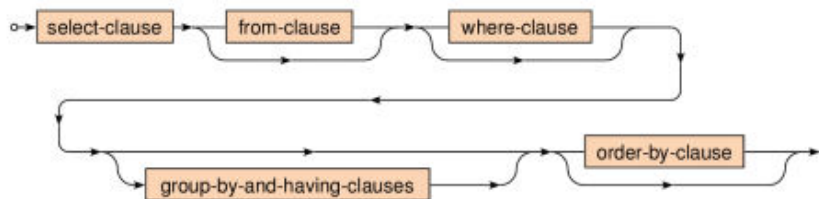


# SQL: Data Querying (4)

## SQL: Select Queries

- skládají se z 1-5 klauzulí (clauses) a volitelně také z ORDER BY klauzule
- SELECT – které sloupce mají být ve výsledné tabulce
- FROM – která tabulka by měla být zdrojem dat pro dotazování
- WHERE – podmínka, kterou musí řádka splnit
- GROUP BY – jaké atributy by měly být použity pro agregaci
- HAVING – podmínka, kterou musí agregovaná řádka splnit
- ORDER BY – atributy, které jsou použity pro řazení řádků výsledné tabulky



## Database of flights and aircrafts

Flights:

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

Aircrafts:

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

- Which aircrafts can be used for the scheduled flights?
  - Only aircrafts of a given company and sufficient capacity can be used

```
SELECT Flights.*, Aircraft
FROM Flights NATURAL JOIN Aircrafts
WHERE (Passengers <= Capacity)
ORDER BY Flight
```

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253

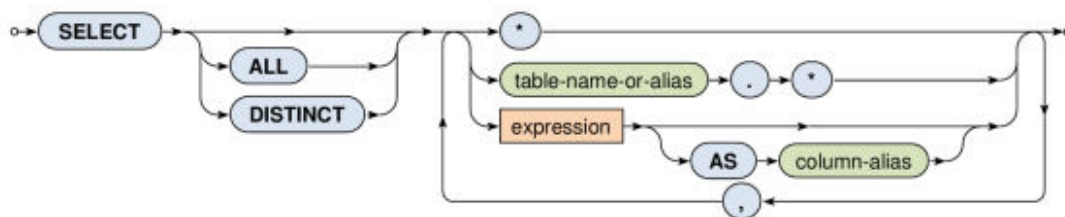
Flight	Company	Destination	Passengers	Aircraft
KL1245	KLM	Amsterdam	130	Airbus A380
KL1245	KLM	Amsterdam	130	Airbus A350
KL7621	KLM	Rotterdam	75	Airbus A380
KL7621	KLM	Rotterdam	75	Airbus A350
OK012	CSA	Milano	37	Boeing 717



Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130

## SELECT Clause

- **SELECT ...** FROM ... WHERE ... ORDER BY ...
- seznam sloupců, které mají být ve výsledku
- projekce vstupních sloupců:
  - o jméno sloupce
  - o \* (všechny sloupce), **table.\*** (všechny sloupce z dané tabulky)
- definice nových, odvozených či agregovaných sloupců
  - o použití výrazů založených na literálech, funkcích, subdotazech,...
- sloupcům také mohou být přiřazena nová jména použitím **AS**



- **modifikátory výstupu:**
  - o **ALL** (výchozí) – všechny řádky jsou zahrnuty do výstupu
  - o **DISTINCT** – duplicity jsou odstraněny

## Examples

- **SELECT ALL \*** ...
- **SELECT Flights.\*, Aircraft** ...
- **SELECT DISTINCT Company AS Carrier** ...
- **SELECT ((3\*5) + 5) AS MyNumber, 'Hello' AS MyString** ...
- **SELECT SUM(Capacity)** ...
- **SELECT (SELECT COUNT(\*) FROM Table) AS Result** ...

