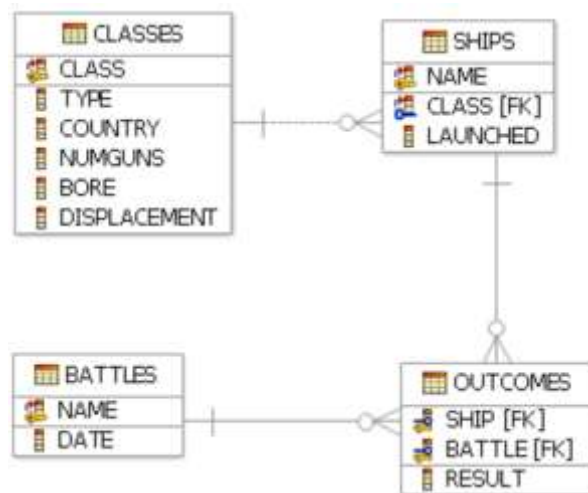


Задача 4. Дадена е базата от данни *Ships*, в която се съхранява информация за кораби и тяхното участие в битки по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба. Таблицата *Classes* съдържа информация за класовете кораби:

- **class** – име на клас, първичен ключ;
- **type** – тип ('bb' за бойни кораби, 'bc' за бойни крайцери);
- **country** – държава, която строи такива кораби;
- **numguns** – брой оръдия, може да приема стойност null;
- **bore** – калибър на оръдието (в инчове), може да приема стойност null;
- **displacement** – водоизместимост (в тонове), може да приема стойност null.



Таблицата *Ships* съдържа информация за корабите:

- **name** – име на кораб, първичен ключ;
- **class** – име на клас, външен ключ към *Classes.class*;
- **launched** – година, в която корабът е пуснат на вода, може да приема стойност null.

Таблицата *Battles* съхранява информация за битките:

- **name** – име на битка, първичен ключ;
- **date** – дата на провеждане.

Таблицата *Outcomes* съдържа информация за резултата от участието на даден кораб в дадена битка. Атрибутите *ship* и *battle* заедно формират първичния ключ.

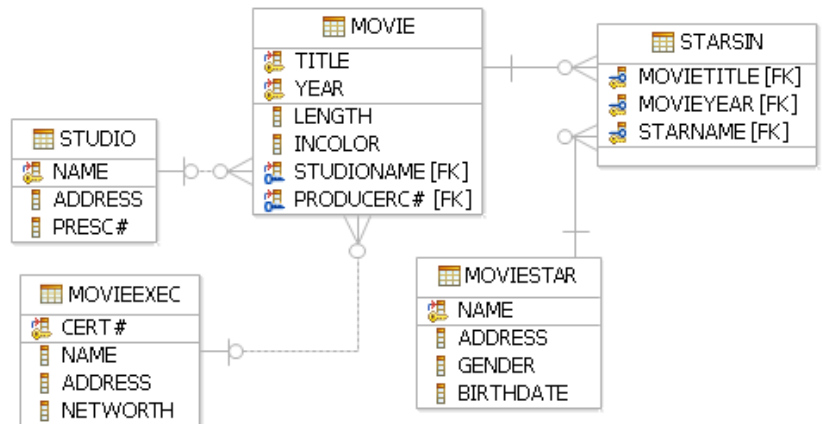
- **ship** – име на кораб, външен ключ към *Ships.name*;
- **battle** – име на битка, външен ключ към *Battles.name*;
- **result** – резултат (потънал – 'sunk', повреден – 'damaged', победил – 'ok').

Забележка за всички таблици: За всички атрибути, за които не е указано, че могат да приемат стойност null, да се счита, че съществува not null ограничение.

1. Да се напише заявка, която извежда име на клас, годината на първата битка, в която кораб на този клас е участвал, годината на последната битка, в която кораб на този клас е участвал, и броя на всички различни битки, в които кораби на този клас са участвали, само за тези класове, започващи с буквата N. Ако за даден клас няма кораб, който да е участвал в битка, за съответните години да се върне стойност null.
2. Да се напише заявка, която да изведе имената на тези битки, за които броят на корабите от тип 'bb', участвали в тази битка, е по-голям от броя на корабите от тип 'bc', участвали в същата битка. Битки, в които не е участвал нито един кораб, да не се извеждат в резултата.

Задача 4. Дадена е базата от данни **Movies**, в която се съхранява информация за филми, филмови студиа, които ги произвеждат, продуцентите на филмите, както и актьорите, които участват в тях. Таблицата **Movie** съдържа информация за филми. Атрибутите **title** и **year** заедно формират първичния ключ.

- **title** — заглавие;
- **year** — година, в която е заснет филмът;
- **length** — дължина в минути;
- **incolor** — 'Y' за цветен филм и 'N' за чернобял;
- **studioName** — име на студио, външен ключ към **Studio.name**;
- **producerc#** — номер на сертификат на продуцента, външен ключ към **MovieExec.cert#**.



Таблицата **MovieStar** съдържа информация за филмови звезди:

- **name** — име, първичен ключ;
- **address** — адрес;
- **gender** — пол, 'M' за мъж (актьор) и 'F' за жена (актриса);
- **birthdate** — рождена дата.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ.

Атрибутите **movietitle** и **movieyear** образуват външен ключ към **Movie**.

- **movietitle** — заглавие на филма;
- **movieyear** — година на заснемане на филма;
- **starname** — име на филмовата звезда, външен ключ към **MovieStar.name**.

Таблицата **Studio** съдържа информация за филмови студиа:

- **name** — име, първичен ключ;
- **address** — адрес;
- **presc#** — номер на сертификат на президента на студиото.

Таблицата **MovieExec** съдържа информация за продуцентите на филми.

- **cert#** — номер на сертификат, първичен ключ;
- **name** — име;
- **address** — адрес;
- **networth** — нетни активи;

Забележка за всички таблици: Всички атрибути, които не участват във формирането на първичен ключ, могат да приемат стойност **NULL**.

1. Да се напише заявка, която да изведе име на студио, годината на първия филм за това студио, годината на последния филм за това студио и броя на всички филми за това студио, само за тези студиа започващи с буквата 'M'.
2. Да се напише заявка, която да изведе името на актрисата, участвала в най-много филми, и броя на филмите, в които е участвала.

Задача 3. (35 min)

Дадена е базата от данни Ships, в която се съхранява информация за кораби и тяхното участие в битки по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба.

Таблицата **Classes** съдържа информация за класовете кораби:

class – име на клас, първичен ключ;
 type – тип ('bb' за бойни кораби, 'bc' за бойни крайцери);
 country – държава, която строи такива кораби;
 numguns – брой оръдия, може да приема null стойност;
 bore – калибър на оръдието (в инчове), може да приема null стойност;
 displacement – водоизместимост (в тонове), може да приема null стойност.

Таблицата **Ships** съдържа информация за корабите:

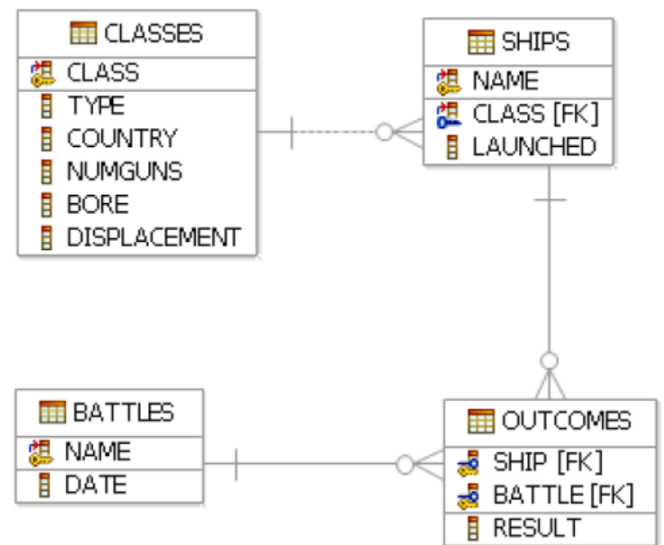
name – име на кораб, първичен ключ;
 class – име на клас, външен ключ към Classes.class;
 launched – година, в която корабът е пуснат на вода, може да приема null стойност.

