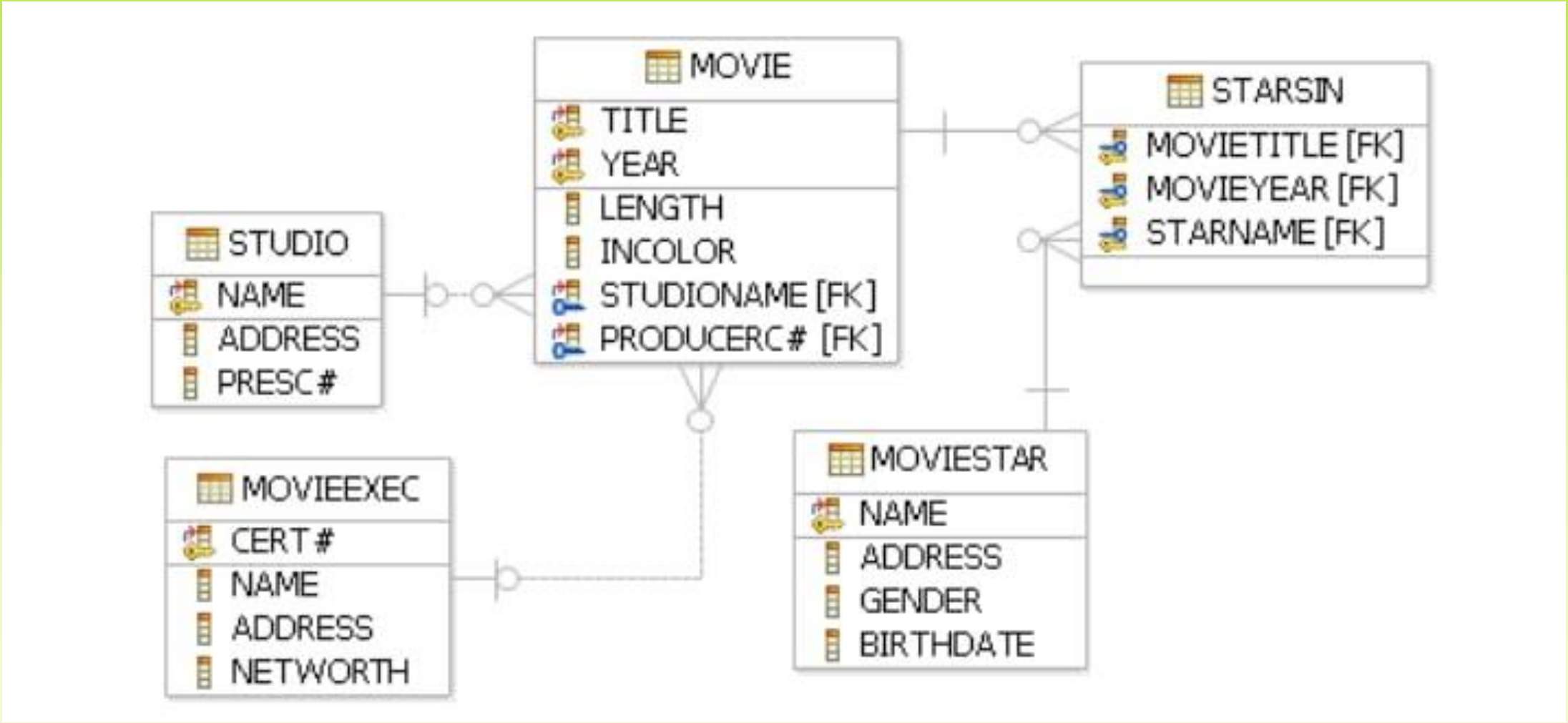
