2,'Takarító');

Insert into Dolgozó_Beosztás values(3,'Takarító');

Insert into Dolgozó_Beosztás values(4,'Eladó');

Insert into Dolgozó_Beosztás values(5,'Sofőr');

Foglalkoztatás:

insert into Foglalkoztatás values(1,1);

insert into Foglalkoztatás values(2,2);

insert into Foglalkoztatás values(3,3);

insert into Foglalkoztatás values(4,4);

insert into Foglalkoztatás values(5,5);

Rendelés:

Insert into Rendelés values(123,1,'2015-7-7',1,1);

Insert into Rendelés values(642,2,'2015-8-7',2,2);

Insert into Rendelés values(714,3,'2015-9-7',3,3);

Insert into Rendelés values(801,4,'2015-10-7',4,4);

Insert into Rendelés values(264,5,'2015-7-1',5,5);

Adattartalom módosítása:

Módosítsa a Skoda YHGT árát 3580000-re.

UPDATE Autó set Ár= 3580000 where Típus=' Skoda YHGT ';

Módosítsa a termékek árát 300000-el abban az esetben, ha az ára kisebb mint 3500000.

UPDATE Autó set Ár=+ 300000 where Ár< 3500000;

Törölje a dolgozót, akinek a fizetése több 400000-nél.

Delete from Dolgozó where Fizetés>400000;

Törölje a számlát, ahol a dátum 2019.04.05.

Delete from Számla where Dátum='05/04/2019';

Módosítsa a '286985' dolgozó fizetését 435000-re.

UPDATE Dolgozó set Fizetés=435000 where DolgozóId= '286985';

Lekérdezések:

Az összes adat lekérdezése a Dolgozók táblából:

Select * from Dolgozó;

$\Pi(\text{dolgozó})$

Az autók típusa és ára:

Select Típus, Ár from Autó;

$\Pi_{\text{típus, ár}}(\text{AUTO})$

Az autók színének lekérdezése, ismétlődés nélkül:

Select distinct Szín from Autó;

$\Pi_{\text{szín}}(\text{AUTO})$

Azon dolgozók Települése, ahol a fizetés meghaladja a 100000-et:

Select Település from Dolgozó WHERE Fizetés > 100000;

$\Pi_{\text{település}}(\sigma_{\text{fizetés} > 100000}(\text{Dolgozó}))$

Azon autók Típusának lekérdezése, ahol az ár 1000000 és 1200000 között van: Select Típus from Autó WHERE Ár BETWEEN 1000000 AND 1200000;

$\Pi_{a.típus} (\sigma_{200000 < ár < 300000} (Autó))$

A "Péter" keresztnévű emberek fizetése:

Select Fizetés from Dolgozó WHERE Név LIKE '% Péter%';

$\Pi_{fizetes} (\sigma_{Név = 'Péter'} (Dolgozó))$

Az autók átlagára:

Select avg(Ár) from Autó;

$\Gamma_{típus}^{típus, avg(*)} (Auto)$

Hány olyan dolgozó van, ahol a fizetés nagyobb 400000-nél:

Select COUNT(Fizetés) from Dolgozó WHERE Fizetés >400000;

$\Gamma_{dolgozo}^{count(dolgozó)} (\sigma_{fizetés > 400000} (Dolgozó))$

Olyan autótípus, ahol az átlagár meghaladja a 4000000 Ft-ot:

Select Típus, avg(Ár) from Autó GROUP BY Típus HAVING avg(Ár) > 4000000;

$\Pi_{típus} (\sigma_{avg(ar) > 400000} (Autó))$

Az autók típusának min és max ára, típusra csoportosítva:

Select Típus, min(Ár), max(Ár) from Autó GROUP BY Típus;

$\Pi_{min(ar), max(ar)} (Autó > < Típus)$

A takarító beosztású dolgozók nevei:

SELECT Dolgozó.Név FROM Dolgozó_Beosztás INNER JOIN Dolgozó ON
Dolgozó.Beosztásid = Dolgozó_Beosztás.Beosztásid WHERE
Dolgozó_beosztás.beosztás = 'Takarító';

Autók típusa és ára, ár szerint csökkenő sorrendben:

Select Típus, Ár from Autó ORDER BY Ár DESC;

$\Pi_{ár, típus} (Autó)$

A Miskolci emberek neve és autójuk típusa:

SELECT Megrendelő.Név, Autó.Típus FROM Megrendelő INNER JOIN Autó ON
Megrendelő.autóid = Autó.Autóid WHERE Megrendelő.Település = 'Miskolc';

$\Pi_{t.név, a.típus} (\sigma_{lakhely='Tapolca'} (Megrendelő))$

Azok a rendelések dátumai, ahol a beszállító Frankfurti:

```
SELECT Rendelés.Dátum FROM Rendelés INNER JOIN Beszállító ON  
Rendelés.Beszállítóid = Beszállító.Beszállítóid WHERE Beszállító.Település = 'Frankfurt';
```

$\Pi_{\text{dátum}} (\sigma_{\text{Beszállító}='Frankfurt'}(\text{Rendelés}))$

Megrendelők autójának darabszáma:

```
Select név,count(megrendelőid) from megrendelő group by megrendelőid;
```

$\Gamma_{\text{Megrendelő}} \text{count(auto)}$

Milyen színű autójuk van egyes embereknek:

```
Select Megrendelő.név,Autó.szín from megrendelő inner join autó on megrendelő.autóid =  
autó.autóid;
```

$\Pi_{\text{szín}} (\sigma_{\text{MegrendelőID}='X'}(\text{Megrendelő}))$