Таблицата **Battles** съхранява информация за битките:

name – име на битка, първичен ключ;
 date – дата на провеждане.

Таблицата **Outcomes** съдържа информация за резултата от участието на даден кораб в дадена битка. Атрибутите ship и battle заедно формират първичния ключ.

ship – име на кораб, външен ключ към Ships.name;
 battle – име на битка, външен ключ към Battles.name;
 result – резултат (потънал – 'sunk', повреден – 'damaged', победил – 'ok').



Забележка за всички таблици: За всички атрибути, за които не е указано, че могат да приемат null стойност, да се счита, че съществува not null ограничение.

1. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име и държава на корабите, които никога не са потъвали в битка (може и да не са участвали).

```
SELECT name, country
FROM ships _____ outcomes ON name=ship
_____ classes ON ships.class=classes.class
WHERE _____;
```

2. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име, водоизместимост и брой оръдия на най-леките кораби с най-много оръдия.

```
SELECT name, displacement, numguns
FROM classes c JOIN ships s ON s.class=c.class
WHERE displacement = (SELECT _____
                      FROM classes) AND
numguns = (SELECT _____
            FROM classes c1
            WHERE _____);
```

3. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име на битките, в които е участвал един кораб.

```
SELECT battle
FROM outcomes o1
WHERE _____ (SELECT *
FROM outcomes o2
WHERE _____);
```

4. Да се посочи заявката, която извежда име на класа и брой на потъналите в битка кораби за съответния клас, за тези класове с повече от 5 кораба.

A)

```
SELECT c.class
FROM classes c JOIN ships s ON c.class=s.class
GROUP BY c.class
HAVING COUNT(name)>5
INTERSECT
SELECT class
FROM ships s JOIN outcomes o ON s.name=o.ship
WHERE o.result='sunk';
```

Б)

```
SELECT class, COUNT(DISTINCT name)
FROM ships s JOIN outcomes o ON s.name=o.ship
WHERE result='sunk' AND class IN (SELECT c.class
FROM classes c JOIN ships s ON c.class=s.class
GROUP BY c.class
HAVING COUNT(name)>5)
GROUP BY class;
```

В)

```
SELECT class, COUNT(DISTINCT name)
FROM ships s JOIN outcomes o ON s.name=o.ship
WHERE result='sunk' AND class IN (SELECT c.class
FROM classes c JOIN ships s ON c.class=s.class
WHERE COUNT(name)>5
GROUP BY c.class)
GROUP BY class;
```

Г)

```
SELECT s.class, COUNT(DISTINCT name)
FROM ships s JOIN outcomes o ON s.name=o.ship
JOIN (SELECT c.class
FROM classes c JOIN ships s ON c.class=s.class
GROUP BY c.class, name
HAVING COUNT(name)>5) t ON s.class=t.class
WHERE result='sunk'
GROUP BY s.class;
```

Задача 3. (30 min)

Дадена е базата от данни **Movies**, в която се съхранява информация за филми, филмови студии, които ги произвеждат, продуцентите на филмите, както и актьорите, които участват в тях.

Таблицата **Movie** съдържа информация за филми. Атрибутите *title* и *year* заедно формират първичния ключ.

title – заглавие;

year – година, в която е заснет филмът;

length – дължина в минути;

incolor – 'Y' за цветен филм и 'N' за чернобял;

studioName – име на студио, външен ключ към

Studio.name;

producerc# - номер на сертификат на продуцента, външен ключ към *MovieExec.cert#*.

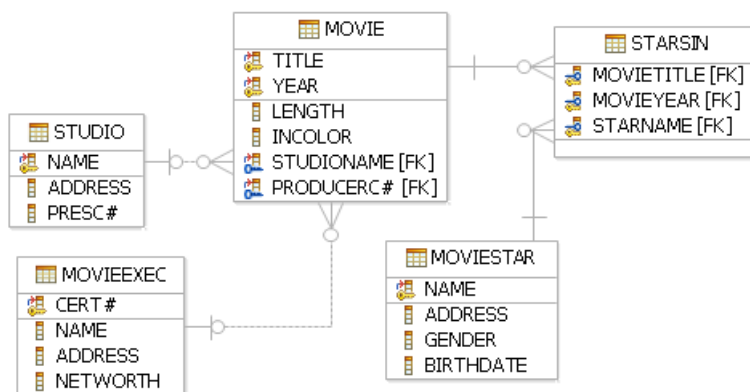
cert#.

Таблицата **Studio** съдържа информация за филмови студия:

name – име, първичен ключ;

address – адрес;

presc# - номер на сертификат на президента на студиото.



Таблицата **MovieStar** съдържа информация за филмови звезди:

name – име, първичен ключ;

address – адрес;

gender – пол, 'M' за мъж и 'F' за жена;

birthdate – рождена дата.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ. Атрибутите *movietitle* и *movieyear* образуват външен ключ към *Movie*.

movietitle – заглавие на филма;

movieyear – година на заснемане на филма;

starname – име на филмовата звезда, външен ключ към *MovieStar.name*.

Таблицата **MovieExec** съдържа информация за продуцентите на филми.

cert# – номер на сертификат, първичен ключ;

name – име;

address – адрес;

networth – нетни активи;

Забележка за всички таблици: Всички атрибути, които не участват във формирането на първичен ключ, могат да приемат null стойност.

1. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе за всяко студио името на студиото, заглавието и годината на филма, излязъл последно на екран за това студио.

```

SELECT studioName, title, year
FROM movie m
WHERE year = (SELECT _____
               FROM movie
               WHERE _____);
  
```

2. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име на продуцент и обща дължина на продуцираните от него филми, за тези продуценти, които имат поне един филм преди 1980 г.

```
SELECT name, _____
FROM movieexec JOIN movie ON producerc# = cert#
_____;
```

3. Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име на актьорите, участвали във филми на продуценти с най-големи нетни активи, както и заглавие на филмите, в които са участвали, име на продуцент и нетни активи.

```
SELECT starname, title, name, networth
FROM starsin JOIN movie ON movietitle=title AND movieyear=year
      JOIN      (SELECT cert#, networth, name
                  FROM movieexec
                  WHERE _____) t
_____;
```

4. Заградете буквата на заявката, която извежда името на продуцента, заглавието и годината на всички филми, продуцирани от продуцента на филма 'Interstellar'.

A)

```
SELECT name, title, year
FROM movie, movieexec
WHERE producerc#=cert# AND title='Interstellar' AND cert# IN (SELECT producerc#
                                                                FROM movie
                                                                WHERE title='Interstellar');
```

Б)

```
SELECT t.name, title, year
FROM movie m JOIN (SELECT name, cert#
                   FROM movieexec
                   WHERE EXISTS (SELECT producerc#
                                FROM movie
                                WHERE title='Interstellar')) t
ON m.producerc#=t.cert#;
```

В)

```
SELECT name, title, year
FROM movie JOIN movieexec ON producerc#=cert#
WHERE cert# = ANY (SELECT producerc#
                  FROM movie
                  WHERE title='Interstellar');
```

Г)

```
SELECT DISTINCT name, movietitle, movieyear
FROM movie JOIN movieexec ON producerc#=cert#
      JOIN starsin ON year=movieyear AND title=movietitle
WHERE cert# IN (SELECT producerc#
               FROM movie
               WHERE title='Interstellar');
```

Задача 4. Дадена е базата от данни Ships, в която се съхранява информация за кораби и тяхното участие в битки по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба.