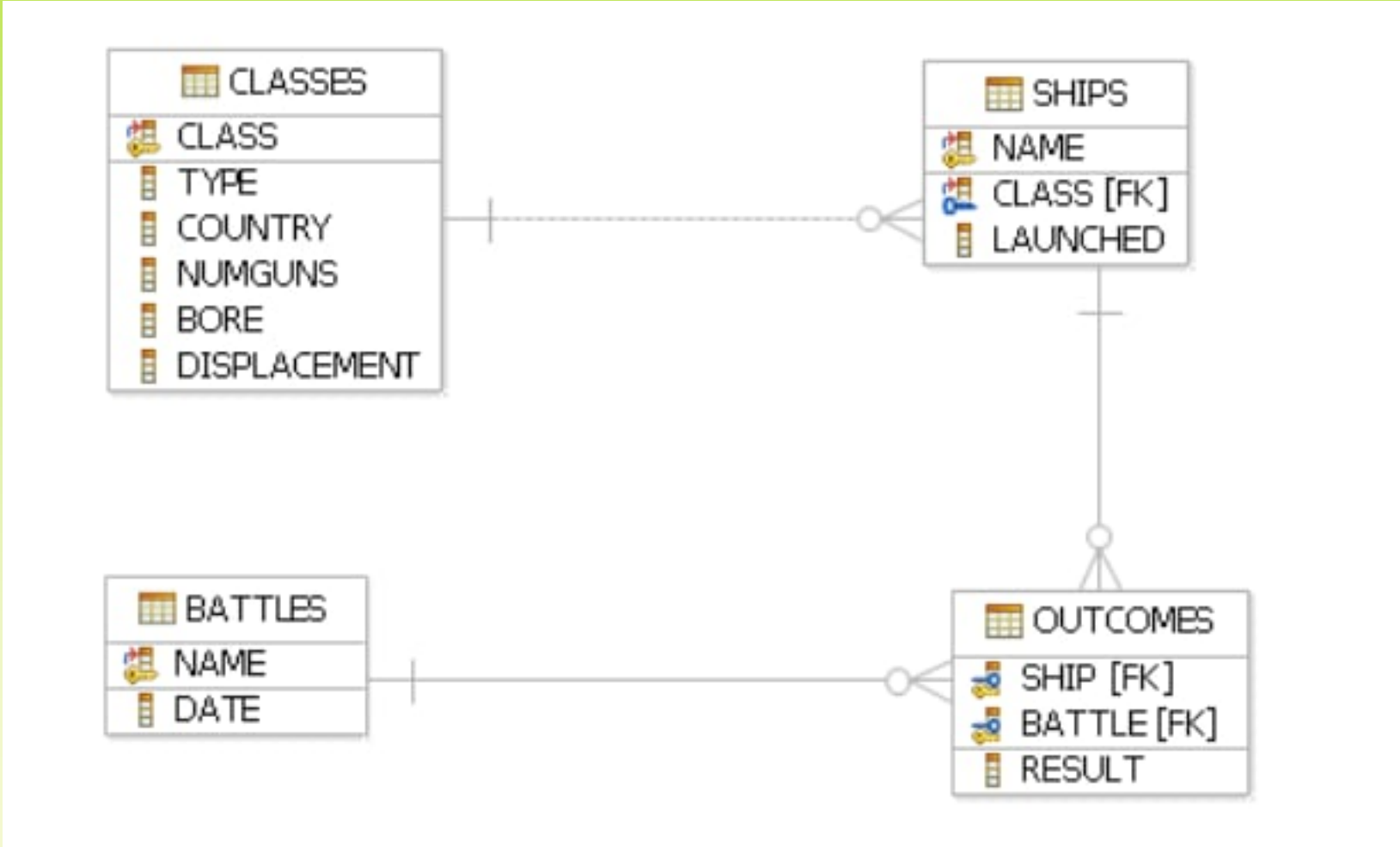
Модификация на данни

