Курсов проект по „Бази данни" на

Софтуерно инженерство

летен семестър 2021/2022г.

Автосервис

Участници:

Васил Андреев - 62597

Калин Борисов - 62580

Михаил Зонев - 62595

Атанас Спиртов - 62588

Калоян Йонов - 62581

Развитието на автоиндустрията води до създаването на нови марки автомобили. Това води до нарастването на информацията, която всяко ново превозно средство носи. Чрез създаването на база данни която съдържа тази информация можем да помогнем на сервизите като решим някои ключови проблеми:

-да се следи всяка част от кое моторно превозно средство е -

-дали си съответстват две части една с друга.

Задача	Фак.Ном.
1. Създаване на схемите и релациите	100%-62588
2. Добавяне на съдържание	100%-62581
3. Примерни прости заявки и заявки върху две и повече релации	50%-62588 50%-62595
4. Примери с подзаявки	50%-62595 50%-62580
5. Примери със съединения	100%-62580
6. Примери с групиране и аграгация	100%-62595
7. Примери с ограничения	100%-62580
8. Примери с изгледи и индекси	50%-62595 50%-62588
9. Примери с тригери	100%-62597
10. Подготовка на презентацията	50%-62581 50%-62597

1.EER Диаграма:
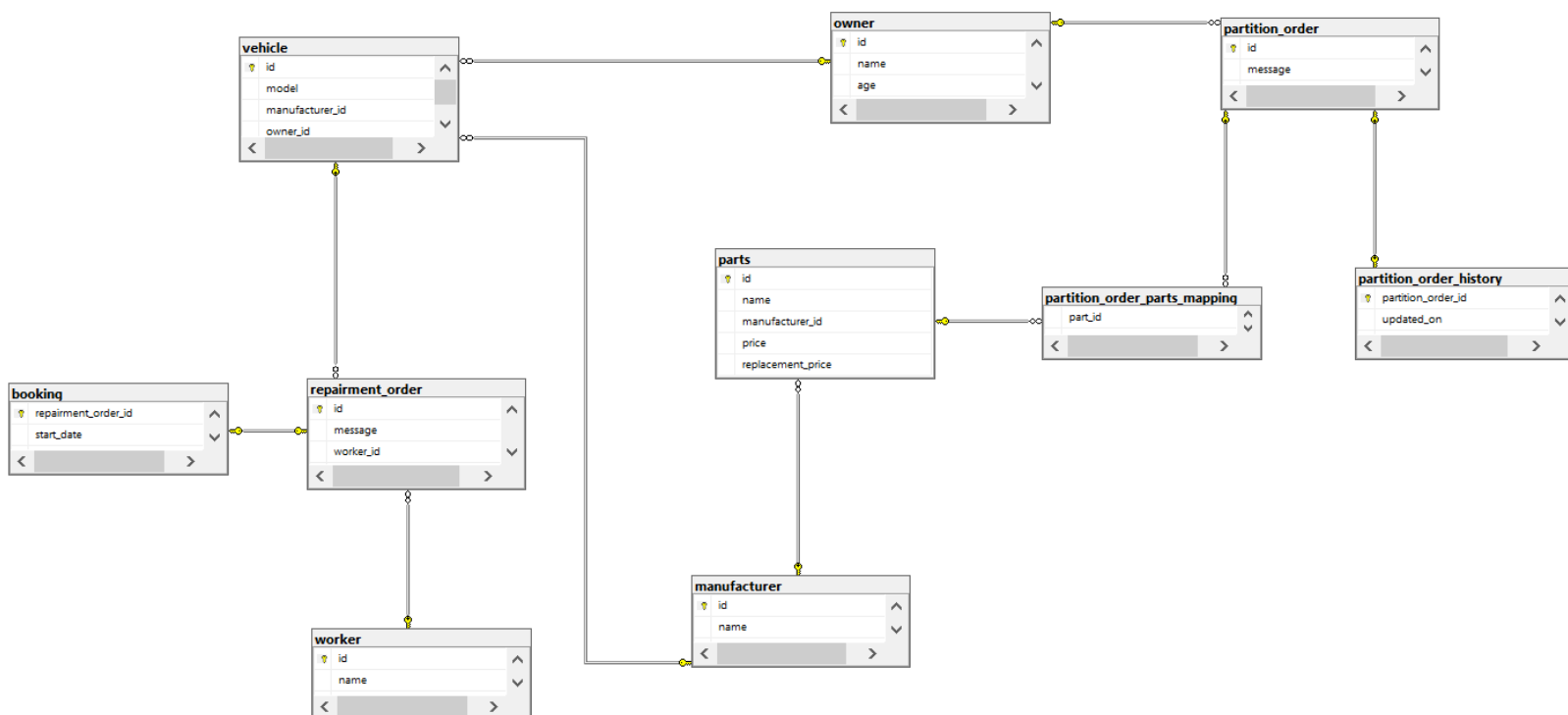
2.База:

```sql
/*TABLE DROPPER*/
DROP TABLE IF EXISTS booking;
DROP TABLE IF EXISTS partition_order_history;
DROP TABLE IF EXISTS partition_order_parts_mapping;
DROP TABLE IF EXISTS partition_order;
DROP TABLE IF EXISTS parts;
DROP TABLE IF EXISTS repairment_order;
DROP TABLE IF EXISTS vehicle;
DROP TABLE IF EXISTS owner;
DROP TABLE IF EXISTS manufacturer;
DROP TABLE IF EXISTS worker;
```

```sql
/*MANUFACTURER*/

CREATE TABLE manufacturer (
  id int NOT NULL,
  name varchar(45) NOT NULL,
  manufacturer_rating tinyint DEFAULT NULL,
  PRIMARY KEY (id),
  CHECK (manufacturer_rating >=0 AND
manufacturer_rating<=10)
)

INSERT INTO manufacturer (id, name, manufacturer_rating)
VALUES
(1,'Dodge',4),(2,'BMW',9),(3,'Mercury',3),(4,'Bentley',8),
(5,'GMC',2),(6,'Dodge',5),(7,'Kia',10),(8,'Volkswagen',9),
(9,'Mitsubishi',3),(10,'Dodge',7),(11,'Audi',8),(12,'Mazda
',5),(13,'Audi',3),(14,'Oldsmobile',4),(15,'Volvo',6),(16,
'Volkswagen',8),(17,'Buick',8),(18,'Suzuki',3),(19,'Mercur
y',9),(20,'Saab',8),(21,'Toyota',2),(22,'Nissan',1),(23,'C
hevrolet',3),(24,'Toyota',1),(25,'Smart',7),(26,'Pontiac',
6),(27,'Chevrolet',7),(28,'Lexus',4),(29,'Lotus',9),(30,'L
incoln',9);
/*WORKER*/

CREATE TABLE worker (
  id int NOT NULL,
  name varchar(45) NOT NULL,
  salary int NOT NULL,
  PRIMARY KEY (id),
  CHECK (salary >= 650)
)
INSERT INTO worker (id, name, salary) VALUES (1,'Marian