## WHERE Clause

- SELECT ... FROM ... **WHERE ...** ORDER BY ...
- **podmínka výběru:**
  - o podmínka, kterou musí řádka splnit, aby se dostala do výsledku
  - o jednoduché výrazy se mohou kombinovat pomocí spojek (AND, OR, NOT)



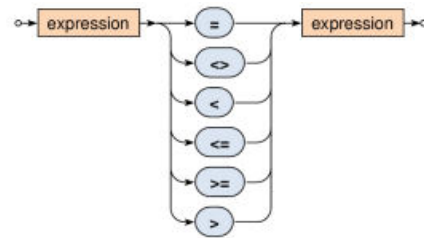
## Examples

- ... **WHERE (Capacity > 200) AND (Aircraft LIKE 'Airbus%')** ...
- ... **WHERE (Company IN ('KLM', 'Emirates'))** ...
- ... **WHERE NOT (Passengers BETWEEN 100 AND 200)** ...

## Podmínky vyhledávání

### - srovnávací predikáty:

- standardní porovnávání
- funguje i pro tuply
  - např. (1,2,3) <= (1,2,5)



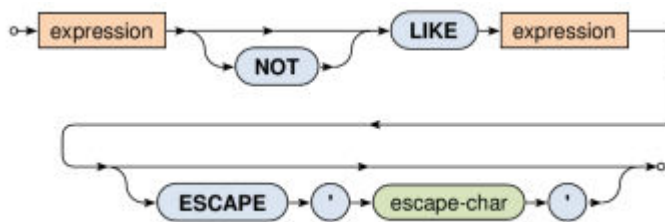
### - intervalové predikáty:

- hodnota BETWEEN Min AND Max
- ekvivalentní s (Min <= Value) AND (Value <= Max)



### - řetězec odpovídající predikátu:

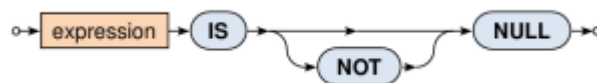
- testuje, jestli string hodnota odpovídá danému vzoru
- vzor může obsahovat speciální znaky
  - % ... zastupuje libovolný podřetězec (i prázdný)
  - \_ ... zastupuje libovolný znak
- případně i escape znaky



— Company LIKE '%Airlines%'

### - predikát pro detekci NULL hodnot:

- testuje, jestli je/není daná hodnota NULL
- nelze použít „expression = NULL“



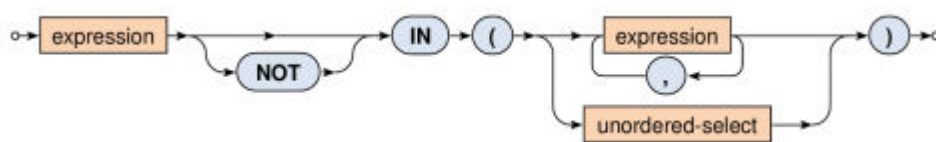
### ○ dopad NULL hodnot:

- byly představeny pro zvládnutí chybějící informace
- když nemůže být **funkce** (či operátor) vyhodnocena, vrátí **NULL**
  - např. 3 + NULL je vyhodnoceno jako NULL
- když nemůže být vyhodnocen **predikát**, vrátí to logický **UNKNOWN**
  - např. 3 < NULL vrátí UNKNOWN
  - potřebujeme tedy pracovat s trojhodnotovou logikou (T, F, U)

p	q	p AND q	p OR q	NOT q
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	UNKNOWN	UNKNOWN	TRUE	UNKNOWN
FALSE	TRUE	FALSE	TRUE	
FALSE	FALSE	FALSE	FALSE	
FALSE	UNKNOWN	FALSE	UNKNOWN	
UNKNOWN	TRUE	UNKNOWN	TRUE	
UNKNOWN	FALSE	FALSE	UNKNOWN	
UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	

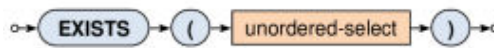
- **set membership predicate:**

- testuje, existuje-li hodnota v dané množině hodnot
- např.: *Company IN ('KLM', 'Emirates')*
- ... IN ( $\emptyset$ ) = FALSE ... prázdná tabulka
- ... IN ( $\aleph$ ) = UNKNOWN ... tabulka mající řádky jen s NULL hodnotami



