

Свързвания

Упражнение 3

CROSS JOIN

MovieExec

Name	PrCert#
George Lucas	111
Ted Turner	222

Movie

Title	Year	PrCert#
Star Wars	1977	111
Star Trek	1979	555

`SELECT * FROM MOVIEEXEC, MOVIE`

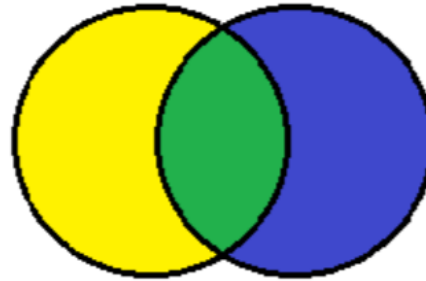
`SELECT * FROM MOVIEEXEC CROSS JOIN MOVIE`

Name	PrCert#	Title	Year	PrCert#
George Lucas	111	Star Wars	1977	111
George Lucas	111	Star Trek	1979	555
Ted Turner	222	Star Wars	1977	111
Ted Turner	222	Star Trek	1979	555

INNER JOIN

MovieExec

Name	PrCert#
George Lucas	111
Ted Turner	222



Movie

Title	Year	PrCert#
Star Wars	1977	111
Star Trek	1979	555

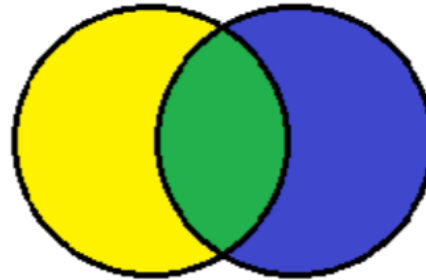
```
SELECT *  
FROM MOVIEEXEC me  
JOIN MOVIE m ON me.PrCert# = m.PrCert#
```

Name	PrCert#	Title	Year	PrCert#
George Lucas	111	Star Wars	1977	111

LEFT [OUTER] JOIN

MovieExec

Name	PrCert#
George Lucas	111
Ted Turner	222



Movie

Title	Year	PrCert#
Star Wars	1977	111
Star Trek	1979	555

```
SELECT *  
FROM MOVIEEXEC me  
LEFT OUTER JOIN MOVIE m ON me.PrCert# = m.PrCert#
```

Name	PrCert#	Title	Year	PrCert#
George Lucas	111	Star Wars	1977	111
Ted Turner	222	NULL	NULL	NULL

Задача

За всяка държава да се изведат имената на корабите, които никога не са участвали в битка

- ❑ Държавата се намира в таблицата CLASSES, битките са в OUTCOMES - нужни са ни и трите таблици CLASSES, SHIPS, OUTCOMES
- ❑ Свързването между CLASSES и SHIPS е вътрешно, понеже не се интересуваме от класове/държави, за които няма кораби
- ❑ Свързването между SHIPS и OUTCOMES е ляво (външно), защото корабите, които не са участвали в битка са точно корабите от SHIPS, за които не съществува съответен ред в таблицата OUTCOMES. След външното свързване - за тях ще има стойности NULL в колоните от OUTCOMES

```
SELECT c.COUNTRY, s.NAME
FROM CLASSES c
      JOIN SHIPS s ON c.CLASS = s.CLASS
      LEFT OUTER JOIN OUTCOMES o ON s.NAME = o.SHIP
WHERE o.SHIP IS NULL
```

Задача

За всеки клас, да се изведат името му, държавата и имената на всички негови кораби, пуснати през 1916г. В резултата да участват и класовете без кораби пуснати през 1916г.

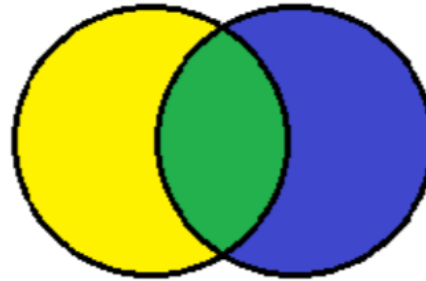
- ❑ Вътрешно свързване на CLASSES с SHIPS и последваща филтрация по LAUNCHED = 1916 би ни дало таблица съдържаща класовете и корабите пуснати на вода през 1916г.
- ❑ Ако направиме свързването ляво и външно с условие за равенство по име на клас (CLASSES.CLASS = SHIPS.CLASS) - ред от CLASSES ще се свърже с ред от SHIPS за всички кораби от съответния клас, без значение от годината на пускане на вода.
- ❑ Нас ни интересуват само случаите, в които кораба е пуснат на вода през 1916. Ако добавиме това условие при свързването - тогава за кораби, които не са пуснати през 1916г. ще получиме нужните стойности NULL за колоните отдясно. NULL стойности ще имаме и за класовете, които нямат никакви кораби (те също би следвало да участват в крайния резултат)

```
SELECT c.CLASS, c.COUNTRY, s.NAME  
FROM CLASSES c  
LEFT OUTER JOIN SHIPS s ON c.CLASS = s.CLASS AND s.LAUNCHED = 1916
```