Таблицата **Classes** съдържа информация за класовете кораби:

class – име на класа, първичен ключ;

type – тип ('bb' за бойни кораби, 'bc' за бойни крайцери);

country – държава, която строи такива кораби;

numGuns – брой на основните оръдия, може да приема стойност *null*;

bore – калибър на оръдието (в инчове), може да приема стойност *null*;

displacement – водоизместимост (в тонове), може да приема стойност *null*.

Таблицата **Ships** съдържа информация за корабите:

name – име на кораб, първичен ключ;

class – име на класа на кораба, външен ключ към таблицата Classes;

launched – година, в която корабът е пуснат на вода, може да приема стойност *null*.

Таблицата **Battles** съхранява информация за битките:

name – име на битката, първичен ключ;

date – дата на провеждане.

Таблицата **Outcomes** съдържа информация за резултата от участието на даден кораб в дадена битка.

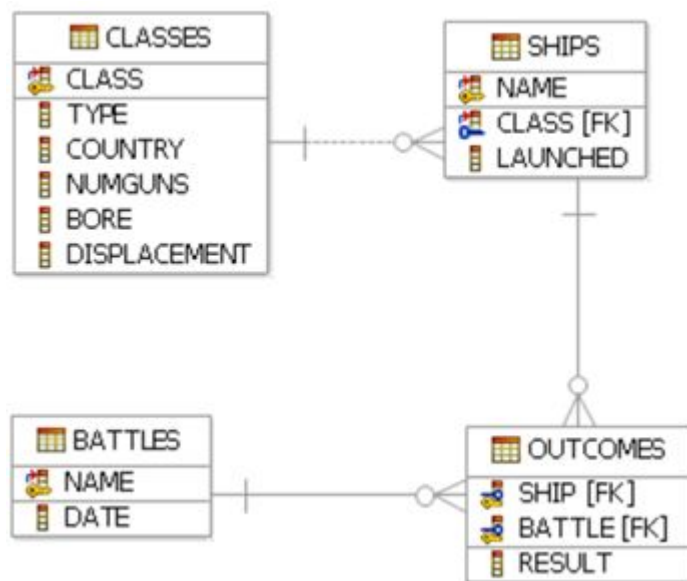
Атрибутите *ship* и *battle* заедно формират първичния ключ.

ship – име на кораба, външен ключ към таблицата Ships;

battle – име на битката, външен ключ към таблицата Battles;

result – резултат (потънал – 'sunk', повреден – 'damaged', победил – 'ok').

Забележка за всички таблици: За всички атрибути, за които не е посочено, че могат да приемат стойност *null*, да се счита, че съществува ограничение *not null*.



1. Да се посочи заявката, която извежда всички държави, които имат поне един кораб, участвал в битка, както и броя на потъналите кораби за всяка от държавите.

A)

```
SELECT c.country , COUNT(o.result)
FROM classes c left join ships s ON c.class=s.class
LEFT JOIN outcomes o ON s.name=o.ship
WHERE result='sunk' OR result IS NOT NULL
GROUP BY c.country;
```

Б)

```
SELECT c.country , COUNT(o.result)
FROM classes c JOIN ships s ON c.class=s.class
JOIN outcomes o ON s.name=o.ship
WHERE result='sunk'
GROUP BY c.country;
```

В)

```
SELECT c.country , COUNT(o.result)
FROM classes c join ships s ON c.class=s.class
JOIN outcomes o ON s.name=o.ship
JOIN battles b ON o.battle=b.name
ORDER BY c.country
HAVING result ='sunk';
```

Г)

```
SELECT DISTINCT c.country, (SELECT COUNT(o.result)
                             FROM classes c1 JOIN ships s
                             ON c1.class=s.class
                             JOIN outcomes o ON s.name=o.ship
                             WHERE result='sunk'
                             AND c1.country=c.country)
FROM classes c;
```


A)

Б)

B)

Г)

```
SELECT DISTINCT battle
FROM outcomes o CROSS JOIN classes c
GROUP BY battle
HAVING COUNT(DISTINCT country)>(SELECT COUNT(DISTINCT country)
                                FROM outcomes o, classes c
WHERE battle='Coral Sea');
```

Задача 4. Дадена е базата от данни *Movies*, в която се съхранява информация за филми, филмови студиа, които ги произвеждат, продуцентите на филмите, както и актьорите, които участват в тях.

Таблицата **Movie** съдържа информация за филми. Атрибутите *title* и *year* заедно формират първичния ключ.

title – заглавие;

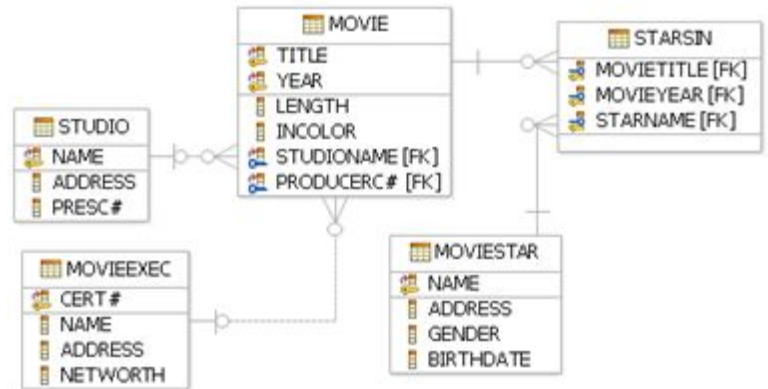
year – година, в която е заснет филмът;

length – дължина в минути;

incolor – 'Y' за цветен филм и 'N' за чернобял;

studioName – име на студио, външен ключ към таблицата *Studio*;

producerc# – име на продуцент, външен ключ към таблицата *MovieExec*.



Таблицата **Studio** съдържа информация за филмови студиа:

name – име, първичен ключ;

address – адрес;

presc# – номер на сертификат на президента на студиото.

Таблицата **MovieStar** съдържа информация за филмови звезди:

name – име, първичен ключ;

address – адрес;

gender – пол, 'M' за мъж и 'F' за жена;

birthdate – рождена дата.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ. Атрибутите *movietitle* и *movieyear* образуват външен ключ към таблицата *Movie*.

movietitle – заглавие на филма;

movieyear – година на заснемане на филма;

starname – име на филмовата звезда, външен ключ към таблицата *MovieStar*.

Таблицата **MovieExec** съдържа информация за продуцентите на филми.

cert# – номер на сертификат, първичен ключ;

name – име;

address – адрес;

networth – нетни активи;

birthdate – рождена дата.

Забележка за всички таблици: Всички атрибути, които не участват във формирането на първичен ключ, могат да приемат стойност *null*.

1. Да се посочи заявката, която извежда името на продуцента и имената на филмите, продуцирани от продуцента на 'Pretty Woman'. Възможно е името на продуцента на филма да не е известно.

A)

```
SELECT t.name, title
FROM movie m JOIN (SELECT name, cert#
                    FROM movieexec
                    WHERE cert# IN (SELECT producerc#
                                    FROM movie
                                    WHERE title='Pretty Woman')) t
ON m.producerc#=t.cert#;
```

Б)

```
SELECT name, title
FROM (SELECT cert#
      FROM movieexec
      INTERSECT
      SELECT producerc#
      FROM movie
      WHERE title ='Pretty Woman') t
```

В)

```
SELECT name, title
FROM movie m JOIN movieexec me ON m.producerc#=me.cert#
WHERE title = 'Pretty Woman';
```

Г)

```
SELECT name, title
FROM movie m LEFT JOIN movieexec me ON m.producerc#=me.cert#
WHERE me.cert# NOT IN (SELECT producerc#
                      FROM movie
                      WHERE title='Pretty Woman');
```

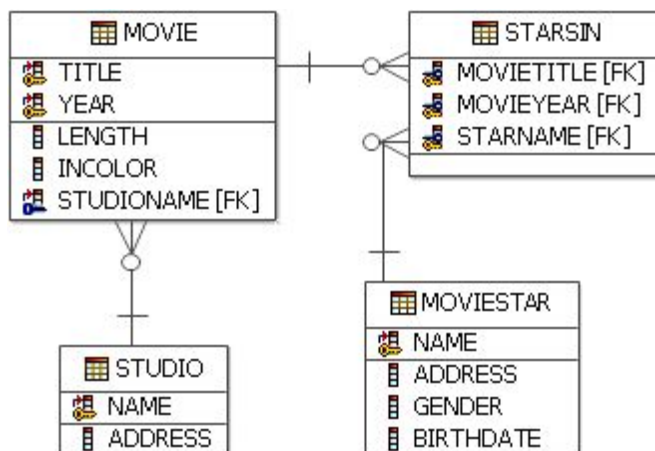

Задача 4 (10 точки). Дадена е базата от данни Movies.

