

Example of GROUP BY

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC
```

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC
```

首先执行查询，得到待统计的元组集合

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

WHERE JOB<>'M'

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC

```

- 根据**GROUP**属性上的取值，将元组集合划分成若干个小组（子集）；
- 在同一个小组中，分组属性上的取值都相同；不同小组之间，至少在某一个分组属性上的取值是不同的。
- 组与组之间依据它们的分组属性值进行了排序。

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

GROUP BY DEP, JOB

JOB	SAL	DEP
C	27000	BLU
S	31000	BLU
S	29000	BLU
S	30000	BLU
C	27000	GRE
C	28000	GRE
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

JOB	SAL	DEP
C	27000	BLU
S	31000	BLU
S	29000	BLU
S	30000	BLU
C	27000	GRE
C	28000	GRE
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

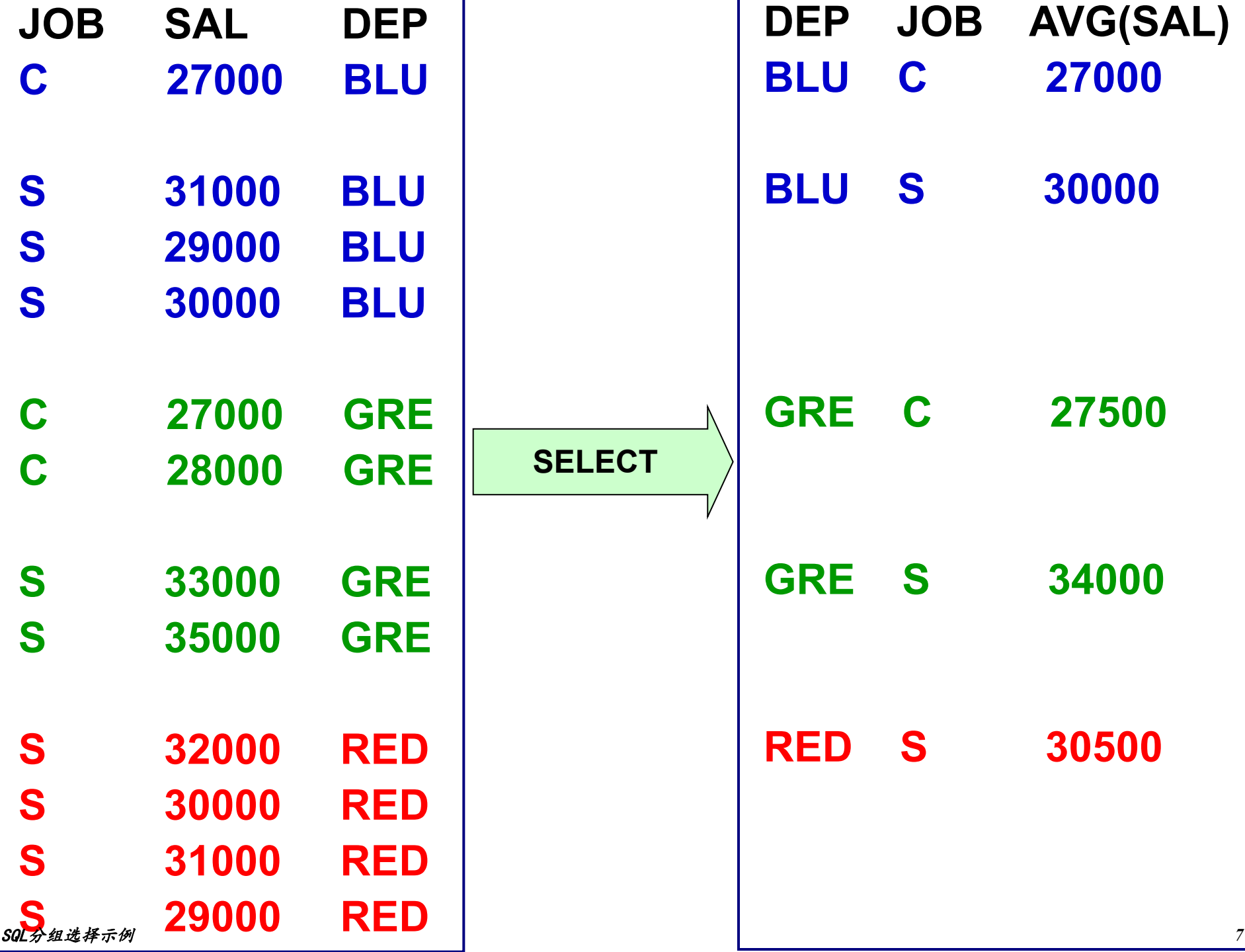
```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC

```

根据**SELECT**子句进行如下统计：

- 每一个小组(**group**)都单独进行统计；
- 一个小组产生一条结果元组；



DEP	JOB	AVG(SAL)
BLU	C	27000
BLU	S	30000
GRE	C	27500
GRE	S	34000
RED	S	30500

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC
```