RIGHT [OUTER] JOIN

MovieExec

Name	PrCert#
George Lucas	111
Ted Turner	222



Movie

Title	Year	PrCert#
Star Wars	1977	111
Star Trek	1979	555

```
SELECT *  
FROM MOVIEEXEC me  
      RIGHT OUTER JOIN MOVIE m ON me.PrCert# = m.PrCert#
```

Name	PrCert#	Title	Year	PrCert#
George Lucas	111	Star Wars	1977	111
NULL	NULL	Star Trek	1979	555

Задача

За всеки клас британски кораби - да се извлече името на класа и имената на всички битки, в които са участвали кораби от този клас. Ако за даден клас няма кораби или те не са участвали в битка - да се включат със стойност NULL за името на битката

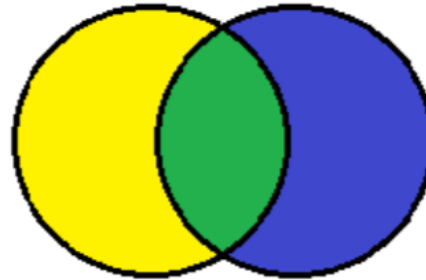
- Ако използваме ляво свързване на CLASSES със SHIPS - успешно ще включим класовете, които нямат кораби. След това с резултата може да свържем OUTCOMES отново чрез ляво външно свързване и ще имаме NULL стойности в колоните от OUTCOMES, които съответстват на ред от SHIPS за който няма съответни битки в OUTCOMES (кораби, които не са участвали в битка).
- Ако в един клас има няколко кораба и някой от тях са участвали в битка а други не - в резултата от горните свързвания ще получим за даден клас както ред от вида (име на клас, име на кораб участвал в битка), така и ред (име на клас, NULL стойност за име на кораб). Редове от втория вид искаме само в случаите, които за класа няма кораб отговарящ на условията.
- Ако свържем таблиците в обратен ред, започвайки първо с OUTCOMES и SHIPS с вътрешно свързване - ще имаме само редове от вида (име на битка, име на участвал кораб). При последващо дясно външно свързване с CLASSES - NULL стойности ще имаме само за класове, които или нямат кораби въобще или нямат кораби участвали в битка

```
SELECT *  
FROM OUTCOMES o  
      JOIN SHIPS s ON o.SHIP = s.NAME  
      RIGHT OUTER JOIN CLASSES c ON s.CLASS = c.CLASS  
WHERE c.COUNTRY = 'Gt.Britain'
```


FULL [OUTER] JOIN

MovieExec

Name	PrCert#
George Lucas	111
Ted Turner	222



Movie

Title	Year	PrCert#
Star Wars	1977	111
Star Trek	1979	555

```
SELECT *  
FROM MOVIEEXEC me  
      FULL OUTER JOIN MOVIE m ON me.PrCert# = m.PrCert#
```

Name	PrCert#	Title	Year	PrCert#
George Lucas	111	Star Wars	1977	111
Ted Turner	222	NULL	NULL	NULL
NULL	NULL	Star Trek	1979	555

FULL OUTER JOIN

От базата MOVIES, да се извлече таблица с три колони:

- ❑ Име (на актьор или продуцент)
- ❑ Дата на раждане (да съдържа датата на раждане на актьора, NULL за продуцент)
- ❑ Богатство (стойността от Networth колоната за продуцент, NULL за актьор)

```
SELECT COALESCE(ms.NAME, me.NAME) AS NAME,  
       ms.BIRTHDATE,  
       me.NETWORTH  
FROM MOVIESTAR ms  
FULL OUTER JOIN MOVIEEXEC me ON ms.NAME = me.NAME
```

NAME	NAME	BIRTHDATE	NETWORTH
Alec Baldwin	Alec Baldwin	1977-07-06 00:00:00.000	150000000
Debra Winger	NULL	1978-06-05 00:00:00.000	NULL
Harrison Ford	NULL	1955-05-05 00:00:00.000	NULL
Jack Nicholson	NULL	1949-05-05 00:00:00.000	NULL
Jane Fonda	NULL	1977-07-07 00:00:00.000	NULL
Kim Basinger	NULL	1979-07-05 00:00:00.000	NULL
Sandra Bullock	NULL	1948-12-05 00:00:00.000	NULL
NULL	Calvin Coolidge	NULL	20000000
NULL	Merv Griffin	NULL	112000000
NULL	Stephen Spielberg	NULL	100000000
NULL	Ted Turner	NULL	125000000
NULL	George Lucas	NULL	200000000

