

Бази от данни

Упражнение 5:

Групиране и агрегация

Димитър Димитров

Агрегатни функции

- По даден списък от стойности изчисляват единична стойност
- Често използвани:
 - COUNT
 - SUM
 - AVG
 - MIN
 - MAX

Аргументи на агрегатните функции

- Най-често подаваме колона от таблица:

```
SELECT AVG(netWorth)  
FROM MovieExec;
```

- На COUNT можем да подадем *:

```
SELECT COUNT(*)  
FROM StarsIn;
```

- DISTINCT – премахва повторенията:

```
SELECT COUNT(DISTINCT starName)  
FROM StarsIn;
```

- Може да подадем и много по-сложен израз:

```
SELECT MAX(Quantity * UnitPrice * (1 - Discount))  
FROM "Order Details";
```

Комбиниране с други клаузи

- Можем да използваме WHERE, JOIN, ...

```
SELECT SUM(length)  
FROM Movie  
JOIN StarsIn  
ON title = movietitle AND year = movieyear  
WHERE starname = 'Harrison Ford';
```

Групиране (1)

- За всяко студио искаме да изведем точно по един ред със следната информация:

- Име на студиото
- Сумарна дължина на всички негови филми

```
SELECT studioName, SUM(length)  
FROM Movie  
GROUP BY studioName;
```

- GROUP BY vs. ORDER BY vs. DISTINCT

Алтернативно решение

- Ако в SQL нямаше групиране:

[illegible]

Групиране (2)

- Групиране по по-сложен израз:

За всяка година да се изведе колко филмови звезди са родени:

```
SELECT YEAR(birthdate), COUNT(*)  
FROM MovieStar  
GROUP BY YEAR(birthdate);
```

- Разбира се, горната заявка ще изведе само години, в които има родена поне една филмова звезда

- Групиране по няколко колони/израза:

За всяка година да се изведе колко актриси и колко актьори мъже са родени:

```
SELECT YEAR(birthdate), gender, COUNT(*)  
FROM MovieStar  
GROUP BY YEAR(birthdate), gender;
```

Правила при групиране (1)

- Не можем да използваме агрегатни функции в WHERE клауза!
- Да се изведе най-дългият филм (ако са повече от един, да се изведат всички най-дълги):

- Грешно:

```
SELECT *  
FROM movie  
WHERE length = MAX(length);
```

- Вярно:

```
SELECT *  
FROM movie  
WHERE length = (SELECT MAX(length)  
FROM movie);
```

- Често срещана грешка:

```
SELECT *  
FROM movie  
WHERE length = MAX(SELECT length  
FROM movie);
```


Най-важното правило

- Когато имаме GROUP BY, в SELECT може да използваме само:
 - Колоните (изразите), по които групираме
 - Агрегатни функции
 - Изрази, комбиниращи горните две
- Защото трябва да имаме само една стойност за цялата група
- Правилото важи и за ORDER BY

А ако нарушим правилото?

Disney	100
Fox	120
Fox	130

Disney	
Fox	

Disney	100
Fox	250

(error)

```
select studioname,  
length  
from movie;
```

```
select studioname,  
length  
from movie  
group by studioname;
```

```
select studioname,  
sum( length)  
from movie  
group by studioname;
```

HAVING клауза

- Филтрираме редовете, които се получават с GROUP BY

- Пример:

```
SELECT studioName, SUM(length)  
FROM Movie  
GROUP BY studioName  
HAVING COUNT(*) >= 2;
```

- Правилото важи и за HAVING
- HAVING се използва само с GROUP BY

Ред на клаузите в една SELECT заявка

- Според синтаксиса

SELECT ...
FROM ...
... JOIN ... ON ...
WHERE ...
GROUP BY ...
HAVING ...
ORDER BY ...

- Според реда на изпълнение

FROM ...
... JOIN ... ON ...
WHERE ...
GROUP BY ...
HAVING ...
SELECT ...
ORDER BY ...

- Това обяснява всичко

Пример

- Преди не подозирахме, че следната заявка е грешна:

```
SELECT title AS movietitle  
FROM movie  
WHERE movietitle LIKE 'Star %';
```

NULL стойности

- NULL стойностите се игнорират при агрегиране
 - Разлика между COUNT(*) и COUNT(A)
- NULL стойностите НЕ се игнорират при групиране
 - Могат да образуват група

NULL - типичен пример

- За всяка филмова звезда да се изведе броят на филмите, в които се е снимала.
- Ако за дадена звезда не знаем какви филми има, за нея да се изведе 0.
- **SELECT name, COUNT(starname) as moviesCount
FROM MovieStar
LEFT JOIN StarsIn ON name = starname
GROUP BY name;**
- В примера на COUNT подаваме колона от дясната таблица, която не може да приема NULL стойности

NULL – тънка особеност

- **SELECT ***
FROM movie
WHERE length = (SELECT MAX(length) FROM movie);
- Горната заявка намира най-дългите филми измежду тези, за които дължината е известна
- **SELECT ***
FROM movie
WHERE length >= ALL (SELECT length FROM movie);
- Заявката не връща нищо, ако има дори един филм с неизвестна дължина