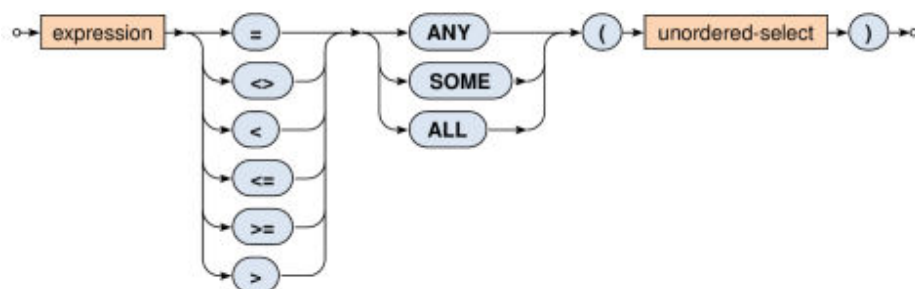
- **existential quantifier predicate:**

- testuje, není-li daná množina prázdná
- může být použito pro simulaci univerzálního kvantifikátoru
- $\forall$  odpovídá  $\neg\exists$
- EXISTS ( $\emptyset$ ) = FALSE
- EXISTS ( $\aleph$ ) = TRUE



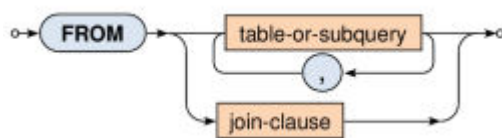
- **predikáty pro porovnávání množin:**

- **ALL**
  - všechny řádky z vnořeného dotazu musí odpovídat operátoru
  - ALL ( $\emptyset$ ) = TRUE
  - ALL ( $\aleph$ ) = FALSE
- **ANY a SOME (synonyma)**
  - alespoň jedna řádka z vnořeného dotazu musí odpovídat danému operátoru
  - ANY ( $\emptyset$ ) = FALSE
  - ANY ( $\aleph$ ) = UNKNOWN



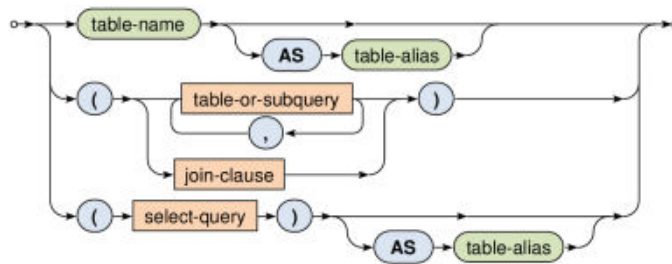
## FROM Clause

- SELECT ... **FROM** ... WHERE ... ORDER BY ...
- popis tabulek pro dotazování (resp. i vložených dotazů či pohledů)
- **starý způsob:**
  - o čárkou rozdělený seznam tabulek
  - o kartézský součin jejich řádků
  - o vyžaduje, aby join-podmínky byly specifikovány ve WHERE klauzuli
  - o např.: `SELECT ... FROM Flights, Aircrafts WHERE ...`
- **nový způsob:**
  - o užití JOIN operátorů s volitelnými podmínkami
  - o např.: `SELECT ... FROM Flights JOIN Aircrafts WHERE ...`



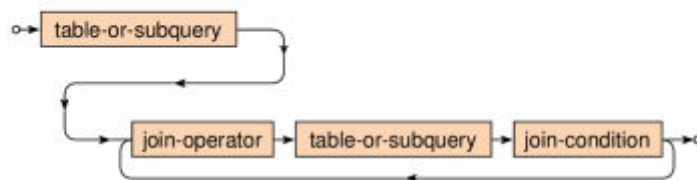
## Tables and subqueries

- Table name, auxiliary parentheses, direct select statement



## Table Joins

### Basic structure of joins



- Examples
  - » `Flights NATURAL JOIN Aircrafts`
  - » `Flights JOIN Aircrafts USING (Company)`
  - » ...



## Cross JOIN

- kartézský součin všech řádek z obou tabulek



**SELECT \* FROM T1 CROSS JOIN T2**

A	T1.*	A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...	1	...	⇒	1	...	1	...
2	...	4	...		1	...	4	...
3	...				2	...	1	...
					2	...	4	...
					3	...	1	...
					3	...	4	...

