Linking with union

If two select blocks are combined with the union operator, the end result consists of the resulting rows from either or both of the select blocks.

Give the number of each player who has incurred at least on penalty, or who is a captain or for whom both conditions apply:

```
Select pe_p_playerno
From pe_penalties
Union
Select t_p_playerno
From t_teams
Order by pe_p_playerno desc
```

All duplicate rows are automatically removed from the end result. Use of distinct is redundant.

It is possible to join more than two select blocks.

Give the number of each player who has incurred at least one penalty, who is captain, who lives in Stratford or for whom two or three of these conditions apply:

```
Select pe_p_playerno
From pe_penalties
Union
Select t_p_playerno
From t_teams
Union
Select p_playerno
From p_players
Where p_town='stratford'
Order by pe_p_playerno desc
```

Rules for using UNION

The select clause of all relevant select blocks must have the same number of expressions.

Expressions which will be combined (or placed under one another) in the end result must have comparable data types.

An order by clause may only specified after the last select block. The ordering is performed on the entire end result, only after all intermediate results have been combined.

Linking with intersect (Durchschnitt)

In mysql as inner join.

Give the number and the year of birth of each player living in Stratford and born after 1960.

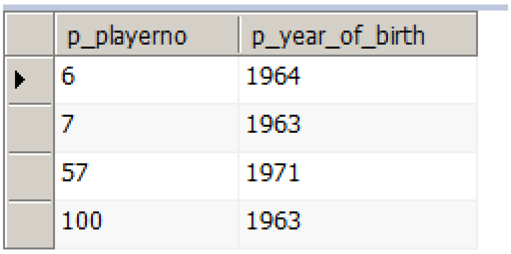
Oracle:

```
Select p_playerno, p_year_of_birth
From p_players
Where p_town=’Stratford’
Intersect
Select p_playerno, p_year_of_birth
From p_players
Where p_year_of_birth > 1960
```

You can also use the **and** operator for this statement.

Mysql:

```
Select ap.p_playerno, ap.p_year_of_birth
From p_players ap join p_players bp
Where ap.p_playerno=bp.p_playerno
And ap.p_town='Stratford'
And bp.p_year_of_birth > 1960
```



p_playerno	p_year_of_birth
6	1964
7	1963
57	1971
100	1963

It is not always possible to substitute the intersect operator for the and operator.

Give the number of each player who is a captain and who has incurred at least one penalty:

Oracle:

```
Select t_p_playerno
From t_teams
Intersect
Select p_playerno
From pe_penalties
```

Mysql:

```
Select distinct t_p_playerno
From t_teams a join pe_penalties b
Where a.t_p_playerno = b.pe_p_playerno
```

Linking with minus (differenz)

If two blocks are combined with the minus operator, the end result consists only of the resulting rows appearing in the result of the first select block, but which do not appear in the result of the second select block.

Mysql: outer left join.

Give the number and the year of birth of each player who lives in Stratford, and was not born after 1960.

Oracle:

```
Select p_playerno, p_year_of_birth
From p_players
Where p_town='Stratford'
Minus
Select p_playerno, p_year_of_birth
From p_players
Where p_year_of_birth > 1960
```

Mysql:

```
Select b.p_playerno, b.p_year_of_birth
From p_players a left join p_players b
                using (p_playerno, p_year_of_birth)
Where a.p_playerno=b.p_playerno and
a.p_town='Stratford' and b.p_year_of_birth < 1960
```

using … useful if both table have the same column names

or like the following statement:

```
Select p_playerno, p_year_of_birth
From p_players
Where p_town='Stratford' and not(p_year_of_birth > 1960)
```

Give the number of each player who incurred at least on penalty and is not a captain:

Oracle:

```
Select pe_p_playerno
From pe_penalties
Minus
Select t_p_playerno
From t_teams
```

Mysql:

```
Select distinct pe_p_playerno, t_p_playerno
From pe_penalties left join t_teams
on pe_p_playerno=t_p_playerno
where t_p_playerno IS null
```