Ognyanov', 750),(2,'Dimitur Valentinov', 1050),(3,'Simeon
Damyanski', 1300),(4, 'Atanas Simeonov',1250),(5,'Valentin
Ivanov',690),(6,'Martin Petrov',1550),(7,'Vladimir
Ivanov',1200),(8,'Atanas Koychev',1400),(9, 'Stefan
Dimitrov',920),(10,'Martin Kirilov',800),(11,'Kalin
Kamenov',1050),(12,'Dimitur Yonov', 1240),(13,'Milen
```

```sql
Dimitrov',2100),(14,'Ivan Shumenov',1320),(15,'Mihail
Stanchev',1455),(16,'Ivan Aleksandrov',1410),(17,'Cvetomir
Kirilov',1600),(18,'Denis Adrianov',750),(19,'Anton
Aleksandrov',1800),(20,'Nikolai
Dimitrov',2100),(21,'Ivelin Vasilev',1300),(22,'Kamen
Krustev',1200),(23,'Mihaela Staneva',950),(24,'Radostin
Danchev',1650),(25,'Velizar Milchev',1950),(26,'Victoria
Traykova',800),(27,'Lubomir Mihaylov',1100),(28,'Denislav
Mladenov',900),(29,'Daniel Makov',700),(30,'Stanislav
Arikov',1360);
/*OWNER*/

CREATE TABLE owner (
  id int NOT NULL IDENTITY,
  name varchar(45) NOT NULL,
  age int NOT NULL,
  gender varchar(45) DEFAULT NULL,
  PRIMARY KEY (id),
  CHECK (age >= 18)
)

SET IDENTITY_INSERT owner ON
INSERT INTO owner (id,name, age, gender) VALUES
(1,'Marjorie Glanville',67,'Female'),(2,'Anthe
Hearfield',74,'Female'),(3,'Eolanda
Teresse',18,'Female'),(4,'Dilly
Chafer',67,'Male'),(5,'Chrystal
Scoines',62,'Female'),(6,'Aluino
MacLise',63,'Male'),(7,'Arleen
Blankman',64,'Female'),(8,'Cherish
Symcock',29,'Female'),(9,'Winthrop
Taverner',69,'Male'),(10,'Fredra
Morffew',18,'Female'),(11,'Corbin
Teaser',18,'Male'),(12,'Barbabas
Rubi',89,'Male'),(13,'Shay
Wynrehame',18,'Female'),(14,'Kim
Garretson',18,'Male'),(15,'Mariann
Onion',29,'Female'),(16,'Cornie
Rosini',47,'Female'),(17,'Felizio
```

```sql
Prover',18,'Male'),(18,'Dorthea
Gard',18,'Female'),(19,'Nev
Leebeter',50,'Male'),(20,'Cesya
Hazeup',18,'Female'),(21,'Sandra
Alleyn',18,'Female'),(22,'Bert
Lamy',98,'Male'),(23,'Gizela
Kruszelnicki',88,'Female'),(24,'Lurette
Kaser',18,'Female'),(25,'Tobie
Locks',69,'Male'),(26,'Mirabelle
Ratlee',19,'Female'),(27,'Cecilia
Trebble',46,'Female'),(28,'Halette
Teresia',89,'Female'),(29,'Twila
Schwand',47,'Female'),(30,'Thedrick Overill',38,'Male');
SET IDENTITY_INSERT owner OFF

/* VEHICLE */

CREATE TABLE vehicle (
  id int NOT NULL IDENTITY,
  model varchar(45) NOT NULL,
  manufacturer_id int NOT NULL,
  owner_id int DEFAULT NULL,
  PRIMARY KEY (id),
  INDEX ownerId_idx (owner_id),
  INDEX manufacturer_id_idx (manufacturer_id),
  INDEX manufacturerId_idx (manufacturer_id),
  CONSTRAINT manufacturerId FOREIGN KEY (manufacturer_id)
REFERENCES manufacturer (id),
  CONSTRAINT ownerId FOREIGN KEY (owner_id) REFERENCES