## Natural JOIN

- dvojice řádků jsou kombinovány, jen pokud mají stejné hodnoty ve všech sdílených sloupcích
  - o tj. sloupcích stejného jména

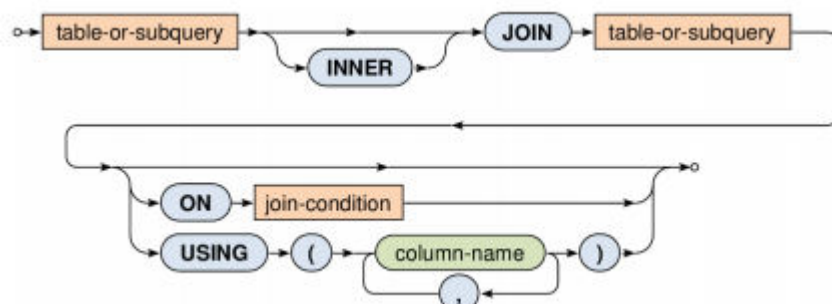


**SELECT \* FROM T1 NATURAL JOIN T2**

A	T1.*		A	T2.*	⇒	A	T1.*	T2.*
1	...		1	...		1	...	...
2	...		4	...				
3	...							

## Inner JOIN

- podmnožina cross joinu
- dvojice hodnot jsou zkombinovány, jen pokud:
  - o **ON** – splňují danou JOIN podmínku
  - o **USING** – mají stejné hodnoty v daných sloupcích



**SELECT \* FROM T1 JOIN T2 ON (T1.A <= T2.A)**

A	T1.*	A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...	1	...	⇒	1	...	1	...
2	...	4	...		1	...	4	...
3	...				2	...	4	...
					3	...	4	...

**SELECT \* FROM T1 JOIN T2 USING (A)**

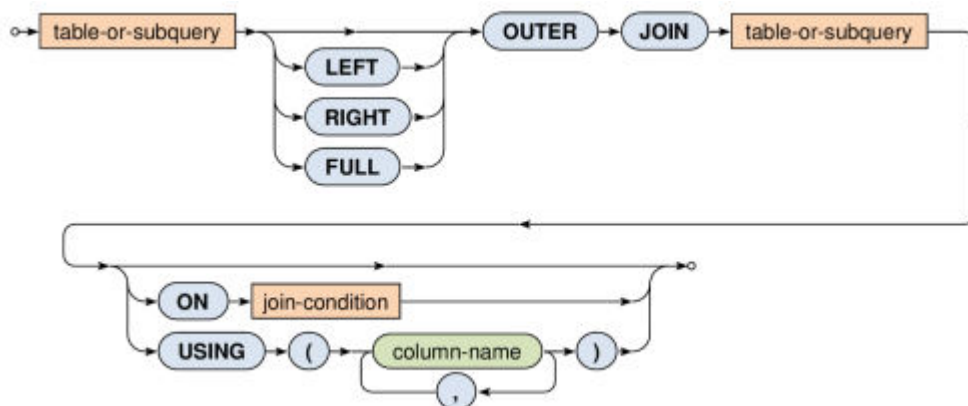
- Equals to the corresponding natural join

**SELECT \* FROM T1 JOIN T2**

- Equals to the corresponding cross join

### Outer JOIN

- dvojice řádků ze standardního inner joinu + řádky, které nemohou být zkombinovány, zvláště
  - **LEFT/RIGHT** – řádky jen z levé/pravé tabulky
  - **FULL** (default) – řádky z obou tabulek



- NULL hodnoty vyplňují chybějící informace v řádcích, které nemohly být zkombinovány

**SELECT \***

**FROM T1 LEFT OUTER JOIN T2 ON (T1.A = T2.A)**

A	T1.*	A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...	1	...	⇒	1	...	1	...
2	...	4	...		2	...	NULL	NULL
3	...				3	...	NULL	NULL

### Union JOIN

- řádky obou tabulek jsou integrovány do jedné, žádné dvojice řádků kombinovány nejsou

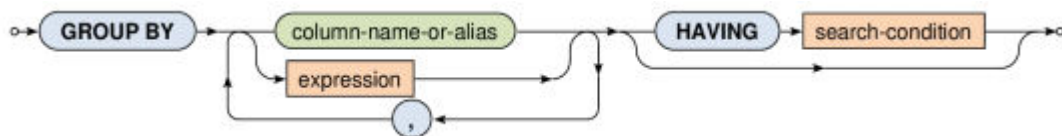


**SELECT \* FROM T1 UNION JOIN T2**

A	T1.*	A	T2.*		T1.A	T1.*	T2.A	T2.*
1	...	1	...	→	1	...	NULL	NULL
2	...	4	...		2	...	NULL	NULL
3	...				3	...	NULL	NULL
					NULL	NULL	1	...
					NULL	NULL	4	...