Таблицата **Studio** съдържа информация за филмови студиа:

name – име, първичен ключ;
address – адрес.

Таблицата **Movie** съдържа информация за филми. Колоните *title* и *year* заедно формират първичния ключ.

title – заглавие;
year – година, в която филмът е заснет;
length – дължина в минути;
incolor – 'Y' за цветен филм и 'N' за черно-бял;
studioName – име на студио, външен ключ.



Таблицата **MovieStar** съдържа информация за филмови звезди:

name – име;
address – адрес;
gender – пол, 'M' за мъж и 'F' за жена;
birthdate – рождена дата.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите колони заедно формират първичния ключ. Колоните *movietitle* и *movieyear* образуват външен ключ към Movie.

movietitle – заглавие на филма;
movieyear – година на заснемане на филма;
starname – име на филмовата звезда, външен ключ.

Условие 1. Да се посочи заявката, която извежда имената и адресите на всички актриси от София, както и на всички филмови студиа от София. Резултатите да са сортирани по адрес.

а)

```
SELECT MS.name, MS.address
FROM MovieStar AS MS
JOIN Studio S ON MS.address =
S.address
WHERE MS.address LIKE '%Sofia%'
AND gender = 'F'
ORDER BY MS.address;
```

в)

```
SELECT name, address
FROM MovieStar MS
WHERE gender = 'F'
ORDER BY address
UNION ALL
SELECT name, address
FROM Studio
WHERE address LIKE '%Sofia%'
ORDER BY address;
```

г)

```
SELECT DISTINCT name, address
FROM MovieStar
FULL JOIN Studio ON address LIKE
'%Sofia%'
WHERE gender LIKE 'F'
ORDER BY address;
```

б)

```
SELECT *
FROM (SELECT name, address
FROM MovieStar
WHERE gender = 'F'
UNION
SELECT name, address
FROM Studio) T
WHERE T.address LIKE '%Sofia%'
ORDER BY T.address;
```

г)

```
SELECT DISTINCT name, address
FROM MovieStar INTERSECT Studio
WHERE address IS NOT NULL
AND gender LIKE 'F'
GROUP BY address
HAVING address LIKE '%Sofia%';
```

Условие 2. Да се посочи заявката, която за всяко студио с най-много три черно-бели филма извежда името му, адреса и средната дължина на филмите (без значение дали са цветни) на това студио. Студиа без филми също да се извеждат.

```
a) SELECT name, address, AVG(length) AS avgLength
FROM Studio
LEFT JOIN Movie ON name = studioName
GROUP BY studioName, address
HAVING COUNT(inColor = 'y') <= 3;
```

```
б) SELECT DISTINCT name, address, avgLength
FROM Studio, (SELECT studioName, AVG(length) AS avgLength
              FROM Movie
              GROUP BY studioName) Averages
WHERE NAME = ANY (SELECT studioName
                  FROM Movie
                  WHERE inColor = 'n'
                  GROUP BY studioName
                  HAVING COUNT(title) <= 3);
```

```
в) SELECT Studio.name, Studio.address, AVG(Movie.length) AS avgLength
FROM Movie
RIGHT JOIN Studio ON studioName = name
GROUP BY name, address
HAVING (SELECT COUNT(*) FROM Movie WHERE inColor = 'n') <= 3;
```

```
г) SELECT name, address, AVG(length) AS avgLength
FROM Studio
LEFT JOIN Movie ON name = studioName
WHERE NAME NOT IN (SELECT studioName
                  FROM Movie
                  WHERE inColor = 'n'
                  GROUP BY studioName
                  HAVING COUNT(*) > 3)
GROUP BY name, address;
```


1. Да се посочи заявката, която извежда имената на всички филмови звезди, чието име не завършва на "а" и са играли както в цветни, така и в черно-бели филми.

a)

```
SELECT name
FROM MovieStar, StarsIn, Movie
WHERE name = starName AND movieTitle = title AND movieYear =
year
      AND name != '%a' AND inColor = 'y' AND inColor = 'n';
```

б)

```
SELECT MovieStar.name
FROM MovieStar
WHERE NOT (name LIKE '%a')
      AND name IN (SELECT starName
                  FROM StarsIn
                  JOIN Movie ON movieTitle = title AND movieYear = year
                  WHERE inColor = 'y' OR inColor = 'n');
```

в)

```
SELECT DISTINCT starName
FROM StarsIn
INNER JOIN Movie ON movieTitle = title AND movieYear = year
WHERE starName NOT LIKE '%a' AND inColor = 'y'
      AND starName = (SELECT starName
                    FROM StarsIn, Movie
                    WHERE inColor = 'n');
```

г)

```
SELECT starName
FROM StarsIn
JOIN Movie ON movieTitle = title AND movieYear = year
WHERE starName NOT LIKE '%a' AND inColor = 'y'
INTERSECT
SELECT starName
FROM StarsIn
JOIN Movie ON movieTitle = title AND movieYear = year
WHERE inColor = 'n';
```

2. Посочете заявката, която извежда за всяка филмова звезда, играла в най-много 5 филма, следната информация:

- име;
- рождена година;
- брой студия, с които е работила.

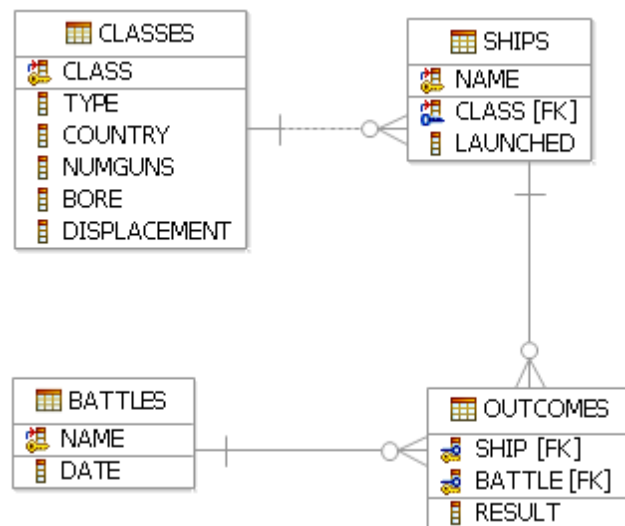
Ако за дадена звезда няма информация в какви филми е играла, за нея също да се извежда ред (с брой студия, равен на 0).

- a) `SELECT DISTINCT name, birthdate.year, COUNT(studioName)
FROM MovieStar, StarsIn, Movie
WHERE name = starname AND
((movieTitle = title AND movieYear = year) OR title IS
NULL)
GROUP BY name, birthdate.year
HAVING COUNT(title) <= 5;`
- б) `SELECT name, YEAR(birthdate), COUNT(DISTINCT studioName)
FROM MovieStar
LEFT OUTER JOIN StarsIn ON name = starname
LEFT JOIN Movie ON movieTitle = title AND movieYear = year
GROUP BY name
HAVING COUNT(title) <= 5;`
- в) `SELECT StarsIn.starname, YEAR(birthdate),
COUNT(DISTINCT studioName)
FROM Movie
JOIN StarsIn ON movieTitle = title AND movieYear = year
RIGHT OUTER JOIN MovieStar ON MovieStar.name = StarsIn.starname
GROUP BY StarsIn.starname
HAVING COUNT(DISTINCT title) <= 5;`
- г) `SELECT name, year(birthdate), COUNT(SELECT DISTINCT studioName
FROM Movie
JOIN StarsIn ON title = movieTitle AND year = movieYear
WHERE starname = name)
FROM MovieStar
HAVING COUNT(SELECT * FROM StarsIn WHERE starname = name) <= 5
ORDER BY name, year(birthdate);`

Задача 8. (10 т.) Дадена е базата от данни Ships, в която се съхранява информация за кораби (*Ships*) и тяхното участие в битки (*Battles*) по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба (*Classes*).