根据**ORDER BY**子句的要求，对
结果元组进行排序输出。

DEP	JOB	AVG(SAL)
BLU	C	27000
BLU	S	30000
GRE	C	27500
GRE	S	34000
RED	S	30500

ORDER BY
AVG(SAL)
DESC

DEP	JOB	AVG(SAL)
GRE	S	34000
RED	S	30500
BLU	S	30000
GRE	C	27500
BLU	C	27000

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
ORDER BY AVG(SAL) DESC

```



DEP	JOB	AVG(SAL)
GRE	S	34000
RED	S	30500
BLU	S	30000
GRE	C	27500
BLU	C	27000

查询结果

Example of GROUP BY+HAVING

□ 分组选择子句：**HAVING group_condition**

➤ 根据**GROUP BY**子句的分组结果，定义分组选择条件

- 在**HAVING**子句中给出的查询条件是定义在分组后的元组集合上的，通常是根据该集合上的某种统计计算的结果来定义查询条件
- 只有满足条件**group_condition**的元组集合才会被保留下来，用于生成最终的查询结果

Example of GROUP BY+HAVING

- ❑ **GROUP BY & HAVING**此两子句可以对映像语句所得到的集合元组分组（用**GROUP BY**子句），并还可利用**HAVING**子句设置逻辑条件

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL) > 28000
ORDER BY AVG(SAL) DESC
```

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC

```

首先根据**WHERE**子句进行查询，得到待统计的元组集合

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

WHERE JOB<>'M'

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC

```

- 根据**GROUP**属性上的取值，将元组集合划分成若干个小组（子集）；
- 在同一个小组中，分组属性上的取值都相同；不同小组之间，至少在某一个分组属性上的取值是不同的。
- 组与组之间依据它们的分组属性值进行了排序。

JOB	SAL	DEP
S	31000	BLU
S	30000	BLU
C	27000	GRE
S	33000	GRE
S	32000	RED
C	28000	GRE
S	30000	RED
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

GROUP BY DEP, JOB

JOB	SAL	DEP
C	27000	BLU
S	31000	BLU
S	29000	BLU
S	30000	BLU
C	27000	GRE
C	28000	GRE
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

JOB	SAL	DEP
C	27000	BLU
S	31000	BLU
S	29000	BLU
S	30000	BLU
C	27000	GRE
C	28000	GRE
S	33000	GRE
S	35000	GRE
S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC

```

根据HAVING子句进行如下GROUP选择：

- 对每一个小组(group)，都按照HAVING子句的要求进行统计计算；
- 如果一个group的统计结果满足HAVING子句的要求，那么该group将被保留下来，否则整个group将被舍弃；

JOB	SAL	DEP
C	27000	BLU

S	31000	BLU
S	29000	BLU
S	30000	BLU

C	27000	GRE
C	28000	GRE

S	33000	GRE
S	35000	GRE

S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

HAVING
AVG(SAL) >
28000

JOB	SAL	DEP
S	31000	BLU
S	29000	BLU
S	30000	BLU

S	33000	GRE
S	35000	GRE

S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

JOB	SAL	DEP
-----	-----	-----

S	31000	BLU
S	29000	BLU
S	30000	BLU

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC
```

S	33000	GRE
S	35000	GRE

S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

根据**SELECT**子句进行如下统计：

- 每一个小组(**group**)都单独进行统计；
- 一个小组产生一条结果元组；

JOB	SAL	DEP
-----	-----	-----

S	31000	BLU
S	29000	BLU
S	30000	BLU

S	33000	GRE
S	35000	GRE

S	32000	RED
S	30000	RED
S	31000	RED
S	29000	RED

SELECT

DEP	JOB	AVG(SAL)
-----	-----	----------

BLU	S	30000
-----	---	-------

GRE	S	34000
-----	---	-------

RED	S	30500
-----	---	-------

DEP	JOB	AVG(SAL)
-----	-----	----------

BLU	S	30000
-----	---	-------

GRE	S	34000
-----	---	-------

RED	S	30500
-----	---	-------

```
SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC
```

根据**ORDER BY**子句的要求，对结果元组进行排序输出。

DEP	JOB	AVG(SAL)
-----	-----	----------

BLU	S	30000
-----	---	-------

GRE	S	34000
-----	---	-------

RED	S	30500
-----	---	-------

ORDER BY
AVG(SAL)
DESC

DEP	JOB	AVG(SAL)
-----	-----	----------

GRE	S	34000
-----	---	-------

RED	S	30500
-----	---	-------

BLU	S	30000
-----	---	-------

JOB	SAL	DEP
S	31000	BLU
M	32000	RED
S	30000	BLU
C	27000	GRE
S	33000	GRE
M	31000	BLU
S	32000	RED
C	28000	GRE
S	30000	RED
M	33000	GRE
S	31000	RED
S	35000	GRE
C	27000	BLU
S	29000	RED
S	29000	BLU

```

SELECT DEP, JOB, AVG(SAL)
FROM EMPL
WHERE JOB <> 'M'
GROUP BY DEP, JOB
HAVING AVG(SAL)>28000
ORDER BY AVG(SAL) DESC

```



DEP	JOB	AVG(SAL)
GRE	S	34000
RED	S	30500
BLU	S	30000

查询结果

SELECT语句执行的一般规则

1. 合并**FROM**子句中的表（笛卡儿乘积）；
2. 利用**WHERE**子句中的条件进行元组选择，抛弃不满足**WHERE**条件的那些元组；
3. 根据**GROUP BY**子句对保留下来的元组进行分组；
4. 利用**HAVING**子句中的条件对分组后的元组集合（group）进行选择，抛弃不满足**HAVING**条件的那些元组集合；
5. 根据**SELECT**子句进行统计计算，生成结果关系中的元组：**a group → a result row**
6. 根据**ORDER BY**子句对查询结果进行排序。