## Agregace

- základní idea:
  - o nejprve...
    - **FROM** a **WHERE** klauzule jsou vyhodnoceny standardní cestou
    - což vede k mezi-tabulce
  - o pak...
    - **GROUP BY** - řádky tabulky jsou rozděleny do skupin dle stejných hodnot v daných sloupcích
    - **HAVING** - tyto agregované řádky (superřádky) mohou být filtrovány užitím dané vyhledávací podmínky



- How many flights does each company have scheduled?
  - However, we are not interested in flights to Stuttgart and Munich
  - As well as we do not want companies with just one flight or less

**SELECT** Company, COUNT(\*) AS Flights **FROM** Flights  
**WHERE** (Destination NOT IN ('Stuttgart', 'Munich'))  
**GROUP BY** Company **HAVING** (Flights > 1)

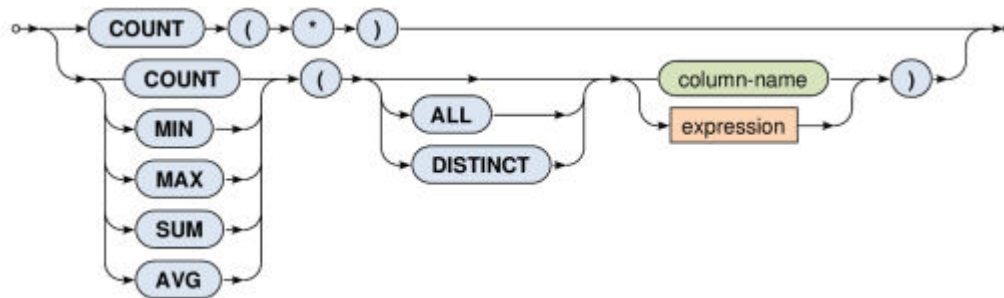
Flight	Company	Destination	Passengers	⇒	Flight	Company	Destination	Passengers	⇒	Company	Flights
OK251	CSA	New York	276		OK251	CSA	New York	276		CSA	3
LH438	Lufthansa	Stuttgart	68		OK012		Milano	37		Air Canada	1
OK012	CSA	Milano	37		OK321		London	156		KLM	2
OK321	CSA	London	156		AC906	Air Canada	Toronto	116			
AC906	Air Canada	Toronto	116		KL7621	KLM	Rotterdam	75			
KL7621	KLM	Rotterdam	75		KL1245		Amsterdam	130			
KL1245	KLM	Amsterdam	130								

Company	Flights
CSA	3
KLM	2



- **Jaké sloupce mohou být použity v SELECT a HAVING při agregaci? Jen...**
  - o agregační sloupce (tzn. ty z GROUP BY klauzule)
  - o sloupce nově odvozené pomocí agregačních funkcí



## Agregační funkce

- umožňují vytvořit hodnoty ze skupiny řádků
- **COUNT(\*)**
  - o počet všech řádků včetně duplicit a NULL hodnot
- **COUNT / SUM / AVG / MIN / MAX**
  - o počet hodnot / součet hodnot / průměr / minimum / maximum
  - o NULL hodnoty jsou vždy automaticky ignorovány
  - o modifikátor ALL (defaultní) zahrnuje duplicity, DISTINCT ne
  - o COUNT( $\emptyset$ ) = 0
  - o SUM( $\emptyset$ ) = NULL
  - o AVG( $\emptyset$ ) = NULL, MIN( $\emptyset$ ) = NULL, MAX( $\emptyset$ ) = NULL

## Aggregations: Example

- Find basic characteristics for all the scheduled flights
  - I.e. return the overall number of flights, the overall number of the involved companies, the sum of all the passengers, the average / minimal / maximal number of passengers