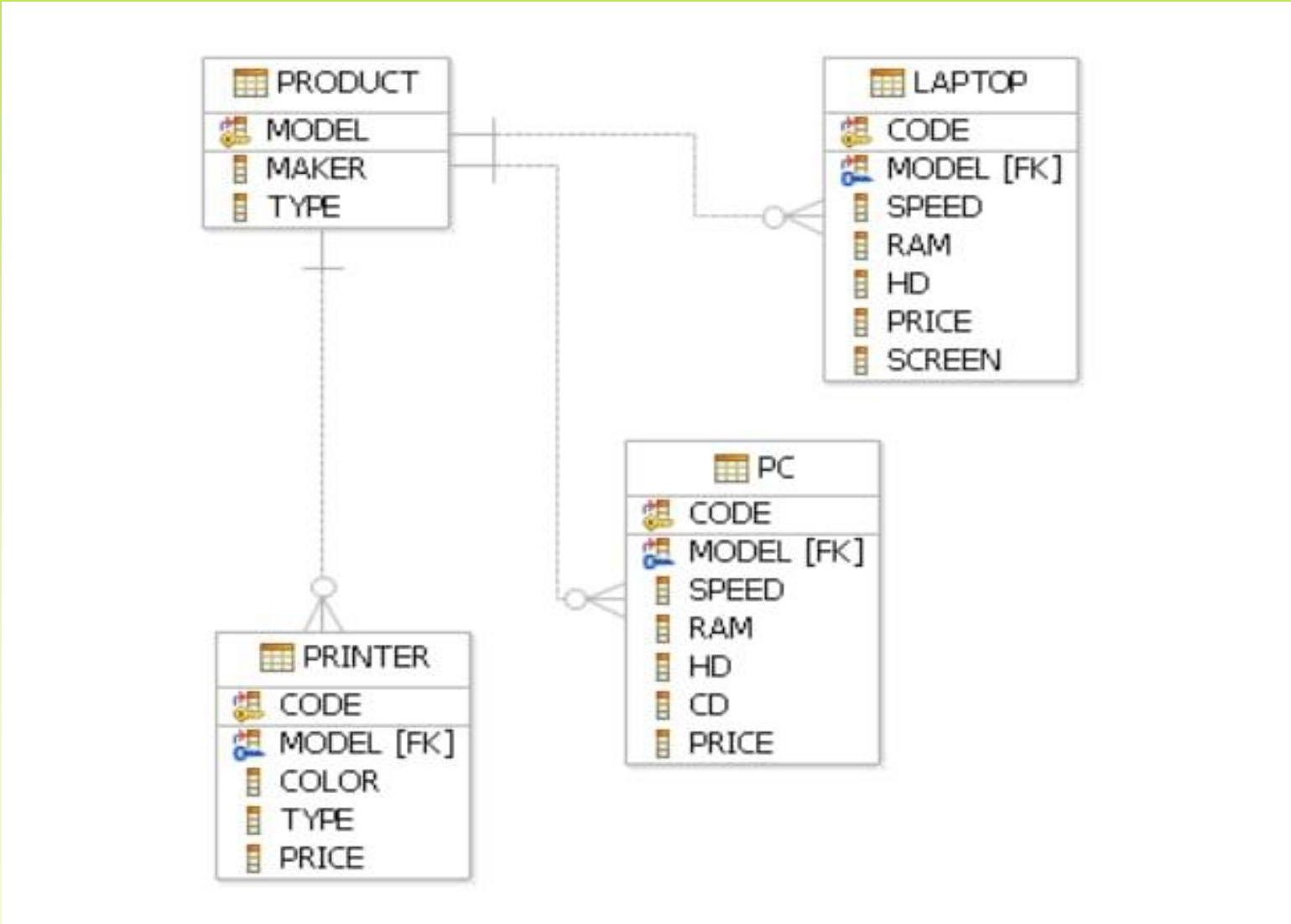
База данни Movies



База данни Ships



База данни PC



Уводни думи

Това упражнение съдържа скрипт за създаване на нова база - mods. Тя съдържа всички таблици от базите MOVIES, PC и SHIPS на едно място, като са премахнати всякакви ограничения свързани с валидация на данните (нужно е заради задачите в това упражнение).