owner (id)
)

SET IDENTITY_INSERT vehicle ON
INSERT INTO vehicle (id, manufacturer_id, model ,owner_id)
VALUES (1,19, 'Sable' ,10),(2,14, 'Bravada' ,22),(3,27,
'Traverse'
,9),(4,14,'Cutlass',14),(5,7,'Sorento',14),(6,10,'Durango'
,22),(7,14,'Aurora',18),(8,21,'Sienna',3),(9,24,'Supra',26
),(10,29,'Esprit',19),(11,28,'UX',18),(12,19,'Milan',19),(
```

```sql
13,17,'Lucerne',12),(14,13,'A6',7),(15,5,'Envoy',22),(16,1
0,'Dakota',16),(17,1,'Magnum',2),(18,15,'C30',27),(19,14,'
Intrigue',26),(20,18,'Ciaz',29),(21,18,'Ertiga',13),(22,7,
'Carens',9),(23,26,'G8',12),(24,11,'Q3',20),(25,28,'CT',22
),(26,10,'Intrepid',2),(27,16,'Passat',20),(28,14,'442',29
),(29,30,'Nautilus',23),(30,3,'Cougar',2);

SET IDENTITY_INSERT vehicle OFF
/*PARTS*/

CREATE TABLE parts (
  id int NOT NULL,
  name varchar(45) DEFAULT NULL,
  manufacturer_id int NOT NULL,
  price int NOT NULL,
  replacement_price int NOT NULL,
  PRIMARY KEY (id),
  INDEX manufacturer_id_idx (manufacturer_id),
  CONSTRAINT manufacturer_id FOREIGN KEY (manufacturer_id)
REFERENCES manufacturer (id),
)

INSERT INTO parts (id, name, manufacturer_id, price,
replacement_price) VALUES
(1,'Mirrors',11,379,542),(2,'Lights',17,249,551),(3,'Hoods
',23,304,583),(4,'Computer',18,381,522),(5,'GPS',9,224,569
),(6,'Tires',17,201,567),(7,'Door',2,355,570),(8,'Radiator
',18,259,587),(9,'Airbag',8,341,556),(10,'Engine',30,444,5
75);


/*REPAIRMENT ORDER*/

CREATE TABLE repairment_order (
  id int NOT NULL IDENTITY,
  message varchar(45) NOT NULL,
  worker_id int NOT NULL,
  vehicle_id int NOT NULL,
  PRIMARY KEY (id),
```

```sql
    INDEX workerId_idx (worker_id),
    INDEX vehicleIdss_idx (vehicle_id),
    CONSTRAINT vehicleIdss FOREIGN KEY (vehicle_id)
REFERENCES vehicle (id),
    CONSTRAINT workerId FOREIGN KEY (worker_id) REFERENCES
worker (id)
)
SET IDENTITY_INSERT repairment_order ON
INSERT INTO repairment_order (id, message, worker_id,
vehicle_id) VALUES (1,'Repair Sweetheart',13,21),(2,'Woman
100 crash repair',19,1),(3,'Money Order',1,27),(4,'My
order',3,16),(5,'Dude
Car',7,8),(6,'Mitsu',24,8),(7,'Latlux',26,19),(8,'Alphazap
',27,28),(9,'Alpha',10,10),(10,'Ventosanzap',29,15);
SET IDENTITY_INSERT repairment_order OFF
/*BOOKING*/

CREATE TABLE booking (
    repairment_order_id int NOT NULL,
    start_date datetime NOT NULL,
    end_date datetime NOT NULL,
    PRIMARY KEY (repairment_order_id),
    INDEX reparimentOrderId_idx (repairment_order_id),
    CONSTRAINT reparimentOrderId FOREIGN KEY
(repairment_order_id) REFERENCES repairment_order (id)