Труден пример

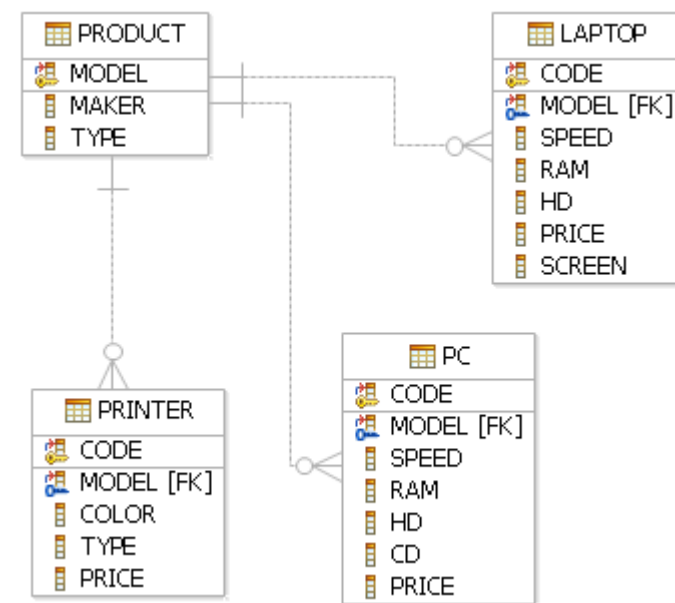
- Да се изведе средният брой филми, в които са се снимали актьорите
- Грешно: `SELECT AVG(COUNT(...`
- Вярно:
- **`SELECT AVG(moviescount)`**
-- или: `select avg(convert(real, moviescount))`
`FROM (SELECT COUNT(movietitle) AS moviescount`
`FROM MovieStar`
`LEFT JOIN StarsIn ON name = starname`
`GROUP BY name) stat;`

Въпроси?

Следват задачи

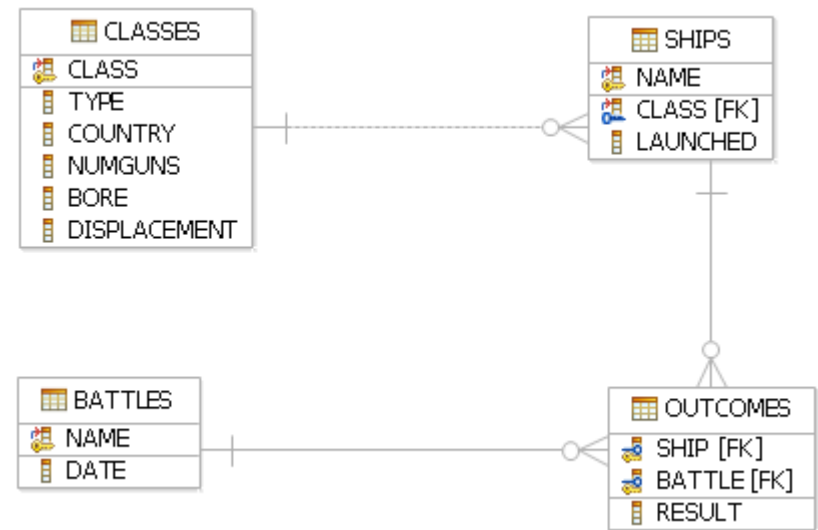
1. Задачи - PC

1. Напишете заявка, която извежда средната честота на процесорите на компютрите.
2. Напишете заявка, която за всеки производител извежда средния размер на екраните на неговите лаптопи.
3. Напишете заявка, която извежда средната честота на лаптопите с цена над 1000.
4. Напишете заявка, която извежда средната цена на компютрите, произведени от производител 'A'.
5. Напишете заявка, която извежда средната цена на компютрите и лаптопите на производител 'B' (едно число).
6. Напишете заявка, която извежда средната цена на компютрите според различните им честоти на процесорите.
7. Напишете заявка, която извежда производителите, които са произвели поне по 3 различни модела компютъра.
8. Напишете заявка, която извежда производителите на компютрите с най-висока цена.
9. Напишете заявка, която извежда средната цена на компютрите за всяка честота, по-голяма от 800 MHz.
10. Напишете заявка, която извежда средния размер на диска на тези компютри, произведени от производители, които произвеждат и принтери.
11. Напишете заявка, която за всеки размер на лаптоп намира разликата в цената на най-скъпия и най-евтиния лаптоп със същия размер.



2. Задачи - Ships

1. Напишете заявка, която извежда броя на класовете кораби.
2. Напишете заявка, която извежда средния брой на оръдията (numguns) за всички кораби, пуснати на вода (т.е. изброени са в таблицата Ships).
3. Напишете заявка, която извежда за всеки клас първата и последната година, в която кораб от съответния клас е пуснат на вода.
4. Напишете заявка, която за всеки клас извежда броя на корабите, потънали в битка.
5. Напишете заявка, която за всеки клас с над 4 пуснати на вода кораба извежда броя на корабите, потънали в битка.
6. Напишете заявка, която извежда средното тегло на корабите (displacement) за всяка страна.



3. Още задачи

- За всеки актьор/актриса изведете броя на различните студиа, с които са записвали филми. Включете и тези, за които няма информация в кои филми са играли.
- Изведете имената на актьорите, участвали в поне 3 филма след 1990 г.
- Да се изведат различните модели компютри, подредени по цена на най-скъпия конкретен компютър от даден модел.
- Намерете броя на потъналите американски кораби за всяка проведена битка с поне един потънал американски кораб.
- Битките, в които са участвали поне 3 кораба на една и съща страна.
- Имената на класовете, за които няма кораб, пуснат на вода след 1921 г., но имат пуснат поне един кораб.
- (*) За всеки кораб броя на битките, в които е бил увреден (result = 'damaged'). Ако корабът не е участвал в битки или пък никога не е бил увреждан, в резултата да се вписва 0.
- (*) Намерете за всеки клас с поне 3 кораба броя на корабите от този клас, които са победили в битка (result = 'ok')
- За всяка битка да се изведе името на битката, годината на битката и броят на потъналите кораби, броят на повредените кораби и броят на корабите без промяна.
- (*) Намерете имената на битките, в които са участвали поне 3 кораба с под 9 оръдия и от тях поне два са с резултат 'ok'.

4. Общи задачи върху материала до момента

- Разгледайте документа с общи задачи