В случаи, че задача от това упражнение променя данни по начин, които влиза в конфликт с други задачи - можете да пресъздадете наново базата с този скрипт.

Също така може да използвате транзакции, за да възстановявате базата в състоянието, в което е била преди да започнете да решавате някоя задача. Чрез

BEGIN TRANSACTION

може да укажете начало на транзакция. След това да изпълните последователност от заявки, свързани с решаването на някоя задача. Накрая чрез:

ROLLBACK

Добавяне на нови редове в таблици

INSERT командата може да се използва за добавяне на един или повече редове в таблица.

Синтаксисът на командата е:

```
INSERT INTO table_name (column1, column2, column3, ...)
        VALUES (value11, value12, value13, ...),
               (value21, value22, value23, ...), ...
```

Пример: Да се добави нов немски клас кораби 'Bayern' от тип 'bc' в таблицата classes на базата ships – да има 8 броя 15-инчови оръдия и водоизместимост 32200.

```
INSERT INTO classes (class, type, country, numguns, bore, displacement)
VALUES ('Bayern', 'bc', 'Germany', 8, 15, 32200)
```

Възможно е да не указваме стойност за някоя колона. Например:

```
INSERT INTO classes (class, country, numguns, bore, displacement)
VALUES ('Bayern', 'Germany', 8, 15, 32200)
```

Ако колоната **type** не допуска NULL стойности - ще получиме грешка. Ако допуска - тогава този нов ред ще има стойност NULL за колоната **type**.

Ако указваме стойности за всички колони на таблицата и го правим в реда, в който са описани при създаването на таблица (реда в който ги виждаме при **SELECT** *) - тогава може да не изреждаме имената на колоните в **INSERT** командата.

Например:

```
INSERT INTO classes VALUES ('Bayern', 'bc', 'Germany', 8, 15, 32200)
```

Новите редове, които добавяме с **INSERT**, може да са резултат от **SELECT** заявка. Например, нека да копираме всички налични британски класове и като американски със същите параметри:

```
INSERT INTO classes
    SELECT class, type, 'USA', numguns, bore, displacement
     FROM classes
     WHERE country = 'Gt.Britain'
```

Обновяване на стойности в колони

Това става с командата **UPDATE**, чийто синтаксис е:

UPDATE table_name

SET column1 = value1, column2 = value2, ...

WHERE condition

Например, нека да увеличим с 2 броя на оръдията на всички британски кораби и да зададем стойност 18 за калибъра:

```
UPDATE classes
SET numguns = numguns + 2, bore = 18
WHERE country = 'Gt.Britain'
```

Ако няма **WHERE** клауза (или условието в **WHERE** е винаги истина) - командата ще обнови всички редове в таблицата.

Изтриване на редове

Изтриването става с командата **DELETE**. Синтаксисът е:

DELETE FROM table_name **WHERE** condition

Ако няма **WHERE** клауза (или условието в **WHERE** е винаги истина) - командата ще обнови всички редове в таблицата.

Например, нека изтрием американските кораби с 9 оръдия:

DELETE FROM classes **WHERE** country = 'USA' **AND** numguns = 9

Задачи

Задача 1: Да се добави информация за актрисата Nicole Kidman. За нея знаем само, че е родена на 20-и юни 1967.

Задача 2: Да се изтрият всички продуценти с печалба (networth) под 10 милиона.

Задача 3: За всеки персонален компютър се продава и 15-инчов лаптоп със същите параметри, но с $500 по-скъп. Кодът на такъв лаптоп е със 100 по-голям от кода на съответния компютър. Добавете тази информация в базата.

Задачи

Задача 4: Производител A купува производител B. На всички продукти на B променете производителя да бъде A.