)

INSERT INTO booking
(repairment_order_id,start_date,end_date) VALUES (1,'2022-
06-12 23:46:49','2022-08-23 21:02:09'),(2,'2022-06-12
17:40:45','2022-01-20 13:01:37'),(3,'2022-06-29
20:32:46','2022-06-09 01:42:25'),(5,'2022-06-10
03:39:07','2021-10-04 04:05:53'),(4,'2022-05-27
19:55:37','2021-08-24 03:30:47'),(7,'2022-05-31
15:02:27','2022-01-16 19:19:21'),(6,'2022-06-26
03:51:02','2021-12-24 08:57:05'),(8,'2022-06-06
20:38:29','2022-07-23 12:53:30'),(9,'2022-06-05
13:54:59','2021-07-25 23:39:36'),(10,'2022-06-14
03:04:03','2021-12-05 17:11:42');
```

```sql
/*PARTITION ORDER*/

CREATE TABLE partition_order (
  id int NOT NULL IDENTITY,
  message varchar(45) DEFAULT NULL,
  owner_id int NOT NULL,
  PRIMARY KEY (id),
  INDEX ownerId_idx (owner_id),
  CONSTRAINT ownerIds FOREIGN KEY (owner_id) REFERENCES
owner (id)
)

SET IDENTITY_INSERT partition_order ON
INSERT INTO partition_order (id, message, owner_id) VALUES
(1,'Spirtow Order',26),(2,'Gift',28),(3,'For
Sam',30),(4,'For Nelly',27),(5,'Maria',8),(6,'Sons first
car upgrade',1),(7,'Kaboom',30),(8,'For my sweet
car',2),(9,'First Order for
enhancement',4),(10,'Pannier',3);
SET IDENTITY_INSERT partition_order OFF
/*PARTITION ORDER HISTORY*/

CREATE TABLE partition_order_history (
  partition_order_id int NOT NULL,
  updated_on datetime DEFAULT NULL,
  is_completed tinyint NOT NULL,
  PRIMARY KEY (partition_order_id),
  INDEX orderForPartsIds_idx (partition_order_id),
  CONSTRAINT partitionOrderId FOREIGN KEY
(partition_order_id) REFERENCES partition_order (id)
)

INSERT INTO partition_order_history (partition_order_id,
updated_on, is_completed) VALUES (10,'2022-05-10
04:05:40',0),(7,'2022-05-18 20:20:38',0),(8,'2022-05-14
17:41:58',0),(1,'2022-05-09 10:10:47',0),(5,'2022-05-27
11:35:06',1),(3,'2022-05-17 09:00:13',1),(9,'2022-05-08
```

```
13:04:11',0),(4,'2022-05-03 13:10:50',0),(2,'2022-05-23
00:50:54',1),(6,'2022-05-20 00:09:49',0);


/*PARTITION ORDER PARTS MAPPING*/

CREATE TABLE partition_order_parts_mapping (
  part_id int DEFAULT NULL,
  order_id int DEFAULT NULL,
  INDEX partIds_idx (part_id),
  INDEX orderIds_idx (order_id),
  CONSTRAINT orderIds FOREIGN KEY (order_id) REFERENCES
partition_order (id),
  CONSTRAINT partIds FOREIGN KEY (part_id) REFERENCES
parts (id)
)

INSERT INTO partition_order_parts_mapping (part_id,
order_id) VALUES
(9,5),(9,8),(5,6),(1,8),(1,2),(3,8),(1,10),(6,6),(8,3),(8,
2),(3,7),(6,3),(7,7),(3,4),(9,6),(8,1),(4,5),(4,10),(2,3),
(8,8),(1,8),(9,3),(7,8),(3,10),(3,9),(10,4),(4,6),(6,8),(4
,5),(4,10);
```