### SELECT

```

COUNT(*) AS Flights,
COUNT(DISTINCT Company) AS Companies,
SUM(Passengers) AS PSum,
AVG(Passengers) AS PAvg,
MIN(Passengers) AS PMin,
MAX(Passengers) AS PMax

```

FROM Flights

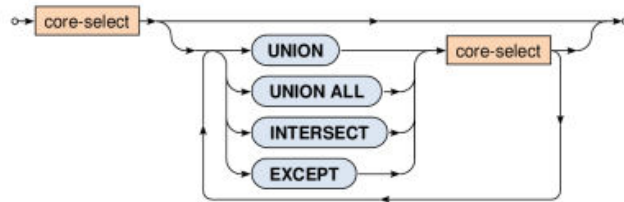
Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130



Flights	Companies	PSum	PAvg	PMin	PMax
7	4	858	123	37	276

## Množinové operace

- **UNION** – sjednocení dvou tabulek (bez duplicit)
- **UNION ALL** – sjednocení dvou tabulek (s duplicitami)
- **INTERSECT** – průnik dvou tabulek
- **EXCEPT** – rozdíl mezi tabulkami



Merge available companies from tables of flights and aircrafts

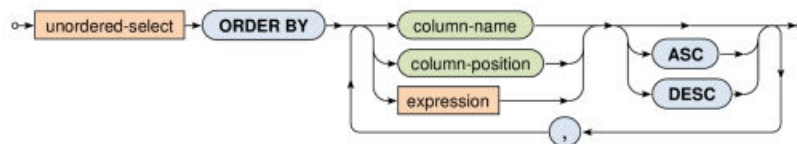
```
SELECT Company FROM Flights
UNION
SELECT Company FROM Aircrafts
```

Company
CSA
Lufthansa
Air Canada
KLM

- oba operandy musí být kompatibilní
  - o tj. mají stejný počet sloupců...
  - o ... a tyto sloupce musí být stejného typu

## Řadící dotazy

- **ORDER BY**
  - o **ASC** (defaultní) – vzestupně
  - o **DESC** - sestupně
- výsledné řádky nemají definované pořadí (není-li explicitně specifikováno)
- pro řazení může být použito vícero řádků
- NULL předchází všechny ostatní hodnoty



```
SELECT DISTINCT Destination
FROM Flights
ORDER BY Destination ASC
```

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130



Destination
Amsterdam
London
Milano
New York
Rotterdam
Stuttgart
Toronto

## Vnořené dotazy

- lze použít:
  - o v predikátech (ANY, SOME, ALL, IN, EXISTS)
  - o pro definici tabulek z FROM klauzule
  - o skoro v jakémkoliv výrazu, jsou-li výsledkem skalární hodnoty

- Find all the scheduled flights which have higher than average number of passengers.

```
SELECT *  
FROM Flights  
WHERE (Passengers > (SELECT AVG(Passengers) FROM Flights))
```

Flight	Company	Destination	Passengers
OK251	CSA	New York	276
LH438	Lufthansa	Stuttgart	68
OK012	CSA	Milano	37
OK321	CSA	London	156
AC906	Air Canada	Toronto	116
KL7621	KLM	Rotterdam	75
KL1245	KLM	Amsterdam	130



Flight	Company	Destination	Passengers
OK251	CSA	New York	276
OK321	CSA	London	156
KL1245	KLM	Amsterdam	130

- Return the number of suitable aircrafts for each flight.
  - Only aircrafts of a given company and sufficient capacity can be used
  - Note how values from the outer query are bound with the inner one

```
SELECT  
Flights.*,  
(  
    SELECT COUNT(*)  
    FROM Aircrafts AS A  
    WHERE  
        (A.Company = F.Company) AND  
        (A.Capacity >= F.Passengers)  
) AS Aircrafts  
FROM Flights AS F
```

Flight	Company	Destination	Passengers	Aircrafts
OK251	CSA	New York	276	0
LH438	Lufthansa	Stuttgart	68	0
OK012	CSA	Milano	37	1
OK321	CSA	London	156	0
AC906	Air Canada	Toronto	116	0
KL7621	KLM	Rotterdam	75	2
KL1245	KLM	Amsterdam	130	2

Aircraft	Company	Capacity
Boeing 717	CSA	106
Airbus A380	KLM	555
Airbus A350	KLM	253