Таблицата *Classes* съдържа информация за класовете кораби:

- *class* – име на класа, първичен ключ;
- *type* – тип ('bb' за бойни кораби и 'bc' за бойни крайцери);
- *country* – държавата, която строи такива кораби;
- *numGuns* – броят на основните оръдия;
- *bore* – калибърът им (диаметърът на отвора на оръдието в инчове);
- *displacement* – водоизместимост (в тонове).



Таблицата *Ships* съдържа информация за корабите:

- *name* – име на кораб, първичен ключ;
- *class* – име на неговия клас, външен ключ към *Classes.class*;
- *launched* – годината, в която корабът е пуснат на вода.

Таблицата *Battles* съхранява информация за битките:

- *name* – име на битката, първичен ключ;
- *date* – дата на провеждане.

Таблицата *Outcomes* съдържа информация за резултата от участието на даден кораб в дадена битка (колониите ship и battle заедно формират първичния ключ):

- *ship* – име на кораба, външен ключ към *Ships.name*;
- *battle* – име на битката, външен ключ към *Battle.name*;
- *result* – резултат (потънал-'sunk', повреден – 'damaged', победил – 'ok').

За така описаната база данни, решете следните задачи:

1. Оградете буквата на заявката, която извежда имената на всички кораби, пуснати на вода в година, в която е имало битка (не е задължително корабът да е участвал в нея).

| | |
|---|--|
| А) <pre>select name from ships where launched = any (select year(date) from battles where count(*) >= 1);</pre> | Б) <pre>select distinct ships.name from battles , ships where launched = year(date);</pre> |
| В) <pre>select name from battles where exists (select distinct * from ships where year(date) = launched);</pre> | Г) <pre>select distinct name from ships join battles on launched = year(date);</pre> |

2. Оградете буквата на заявката, която за всички държави, които имат най-много 3 (евентуално 0) кораба, извежда името на държавата и броя потънали кораби (който също може да бъде 0).

| | |
|---|---|
| А) <pre>select country, count(result) from classes c left join ships s on c.class = s.class left join outcomes o on s.name = o.ship where o.result = 'sunk' group by country having count(ship) <= 3;</pre> | Б) <pre>select country, count(result is 'sunk') from ships, classes, outcomes where count(ship) <= 3 or ship is null;</pre> |
| В) <pre>select distinct classes.country, sunk_cnt from classes right join (select country, count(*) as sunk_cnt from classes c join ships s on c.class = s.class join outcomes o on s.name = o.ship where result = 'sunk' group by country) sunk on classes.country = sunk.country where sunk_cnt <= 3;</pre> | Г) <pre>select country, count(result = 'sunk') as sunk_cnt from ships s join outcomes o on s.name = o.ship right join classes c on s.class = c.class where count(*) <= 3 group by country, sunk_cnt;</pre> |
| Д) <pre>select distinct country, (select count(*) from classes c2 join ships s on c2.class = s.class join outcomes o on s.name = o.ship where c2.country = c.country and result = 'sunk') from classes c where (select count(*) from classes c2 join ships s on c2.class = s.class where c2.country = c.country) <= 3;</pre> | |

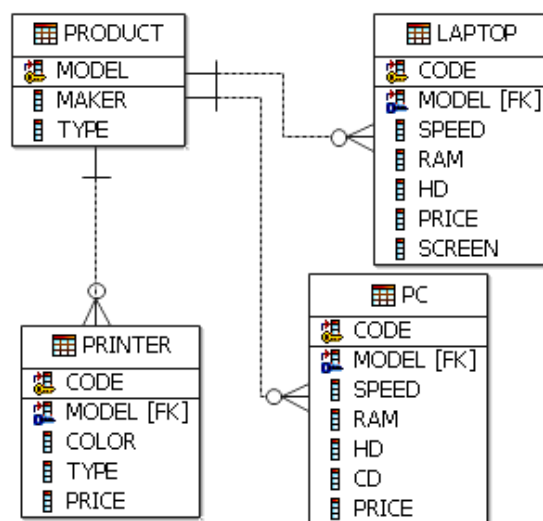
Задача 8. (10 т.) Дадена е базата от данни РС. В нея се съхранява информация за три вида продукти – настолни компютри, лаптопи и принтери.

Таблицата **Product** съдържа базова информация за всеки продукт:

- *model* – модел на продукта, първичен ключ;
- *maker* – производител на продукта;
- *type* – един от следните типове: 'PC', 'Laptop' или 'Printer'.

Таблицата **PC** съдържа специфична информация за настолните компютри:

- *code* – уникален идентификатор на дадена компютърна конфигурация, първичен ключ;
- *model* – модел на настолния компютър, външен ключ към Product.model. Може да имаме няколко различни компютърни конфигурации от един и същ модел, но с различни параметри;
- *speed* – тактова честота на процесора в MHz;
- *ram* – количество RAM памет в MB;
- *hd* – размер на твърдия диск в GB;
- *cd* – скорост на CD устройството;
- *price* – цена на настолния компютър.



Таблицата **Laptop** съдържа специфична информация за лаптопите. Атрибутите са аналогични на тези на PC, но липсва атрибутът CD и има атрибут за размера на екрана.

Таблицата **Printer** съдържа информация за принтерите:

- *code*, *model*, *price* – аналогични на едноименните атрибути в PC;
- *color* – 'y' за цветен принтер, 'n' за черно-бял;
- *type* – тип на принтера – 'Laser', 'Jet', 'Matrix'.

За така описаната база данни решете следните задачи:

1. Оградете буквата на заявката, която извежда всички производители на настолни компютри, които произвеждат и лаптопи.

| | |
|--|--|
| A) <pre>select distinct maker from product where product.type = 'PC' and maker in (select maker from product join laptop on product.model = laptop.model);</pre> | Б) <pre>select maker from product p1 cross join product p2 where p1.maker = p2.maker and p1.type = 'PC' and p2.type = 'Laptop' group by maker;</pre> |
| В) <pre>select distinct maker from product where type = 'PC' and type = 'Laptop';</pre> | Г) <pre>select maker from product where type = 'PC' union select maker from product where type = 'Laptop';</pre> |

2. Оградете буквата на заявката, която извежда кодовете, моделите и размерите на екраните на всички лаптопи, чиито производители имат не повече от три модела принтери (евентуално 0).

| | |
|---|--|
| A) <pre>select code, model, screen from laptop, product where maker is having count(printer.model) <= 3;</pre> | Б) <pre>select code, l.model, screen from product p left join laptop l on p.model = l.model having count(select * from product where maker = p.maker and type = 'Printer') <= 3;</pre> |
| В) <pre>select code, l.model, screen from laptop l inner join product p on l.model = p.model where maker not in (select maker from product where type = 'Printer' group by maker having count(*) > 3);</pre> | Г) <pre>select l.code, l.model, l.screen from laptop l join product p on l.model = p.model where maker in (select maker from product where type = 'Printer' group by maker having count(*) <= 3);</pre> |
| Д) <pre>select l.code, l.model, l.screen from product p left join laptop l on p.model = l.model left join printer on p.model = printer.model group by l.code having count(distinct printer.code) <= 3;</pre> | |

Б) (6 точки) Да се посочи заявката, която за всяка филмова звезда (без значение от пола), родена преди 1990 г., извежда възрастта, на която е играла за първи път във филм. Звезди, за които няма информация за техните участия във филми, не трябва да бъдат извеждани.