3.Прости заявки:

```
/*1.Имената и ранга на производителите с имена започшащи с
'B' подредени по рейтинг.*/
SELECT  M.NAME,
        M.manufacturer_rating
FROM    manufacturer M
WHERE   M.NAME LIKE 'B%'
ORDER   BY M.manufacturer_rating
```

	NAME	MANUFACTURER_RATING
1	Bentley	8
2	Buick	8
3	BMW	9

```sql
/*2.Всички части с цена за смяна по-голяма или равна на
400.*/
SELECT *
FROM    parts
WHERE   replacement_price >= 400;
```

	id	name	manufacturer_id	price	replacement_price
1	1	Mirrors	11	379	542
2	2	Lights	17	249	551
3	3	Hoods	23	304	583
4	4	Computer	18	381	522
5	5	GPS	9	224	569
6	6	Tires	17	201	567
7	7	Door	2	355	570
8	8	Radiator	18	259	587
9	9	Airbag	8	341	556
10	10	Engine	30	444	575

```sql
/*3.Имената на служителите със заплата по-голяма или равна
на 1500.*/
SELECT W.NAME
FROM    worker W
WHERE   W.salary >= 1500
```

	NAME
1	Martin Petrov
2	Milen Dimitrov
3	Cvetomir Kirilov
4	Anton Aleksandrov
5	Nikolai Dimitrov
6	Radostin Danchev
7	Velizar Milchev

```sql
/*4.Името и възрастта на всички собственици между 18 и 30
години.*/
SELECT NAME,
        age
FROM    owner
WHERE   age > 18
        AND age < 30;
```

	name	age
1	Cherish Symcock	29
2	Mariann Onion	29
3	Mirabelle Ratlee	19

```sql
/*5.Средната заплата на работниците.*/
SELECT Avg(salary) AS avgSalary
FROM    worker
```

	Avg Salary
1	1254

4.Заявки с две или повече релации:

```sql
/* 1.Името и годините на собствениците на превозни
средства 'Toyota'. */
select o.name, o.age
from owner o, vehicle v, manufacturer m
where o.id = v.owner_id
    and m.id = v.manufacturer_id
    and m.name = 'Toyota'
order by o.name
```

	name	age
1	Eolanda Teresse	18
2	Mirabelle Ratlee	19

```sql
/*2.Всички поръчки за 'airbag'.*/
select po.message
from  partition_order po, partition_order_parts_mapping
pm, parts p
where po.id = pm.order_id
    and pm.part_id = p.id
    and p.name = 'Airbag'
```

	message
1	Maria
2	For my sweet car
3	Sons first car upgrade
4	For Sam

```sql
/*3. Всички поръчки за поправка и работещите по тях за
коли с производител 'Toyota'.*/
select ro.message as repairment_order_message, w.name as
worker_name
from vehicle v, repairment_order ro, worker w,
manufacturer m
where v.id = ro.vehicle_id and ro.worker_id = w.id
    and v.manufacturer_id = m.id
    and m.name = 'Toyota'
```

	repairment_order_message	worker_name
1	Dude Car	Vladimir Ivanov
2	Mitsu	Radostin Danchev

```sql
/*4. Имената на собствениците на 'Dodge' и моделът, който
карат. */
```

```sql
select o.name, v.model
from owner o, vehicle v, manufacturer m
   where o.id = v.owner_id
           and v.manufacturer_id = m.id
           and m.name = 'Dodge'
order by o.name
```

	name	model
1	Anthe Hearfield	Magnum
2	Anthe Hearfield	Intrepid
3	Bert Lamy	Durango
4	Cornie Rosini	Dakota

```sql
/*5.Имената на работниците, които работят по поръчки за
коли с марка 'Oldsmobile'.*/
select w.name as worker, m.name as carBrand
from worker w, repairment_order ro, vehicle v,
manufacturer m
    where w.id = ro.id
            and ro.vehicle_id = v.id
            and v.manufacturer_id = m.id
            and m.name = 'Oldsmobile'
```

	worker	carBrand
1	Vladimir Ivanov	Oldsmobile
2	Atanas Koychev	Oldsmobile

5.Подзаявки:

```sql
/*1. Моделите на превозни средства на мъже под 50г.*/
SELECT DISTINCT ls.model
FROM vehicle ls, (SELECT * FROM OWNER WHERE age<50 AND
gender = 'male')ow
WHERE ls.owner_id = ow.id;
```

	model
1	Cutlass
2	Sorento

```sql
/*2. Моделите на превозни средства на жени над или на
35г.*/
select m.name as carBrand, v.model as model
from vehicle v, manufacturer m
    where    v.manufacturer_id = m.id
            and v.owner_id in (select o.id
                                    from owner o
```

```
                                        where o.gender = 'Female'
                                        and o.age >= 35)
```

	carBrand	model
1	Audi	A6
2	Dodge	Dakota
3	Dodge	Magnum
4	Volvo	C30
5	Suzuki	Ciaz
6	Dodge	Intrepid
7	Oldsmobile	442
8	Lincoln	Nautilus
9	Mercury	Cougar

```
/*3.Имената на работниците, които работят върху модел
'Sienna'*/
SELECT DISTINCT name
FROM worker
WHERE id IN (SELECT worker_id
             FROM repairment_order
             WHERE vehicle_id in
             (SELECT id from VEHICLE WHERE
model='Sienna'));
```

	name
1	Radostin Danchev
2	Vladimir Ivanov

```
/*4.Името и възрастта на най-възрастният собственик*/
select o.name as oldest_olner, o.age
from owner o
where o.age >= all(select owner.age from owner)
```

	oldest_olner	age
1	Bert Lamy	98

```
/*5.Името и цената на всички части по-скъпи от гумите.*/

select name as part_name, parts.price
from parts
where replacement_price >(select p.replacement_price
                          from parts p
                          where p.name = 'Tires')
order by price desc
```