NAME	BIRTHDATE	NETWORTH
Alec Baldwin	1977-07-06 00:00:00.000	150000000
Debra Winger	1978-06-05 00:00:00.000	NULL
Harrison Ford	1955-05-05 00:00:00.000	NULL
Jack Nicholson	1949-05-05 00:00:00.000	NULL
Jane Fonda	1977-07-07 00:00:00.000	NULL
Kim Basinger	1979-07-05 00:00:00.000	NULL
Sandra Bullock	1948-12-05 00:00:00.000	NULL
Calvin Coolidge	NULL	20000000
Merv Griffin	NULL	112000000
Stephen Spielberg	NULL	100000000
Ted Turner	NULL	125000000
George Lucas	NULL	200000000

Пример с по-сложни условия за свързване

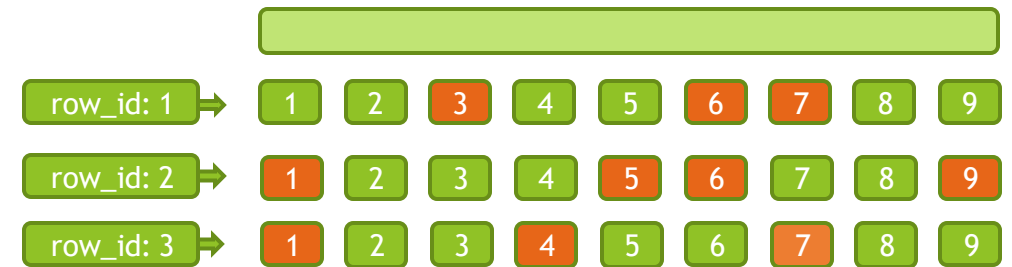
Искаме да извлечеме двойките (начално място, крайно място) за последователни свободни места на даден ред

Представяне (Seats)

row_id	seat_id	is_free
1	1	1
1	2	1
1	3	0
1	4	1
1	5	1
1	6	0
1	7	0
1	8	1
1	9	1
2	1	0
2	2	1
2	3	1
2	4	1
2	5	0
2	6	0
2	7	1
2	8	1
2	9	0
3	1	0
3	2	1
3	3	1
3	4	0
3	5	1
3	6	1
3	7	0
3	8	1
3	9	1

Търсен резултат

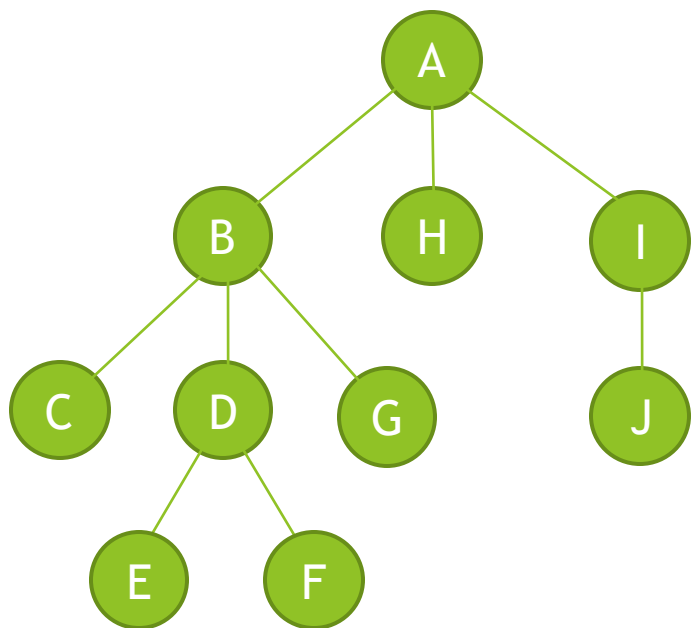
row_id	l_id	r_id
1	1	2
1	4	5
1	8	9
2	2	4
2	7	8
3	2	3
3	5	6
3	8	9