А)

```
SELECT NAME, MIN( MOVIEYEAR - YEAR(BIRTHDATE) ) AS DEBUT_AGE
FROM MOVIESTAR
JOIN STARSIN ON NAME = STARNAME
WHERE YEAR(BIRTHDATE) < 1990
GROUP BY NAME;
```

Б)

```
SELECT NAME, MIN(MOVIEYEAR) - YEAR(BIRTHDATE) AS DEBUT_AGE
FROM MOVIESTAR
JOIN STARSIN ON NAME = STARNAME
WHERE YEAR(BIRTHDATE) < 1990
GROUP BY NAME;
```

В)

```
SELECT NAME, MIN(MOVIEYEAR) - YEAR(BIRTHDATE) AS DEBUT_AGE
FROM MOVIESTAR
LEFT JOIN STARSIN ON NAME = STARNAME AND YEAR(BIRTHDATE) < 1990
GROUP BY NAME
HAVING MIN(MOVIEYEAR);
```

Г)

```
SELECT DISTINCT MS.NAME, MOVIEYEAR - YEAR(BIRTHDATE) AS DEBUT_AGE
FROM MOVIESTAR MS, STARSIN
WHERE MS.NAME = STARNAME AND YEAR(BIRTHDATE) < 1990
HAVING MOVIEYEAR <= ALL (SELECT MOVIEYEAR
                        FROM STARSIN SI
                        WHERE SI.STARNAME = MS.NAME);
```


Задача 9. Дадена е базата от данни *Ships*, в която се съхранява информация за кораби (*Ships*) и тяхното участие в битки (*Battles*) по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба (*Classes*).

Таблицата ***Classes*** съдържа информация за класовете кораби:

class – име на класа, първичен ключ;
type – тип ('bb' за бойни кораби и 'bc' за бойни крайцери);
country – държавата, която строи такива кораби;
numGuns – броя на основните оръдия;
bore – калибъра им (диаметърът на отвора на оръдието в инчове);
displacement – водоизместимост (в тонове).

Таблицата ***Ships*** съдържа информация за корабите:

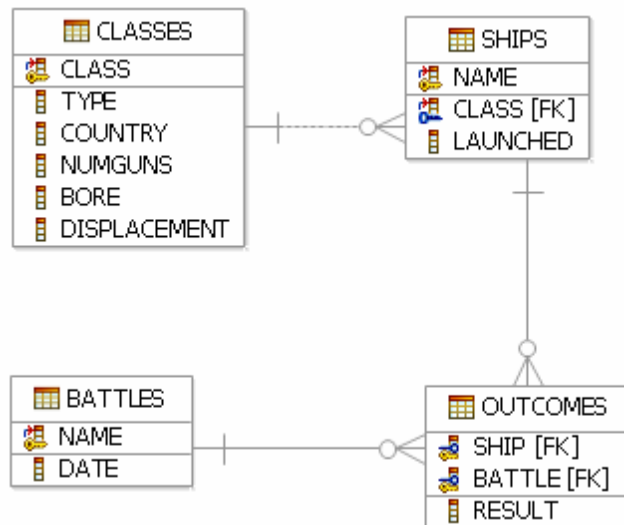
name – име на кораб, първичен ключ;
class – име на неговия клас;
launched – годината, в която корабът е пуснат на вода.

Таблицата ***Battles*** съхранява информация за битките:

name – име на битката, първичен ключ;
date – дата на провеждане.

Таблицата ***Outcomes*** съдържа информация за резултатата от участието на даден кораб в дадена битка (колониите *ship* и *battle* заедно формират първичния ключ):

ship – име на кораба;
battle – име на битката;
result – резултат (потънал-'sunk', повреден – 'damaged', победил – 'ok').



1. Посочете заявката, която извежда всички държави, които имат както класове с по-малко от 9 оръдия (*numguns*), така и класове с над 12 оръдия:

а)
 SELECT DISTINCT COUNTRY
 FROM CLASSES
 WHERE NUMGUNS<9 AND NUMGUNS>12;

б)
 SELECT DISTINCT C1.COUNTRY
 FROM CLASSES C1
 JOIN CLASSES C2 ON C1.COUNTRY=C2.COUNTRY
 WHERE C1.NUMGUNS<9 AND C2.NUMGUNS>12;

в)
 SELECT COUNTRY
 FROM CLASSES
 WHERE NUMGUNS<9
 UNION
 SELECT COUNTRY
 FROM CLASSES
 WHERE NUMGUNS>12;

г)
 SELECT DISTINCT COUNTRY
 FROM CLASSES
 WHERE NUMGUNS<9 AND COUNTRY =

```
(SELECT COUNTRY  
FROM CLASSES  
WHERE NUMGUNS>12);
```

2. Посочете заявката, която за всяка държава, участвала в не повече от 4 битки, извежда името ѝ и броя битки, в които е участвала. Ако дадена държава няма нито един кораб или не е участвала в нито една битка, за нея да извежда 0.

а)

```
SELECT COUNTRY, COUNT(DISTINCT BATTLE)  
FROM CLASSES C, SHIPS S, OUTCOMES O  
WHERE C.CLASS=S.CLASS AND S.NAME=O.SHIP  
GROUP BY COUNTRY  
HAVING COUNT(DISTINCT O.BATTLE)<4;
```

б)

```
SELECT COUNTRY, COUNT(O.BATTLE) AS NUM_BATTLES  
FROM CLASSES C  
LEFT JOIN SHIPS S ON C.CLASS=S.CLASS  
LEFT JOIN OUTCOMES O ON S.NAME=O.SHIP  
GROUP BY COUNTRY  
HAVING COUNT(O.BATTLE)<4;
```

в)

```
SELECT COUNTRY, COUNT(DISTINCT BATTLE)  
FROM OUTCOMES  
JOIN SHIPS ON NAME=SHIP  
RIGHT JOIN CLASSES ON CLASSES.CLASS=SHIPS.CLASS  
GROUP BY COUNTRY  
HAVING COUNT(DISTINCT OUTCOMES.BATTLE)<=3;
```

г)

```
SELECT C.COUNTRY, COUNT(O.BATTLE)  
FROM CLASSES AS C  
INNER JOIN SHIPS AS S ON C.CLASS=S.CLASS  
LEFT OUTER JOIN OUTCOMES AS O ON S.NAME=O.SHIP  
WHERE COUNT(O.BATTLE)<=3;
```

Задача 10. (10 точки) В базата от данни със схема:

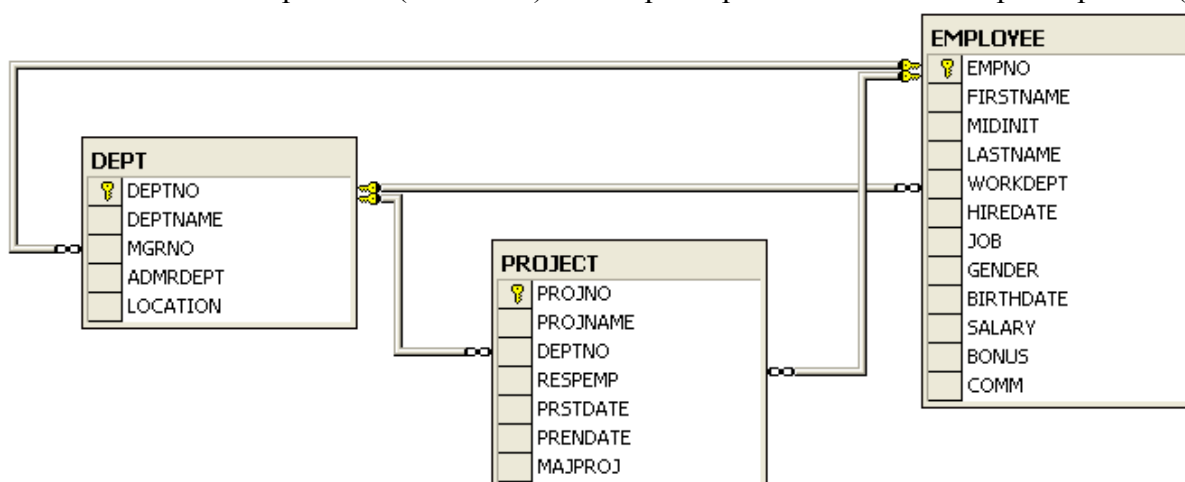
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

```
EMPLOYEE (EMPNO, FIRSTNAME, MIDNAME, LASTNAME, WORKDEPT, HIREDATE, JOB, GENDER,  
BIRTHDATE, SALARY, BONUS, COMM);
```