	part_name	price
1	Engine	444
2	Door	355
3	Hoods	304
4	Radiator	259
5	GPS	224

6.Съединения:

```sql
/*1.Име и възраст на owner-и, които само поръчват части и
са под 60г.*/
select o.name, o.age
from owner o left join vehicle v on o.id = v.owner_id
where v.id is null
      and o.age < 60
```

	name	age
1	Cherish Symcock	29
2	Corbin Teaser	18
3	Mariann Onion	29
4	Felizio Prover	18
5	Sandra Alleyn	18
6	Lurette Kaser	18
7	Thedrick Overill	38

```sql
/*2.Име на производителя и модел на всички превозни
средства на 'Anthe Hearfield'*/
select m.name , v.model
from manufacturer m join vehicle v on m.id =
v.manufacturer_id join owner o on v.owner_id = o.id
where o.name = 'Anthe Hearfield'
```

	name	model
1	Dodge	Magnum
2	Dodge	Intrepid
3	Mercury	Cougar

```sql
/*3.Имената на всички работници които не работят по BMW*/
SELECT DISTINCT w.name
FROM WORKER w
JOIN repairment_order RO ON RO.worker_id=w.id
JOIN vehicle vh ON vh.id = RO.vehicle_id
```

```sql
WHERE vh.manufacturer_id NOT IN (SELECT id FROM
manufacturer WHERE name = 'BMW');
```

	name
1	Anton Aleksandrov
2	Daniel Makov
3	Lubomir Mihaylov
4	Marian Ognyanov
5	Martin Kirilov
6	Milen Dimitrov
7	Radostin Danchev
8	Simeon Damyanski
9	Victoria Traykova
10	Vladimir Ivanov

```sql
/*4.Имената на всички owner-и мъже, над 30г., чиито
поръчки за части са готови.*/
SELECT DISTINCT o.name
FROM OWNER o
JOIN partition_order po ON po.owner_id = o.id
JOIN partition_order_history poh ON po.id =
poh.partition_order_id
WHERE poh.is_completed = 1
AND o.age>30
AND o.gender = 'male';
```

	name
1	Thedrick Overill

```sql
/*5.Име на owner, модел на превозното средство,
съобщение на поръчката за поправка и старт/енд дата
подредени по старт дата за поръчки направени преди '2022-
06-10'*/
select o.name, v.model, ro.message, b.start_date,
b.end_date
from owner o join vehicle v on o.id = v.owner_id
          join repairment_order ro on v.id = ro.id
          join booking b on ro.id =
b.repairment_order_id
where b.start_date < '2022-06-10'
order by b.start_date
```

	name	model	message	start_date	end_date
1	Kim Garretson	Cutlass	My order	2022-05-27 19:55:37.000	2021-08-24 03:30:47.000
2	Dorthea Gard	Aurora	Latlux	2022-05-31 15:02:27.000	2022-01-16 19:19:21.000
3	Mirabelle Ratlee	Supra	Alpha	2022-06-05 13:54:59.000	2021-07-25 23:39:36.000
4	Eolanda Teresse	Sienna	Alphazap	2022-06-06 20:38:29.000	2022-07-23 12:53:30.000

```
/*6.Извежда последната версия на поръчките които са със
статус изпълнени*/
SELECT * FROM partition_order_history ph
join partition_order p on ph.partition_order_id = p.id
where ph.updated_on = (SELECT MAX(ph2.updated_on)
                        FROM partition_order_history ph2
                        GROUP BY ph2.partition_order_id
                        HAVING ph.partition_order_id =
ph2.partition_order_id)
    and ph.is_completed = 1
        order by ph.partition_order_id
```

	partition_order_id	updated_on	is_completed	id	message	owner_id
1	2	2022-05-23 00:50:54.000	1	2	Gift	28
2	3	2022-05-17 09:00:13.000	1	3	For Sam	30
3	5	2022-05-27 11:35:06.000	1	5	Maria	8

7.Групиране и аграгация:

```
/*1.Поръчки със стойност(сума от цената на всички части +
цената за смяната им) по-голяма от 3000.*/
select po.message, sum( p.price + p.replacement_price) as
summedPrice
from partition_order po  join
partition_order_parts_mapping m on po.id = m.order_id
                        join parts p on m.part_id =
p.id
group by po.message
having sum( p.price + p.replacement_price) >= 3000
```

	message	summedPrice
1	For my sweet car	6165
2	For Sam	3311
3	Pannier	3614
4	Sons first car upgrade	3361

```sql
/*2.Общата стойност на частите заедно със смяната за
поръчка с име 'Gift'*/
select po.message, sum( p.price + p.replacement_price) as
summedPrice
from partition_order po  join
partition_order_parts_mapping m on po.id = m.order_id join
parts p on m.part_id = p.id
group by po.message
having po.message = 'Gift'
```