USE sandbox

```
SELECT sr.row_id, sl.seat_id l_id, sr.seat_id r_id
FROM Seats sb
      RIGHT OUTER JOIN Seats sl ON sb.seat_id = sl.seat_id - 1 AND sl.row_id = sb.row_id
      JOIN Seats sr ON sl.seat_id < sr.seat_id AND sl.row_id = sr.row_id
      LEFT OUTER JOIN Seats sa ON sr.seat_id + 1 = sa.seat_id AND sr.row_id = sa.row_id
WHERE 1 = ALL (SELECT is_free
                FROM Seats
                WHERE seat_id BETWEEN sl.seat_id AND sr.seat_id
                  AND sl.row_id = sr.row_id AND row_id = sl.row_id)
AND (sa.is_free = 0 OR sa.is_free IS NULL)
AND (sb.is_free = 0 OR sb.is_free IS NULL)
```

Дървета - Adjacency List Model



Node	Parent
A	NULL
B	A
C	B
D	B
E	D
F	D
G	B
H	A
I	A
J	I

Adjacency List Model - пример

Employees

ID	Name	ManagerID
1	James Smith	NULL
2	Jane Adams	1
3	Maria Alandes	1
4	Philip Anderson	3
5	Anton Weber	3

Да се извлече таблица съдържаща имената на мениджъри и техните подчинени (всеки ред е двойка (мениджър, подчинен))

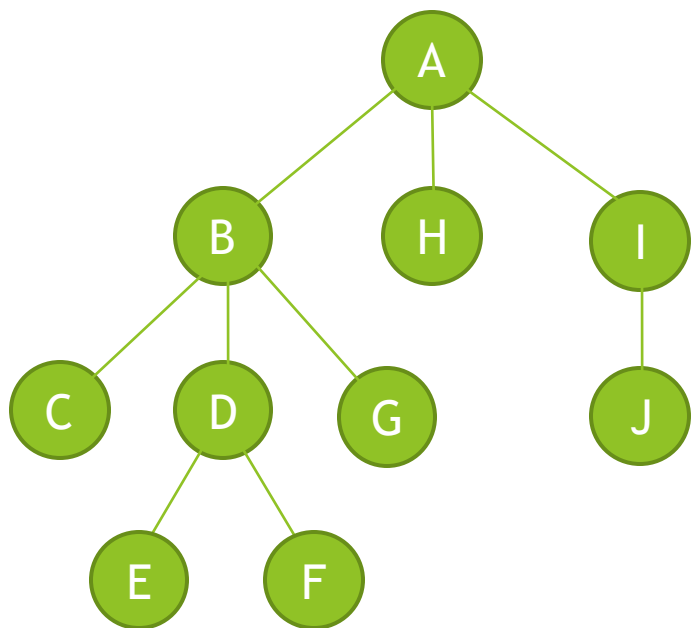
```
SELECT m.Name Manager, s.Name Subordinate
FROM Employees m
      JOIN Employees s ON m.ID = s.ManagerID
```

Manager	Subordinate
James Smith	Jane Adams
James Smith	Maria Alandes
Jane Adams	NULL
Maria Alandes	Philip Anderson
Maria Alandes	Anton Weber
Philip Anderson	NULL
Anton Weber	NULL

Ако искаме да включим и служителите, които нямат подчинени в резултата:

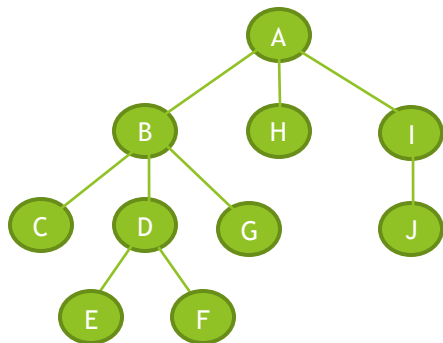
```
SELECT m.Name Manager, s.Name Subordinate
FROM Employees m
      LEFT OUTER JOIN Employees s ON m.ID = s.ManagerID
```

Дървета - Nested Set Model



Node	L	R
A	1	20
B	2	13
C	3	4
D	5	10
E	6	7
F	8	9
G	11	12
H	14	15
I	16	19
J	17	18

Дървета - Nested Set Model



Node	L	R
A	1	20
B	2	13
C	3	4
D	5	10
E	6	7
F	8	9
G	11	12
H	14	15
I	16	19
J	17	18

Да се извлече таблица съдържаща имената на възли и техните деца (всеки ред е двойка (родител, дете))

```
SELECT Parent.Node AS Parent, Child.Node AS Child
FROM Tree AS Parent
      LEFT OUTER JOIN Tree AS Child ON Child.L > Parent.L AND Child.L < Parent.R
WHERE NOT EXISTS (SELECT *
                  FROM Tree AS Middle
                  WHERE Middle.L > Parent.L AND Middle.L < Parent.R
                    AND Child.L > Middle.L AND Child.L < Middle.R)
ORDER BY Parent.Node
```