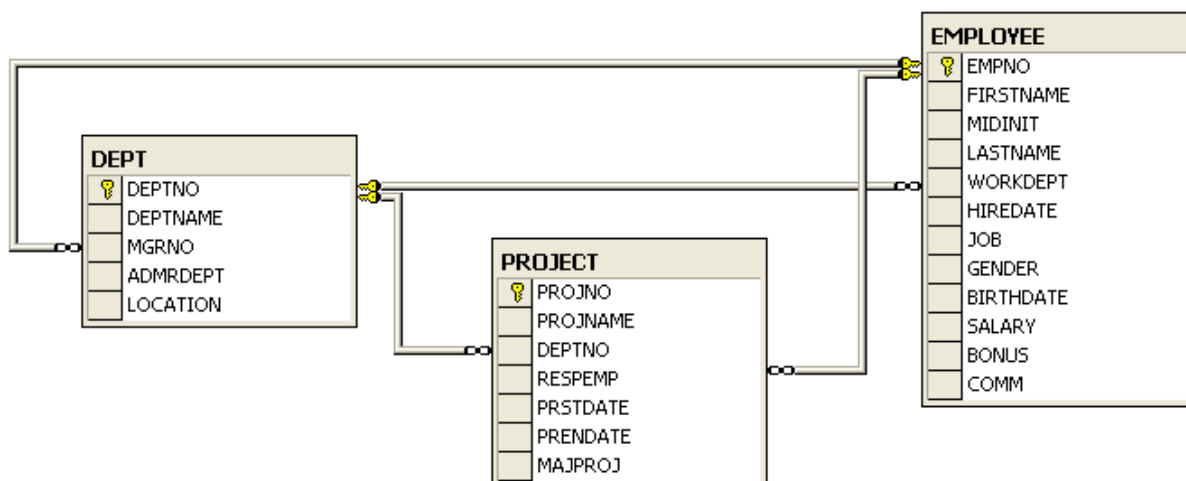
```
PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE, PRENDATE, MAJPROJ);
```

се съхранява информация за департаменти, работници и проекти.

- Таблицата DEPT съдържа информация за номер и име на департамент (DEPTNO и DEPTNAME) и за номер на работник (MGRNO) – менажер за този департамент;
- Таблицата EMPLOYEE съдържа информация за номер на работник (EMPNO), лични данни (FIRSTNAME, MIDINIT, LASTNAME), длъжност (JOB), заплата (SALARY) и номера на департамента (WORKDEPT), към който работи;
- Таблицата PROJECT съдържа информация за номер (PROJNO) и име на проект (PROJNAME), начална и крайна дата на проекта (PRSTDATE, PRENDATE), департамента, към който е проектът (DEPTNO) и номера на работника – менажер на проекта (RESPEMP).



А) Да се напише заявка, която извежда името на департамента, броя на проектите и сумарната заплата на менажерите на проекти от този департамент.



Б) Като се използва заявката от подточка А, да се напише заявка, която извежда името на департамента с най-много проекти.

Забележка: Ако има повече департаменти с брой на проектите, равен на максималния брой на проекти, тези департаменти също да се изведат.

Задача 9. (10 точки) В базата от данни със схема:

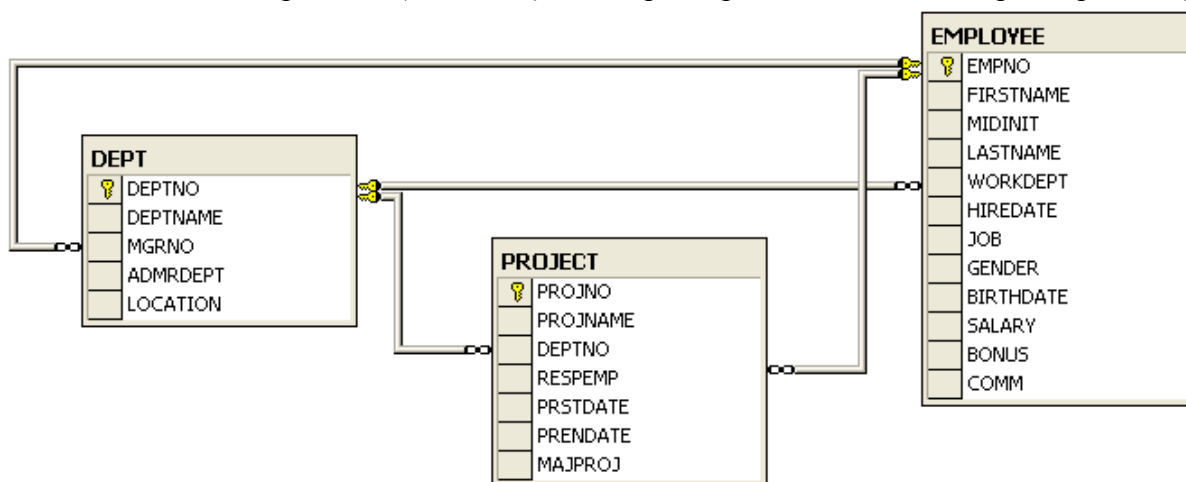
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

```
EMPLOYEE (EMPNO, FIRSTNAME, MIDNAME, LASTNAME, WORKDEPT, HIREDATE, JOB, GENDER,  
BIRTHDATE, SALARY, BONUS, COMM);
```

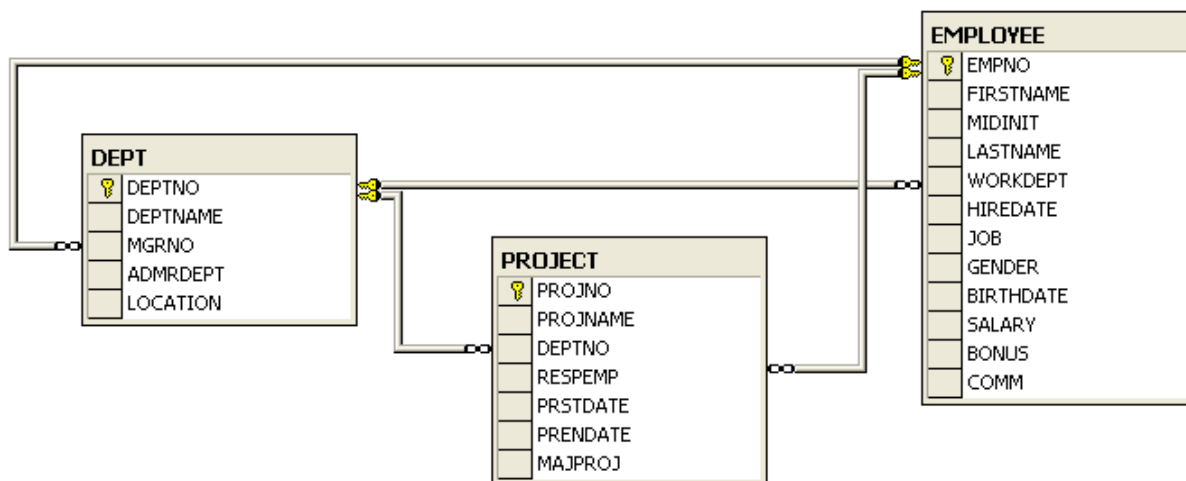
```
PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE, PRENDATE, MAJPROJ);
```

се съхранява информация за департаменти, работници и проекти.