Задача 5: Изтрийте от Ships всички кораби, които са потънали в битка.

Задача 6: Променете данните в релацията Classes така, че калибърът (bore) да се измерва в сантиметри (в момента е в инчове, 1 инч ~ 2.5 см) и водоизместимостта да се измерва в метрични тонове (1 м.т. = 1.1 т.)

Изгледи

Изгледи

- Виртуална таблица, която се дефинира чрез SELECT заявка.
- При създаването на изгледа се указва неговото име и чрез него след това може да се използва в произволни заявки, където се очаква име на таблица
- Резултатът от SELECT заявката, която дефинира изгледа **НЕ СЕ** съхранява физически и при всяко използване на изгледа, въпросната заявка се изпълнява наново от сървъра (освен ако не е създаден индекс върху изгледа; тогава имаме физическа материализация на изгледа и той се пази на диска)

Създаване на изглед

```
CREATE VIEW ShipGunsInfo
AS
SELECT s.NAME, c.BORE, c.NUMGUNS
FROM CLASSES c
    JOIN SHIPS s ON c.CLASS = s.CLASS
GO

SELECT * FROM ShipGunsInfo
```

- ShipGunsInfo е относително прост изглед, който дава информация за корабите в базата и техните оръдия, като свързва таблиците CLASSES и SHIPS
- Всяко създаване на изглед трябва да е в отделен BATCH (след CREATE VIEW трябва да има GO - маркерът за край на BATCH). След това изгледа може да се използва.

Модифициране на изглед

```
ALTER VIEW ShipGunsInfo
AS
SELECT s.NAME, s.LAUNCHED, c.BORE, c.NUMGUNS
FROM CLASSES c
    JOIN SHIPS s ON c.CLASS = s.CLASS
GO

SELECT * FROM ShipGunsInfo
```

- Променяме изгледа ShipGunsInfo чрез ALTER VIEW, като в SELECT заявката добавяме нова колона - s.LAUNCHED.
- За обикновени изгледи - това просто е обновяване на дефиницията на изгледа, която се пази в базата
- За индексирани изгледи (които са материализирани на диска) това ще доведе до изтриване на стария индекс и създаването му наново, заедно със съхраняването на резултата от новата заявка (т.е. операцията може да бъде доста ресурсоемка, ако реферираните таблици са големи)

Schema Binding

- Какво ще стане ако с ALTER TABLE се опитаме да премахнеме колоната LAUNCHED от таблицата SHIPS? ALTER TABLE заявката ще успее и изгледа ще престане да работи, защото няма да може да селектира тази колона
- Може да използваме SCHEMABINDING при създаването на изгледа и тогава сървъра няма да допуска модифициране на реферираните от изгледа обекти, което да води до нарушаване на работоспособността на изгледа.
- При използване на SCHEMABINDING трябва да се указва и името на схемата в която са създадени обектите (таблици или изгледи). Схемата по подразбиране в MSSQL Server се казва dbo.

```
ALTER VIEW ShipGunsInfo
WITH SCHEMABINDING
AS
SELECT s.NAME, s.LAUNCHED, c.BORE, c.NUMGUNS
FROM dbo.CLASSES c
    JOIN dbo.SHIPS s ON c.CLASS = s.CLASS
GO
```

Задачи

Задача 1: Дефинирайте изглед, наречен BritishShips, който за всеки британски кораб дава неговия клас, тип, брой оръдия, калибър, водоизместимост и годината, в която е пуснат на вода.

Задача 2: Напишете заявка, която използва изгледа от предната задача, за да покаже броя оръдия и водоизместимост на британските бойни кораби, пуснати на вода преди 1917.

Задача 3: Създайте изглед за всички потънали кораби по битки.

Задача 4: Създайте изглед с имената на битките, в които са участвали поне 3 кораба с под 9 оръдия и от тях поне един е бил увреден.