	message	summedPrice
1	Gift	1767

```sql
/*3. Име на всеки owner и брой на притежаваните превозни
средства, подредени по броя.*/
select o.name, count(v.id) as numVehicles
from owner o left join vehicle v on o.id = v.owner_id
group by o.name
order by numVehicles desc
```

	name	numVehicles
1	Bert Lamy	4
2	Anthe Hearfield	3
3	Barbabas Rubi	2
4	Cesya Hazeup	2
5	Dorthea Gard	2
6	Kim Garretson	2
7	Mirabelle Ratlee	2
8	Nev Leebeter	2
9	Twila Schwand	2
10	Winthrop Taverner	2
11	Shay Wynrehame	1
12	Eolanda Teresse	1
13	Fredra Morffew	1
14	Gizela Kruszelni...	1
15	Cecilia Trebble	1
16	Cornie Rosini	1
17	Arleen Blankman	1
18	Aluino MacLise	0
19	Cherish Symcock	0
20	Chrystal Scoines	0
21	Corbin Teaser	0
22	Dilly Chafer	0
23	Halette Teresia	0
24	Felizio Prover	0
25	Thedrick Overill	0
26	Tobie Locks	0
27	Sandra Alleyn	0
28	Lurette Kaser	0
29	Mariann Onion	0
30	Marjorie Glanville	0

```sql
/*4.Съобщенията на поръчките и броя части за тях, ако той
е над 3.*/
select po.message, count(p.price) as numberOfParts
from partition_order po join partition_order_parts_mapping
m on po.id = m.order_id
                        join parts p on m.part_id = p.id
group by po.message
having count(p.price) > 3
```

	message	numberOfParts
1	For my sweet car	7
2	For Sam	4
3	Pannier	4
4	Sons first car upgrade	4

```sql
/*5. Брой модели от всяка марка.*/
select m.name, count(v.model) as numModels
from manufacturer m left join vehicle v on m.id =
v.manufacturer_id
group by m.name
order by numModels desc
```

	name	numModels
1	Oldsmobile	5
2	Dodge	4
3	Mercury	3
4	Audi	2
5	Kia	2
6	Lexus	2
7	Suzuki	2
8	Toyota	2
9	Volkswagen	1
10	Volvo	1
11	Lincoln	1
12	Lotus	1
13	Buick	1
14	Chevrolet	1
15	GMC	1
16	Pontiac	1
17	Saab	0
18	Smart	0
19	Mitsubishi	0
20	Nissan	0
21	Bentley	0
22	BMW	0
23	Mazda	0

```sql
/*6. Брой превозни средства разпределени по пол.*/
select o.gender, count(partition_order_id) as numOrders
from owner o join partition_order po on o.id = po.owner_id
            join partition_order_history h on po.id =
h.partition_order_id
group by o.gender
```

	gender	numOrders
1	Female	7
2	Male	3

```
/*7. Марки превозни средства и колко работници работят по
тях. */
select m.name, count(w.id) as numWorkers
from worker w join repairment_order ro on w.id =
ro.worker_id
              join vehicle v on ro.vehicle_id = v.id
              join manufacturer m on v.manufacturer_id =
m.id
group by m.name
order by numWorkers desc
```

	name	numWorkers
1	Oldsmobile	2
2	Toyota	2
3	Volkswagen	1
4	Suzuki	1
5	Dodge	1
6	GMC	1
7	Lotus	1
8	Mercury	1

```
/*8. Изчислява колко би струвала поправката на модел
'Sienna' на 'Eolanda Teresse' със съобщение 'Mitsu'.*/
select ro.message, sum(price + replacement_price) as
totalPrice
from repairment_order ro join vehicle v on ro.vehicle_id =
v.id
   join owner o on o.id = v.owner_id
   join partition_order po on o.id = po.owner_id
   join partition_order_parts_mapping m on po.id =
m.order_id
   join parts p on p.id = m.part_id
where o.name = 'Eolanda Teresse'
      and v.model = 'Sienna'
      and ro.message = 'Mitsu'
group by ro.message
```

	message	totalPrice
1	Mitsu	3614

```sql
/*9. Извежда броя производители за всеки ранк.*/
select manufacturer_rating as rating, count(m.name) as
numManufacturers
from manufacturer m
group by manufacturer_rating
order by manufacturer_rating desc
```

	rating	numManufacturers
1	10	1
2	9	5
3	8	5
4	7	3
5	6	2
6	5	2
7	4	3
8	3	5
9	2	2
10	1	2

```sql
/*10. Броя налични части по производители. */
select m.name, count(p.name) as numParts
from manufacturer m join parts p on m.id =
p.manufacturer_id
group by m.name
order by numParts desc
```

	name	numParts
1	Buick	2
2	Suzuki	2
3	Volkswagen	1
4	Audi	1
5	BMW	1
6	Chevrolet	1
7	Lincoln	1
8	Mitsubishi	1