- Таблицата DEPT съдържа информация за номер и име на департамент (DEPTNO и DEPTNAME) и за номер на работник (MGRNO) – менажер за този департамент;
- Таблицата EMPLOYEE съдържа информация за номер на работник (EMPNO), лични данни (FIRSTNAME, MIDINIT, LASTNAME), длъжност (JOB), заплата (SALARY) и номера на департамента (WORKDEPT), към който работи;
- Таблицата PROJECT съдържа информация за номер (PROJNO) и име на проект (PROJNAME), начална и крайна дата на проекта (PRSTDATE, PRENDATE), департамента, към който е проектът (DEPTNO) и номера на работника – менажер на проекта (RESPEMP).



А) Да се напише заявка, която извежда името на департамента, броя на проектите и сумарната заплата на менажерите на проекти от този департамент.



Б) Като се използва заявката от подточка А, да се напише заявка, която извежда името на департамента с най-висока сумарна заплата на менажерите на проекти от този департамент.

Забележка: Ако има повече департаменти със сумарна заплата равна на най-високата сумарна заплата, тези департаменти също да се изведат.

Задача 8. (7т.) В базата данни със схема:

```
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
Outcomes(ship, battle, result)
```

се съхранява информация за кораби (Ships) и тяхното участие в битки (Battles) по време на Втората Световна Война. Всеки кораб е построен по определен стереотип, определящ класа на кораба (Classes). Обикновено класът носи името на първия построен кораб от този клас.

Таблицата Classes съдържа информация за класовете кораби:

class – името на класа, първичен ключ;
type – типът ('bb' за бойни кораби и 'bc' за бойни крайцери);
country – страната (държавата), която строи такива кораби;
numGuns – броят на основните оръдия;
bore – калибърът им (диаметърът на отвор на оръдето в инчове);
displacement – водоизместимостта (тегло, в тонове).

Таблицата Ships съдържа информация за корабите:

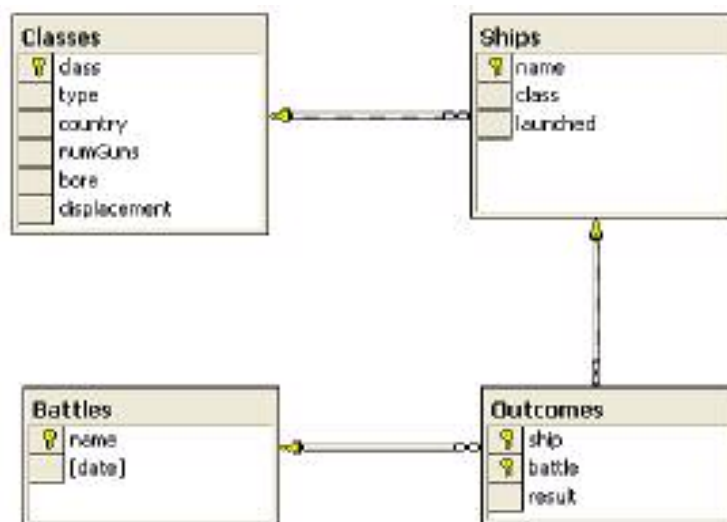
name – името на кораб, първичен ключ;
class – името на неговия клас;
launched – годината, в която корабът е пуснат на вода.

Таблицата Battles съхранява информация за битките:

name – името на битката, първичен ключ;
date – датата на провеждане.

Таблицата Outcomes съдържа информация за резултатата от участието на даден кораб в дадена битка (колоните **ship** и **battle** заедно формират първичния ключ):

ship – името на кораба;
battle – името на битката;
result – резултатът (потънал – 'sunk', повреден – 'damaged', победил – 'ok').



Задача 8.1. Посочете заявката, която извежда за всеки клас годината на най-рано и най-късно пуснатия на вода кораб:

- ```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class;
```
- ```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class
group by c.class;
```
- ```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class
group by c.class
having min(ss.launched) and max(ss.launched);
```
- ```
select c.class, min(ss.launched)
from classes c
join ships ss on ss.class = c.class
union all
select c.class, max(ss.launched)
from classes c
join ships ss on ss.class = c.class;
```

Задача 8.2. Посочете заявката, която извежда държавата/държавите с най-много класове:

- a)

```
select c.country
from classes c
where not exists
    ( select *
      from classes cl
      where cl.country != c.country and
            count(cl.class) > count(c.class)
    );
```
- b)

```
select c.country
from classes c
group by c.country
having max(count(c.class));
```
- c)

```
select c.country
from classes c
where count(*) = ( select max(count(cl.class)
                        from classes cl
                        group by cl.country
                    )
group by c.country;
```
- d)

```
select c.country
from classes c
group by c.country
having count(*)>= all ( select count(*)
                        from classes cl
                        group by cl.country
                    );
```


Задача 9. (4 т.) В базата от данни със схема:

Classes (class, type, country, numGuns, bore, displacement)

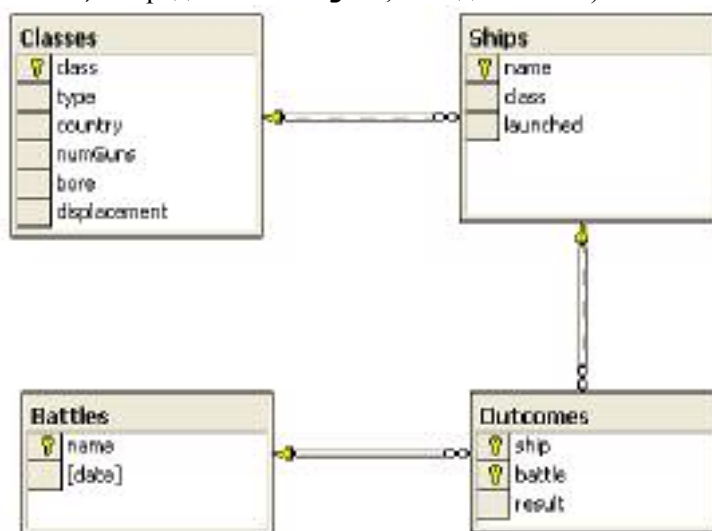
Ships (name, class, launched)

Battles (name, date)

Outcomes (ship, battle, result)

се съхранява информация за кораби и тяхното участие в битки по време на Втората Световна Война. Всеки кораб е построен по определен стереотип, определящ класа на кораба. Обикновено класът носи името на първия построен кораб от този клас.

- Таблицата **Classes** съдържа информация за име на класа, тип 'bb' за бойни кораби и 'bc' за бойни крайцери), страната, която строи такива кораби, броя на основните оръдия, калибъра им (диаметъра на отвора на оръдието в инчове) и водоизместимостта (тегло в тонове).
- Таблицата **Ships** съдържа информация за име на кораб, име на неговия клас и годината, в която корабът е пуснат на вода.
- Таблицата **Battles** съхранява имена и дати на провеждане на битки.
- Таблицата **Outcomes** съдържа информация за резултатата от участието на даден кораб в дадена битка (потънал - 'sunk', повреден - 'damaged', победил - 'ok').



Посочете заявката, която извежда имената на битките, в които няма оцелели кораби (всички участвали кораби са потънали):

a)

```

select distinct b.name
from Battles b
join Outcomes o on b.name = o.battle
where o.result = 'sunk';

```

б)

```

select o.battle
from Outcomes o
group by o.battle,o.result
having o.result = 'sunk';

```

в)

```

select distinct o.battle
from Outcomes o
left join Outcomes o1 on o.battle = o1.battle and
                      o.ship = o1.ship and
                      o1.result = 'sunk'
group by o.battle
having count(o.ship)= count(o1.ship);

```

г)

```

select o.battle
from Outcomes o
group by o.battle
having count(case when o.result = 'sunk' then 'Y' end)=0;

```