8.Ограничения:
А)CHECK:

```sql
/*OWNER*/
CHECK (age >= 18)
/*WORKER*/
CHECK (salary >= 650)
/*MANUFACTURER*/
CHECK (manufacturer_rating >=0 AND manufacturer_rating<=10)
```

Б)FOREIGN KEY:

```
/*PARTITION ORDER PARTS MAPPING*/

CONSTRAINT orderIds FOREIGN KEY (order_id) REFERENCES partition_order (id),
CONSTRAINT partIds FOREIGN KEY (part_id) REFERENCES parts (id)

/*PARTITION ORDER HISTORY*/

CONSTRAINT partitionOrderId FOREIGN KEY (partition_order_id) REFERENCES partition_order
(id)

/*PARTITION ORDER*/

CONSTRAINT ownerIds FOREIGN KEY (owner_id) REFERENCES owner (id)

/*BOOKING*/

CONSTRAINT reparimentOrderId FOREIGN KEY (repairment_order_id) REFERENCES
repairment_order (id)

/*REPAIRMENT ORDER*/

CONSTRAINT vehicleIdss FOREIGN KEY (vehicle_id) REFERENCES vehicle (id),
CONSTRAINT workerId FOREIGN KEY (worker_id) REFERENCES worker (id)

/*PARTS*/

CONSTRAINT manufacturer_id FOREIGN KEY (manufacturer_id) REFERENCES manufacturer (id),

/* VEHICLE */

CONSTRAINT manufacturerId FOREIGN KEY (manufacturer_id) REFERENCES manufacturer (id),
CONSTRAINT ownerId FOREIGN KEY (owner_id) REFERENCES owner (id)
```

B)PRIMARY KEY:

```
/*MANUFACTURER*/

PRIMARY KEY (id),

/*WORKER*/

PRIMARY KEY (id),

/*OWNER*/

PRIMARY KEY (id),

/* VEHICLE */

PRIMARY KEY (id),

/*PARTS*/

PRIMARY KEY (id),

/*REPAIRMENT ORDER*/

PRIMARY KEY (id),
```

```
/*BOOKING*/

PRIMARY KEY (repairment_order_id),

/*PARTITION ORDER*/

PRIMARY KEY (id),

/*PARTITION ORDER HISTORY*/

PRIMARY KEY (partition_order_id),
```

9.Изгледи и индекси:

А)Изгледи:

```
CREATE VIEW V_PARTITION_ORDER_MAP
AS
SELECT P.PART_ID, P.NAME, P.TOTAL_PRICE, PO.ORDER_ID, PO.message
    FROM (SELECT ID AS PART_ID, NAME, price + replacement_price AS
TOTAL_PRICE FROM PARTS ) P,
        partition_order_parts_mapping M,
          (SELECT ID AS ORDER_ID, MESSAGE FROM partition_order) PO
    WHERE P.PART_ID = M.part_id AND M.order_id = PO.ORDER_ID

    GO
    CREATE VIEW V_CALCULATE_TOTAL_PRICE
    AS
    SELECT ORDER_ID, SUM(MAP.TOTAL_PRICE) AS TOTAL_PRICE
        FROM (SELECT ID AS ROREDER_ID, vehicle_id  FROM
    repairment_order) RO
            JOIN (SELECT ID AS VE_ID, MODEL, OWNER_ID FROM vehicle) V
    ON RO.vehicle_id = V.VE_ID
            JOIN (SELECT ID AS O_ID FROM OWNER) O ON V.owner_id =
    O.O_ID
            JOIN (SELECT ID AS PORDER_ID FROM partition_order) PO ON
    O.O_ID = PO.PORDER_ID
            JOIN V_PARTITION_ORDER_MAP MAP ON PO.PORDER_ID =
    MAP.ORDER_ID
    GROUP BY ORDER_ID
```

Б) Индекси:

```
    INDEX ownerId_idx (owner_id),
    INDEX manufacturerId_idx (manufacturer_id),
```

```sql
    INDEX workerId_idx (worker_id),
    INDEX vehicleIdss_idx (vehicle_id),

    INDEX reparimentOrderId_idx (repairment_order_id),

    INDEX orderForPartsIds_idx (partition_order_id),

    INDEX partIds_idx (part_id),
    INDEX orderIds_idx (order_id),
```

10. Тригери:

```sql
CREATE TRIGGER stop_manufacture_delete
ON manufacturer
INSTEAD OF DELETE
AS DECLARE @i bit
    GO

    CREATE TRIGGER UpdatedManufacturerName
    ON manufacturer
    AFTER UPDATE
    AS
    IF EXISTS (
     SELECT *
     FROM
            INSERTED I
          JOIN
          DELETED D
        ON D.ID = I.ID
            AND D.name <> I.name
     )
    Print 'Manufacturer name has changed'
    GO
```