二 OVER(PARTITION BY)

学习目标

- 掌握 OVER（PARTITION BY)的使用方法

1 PARTITION BY基本用法

- 接下来，我们来介绍如何在 `OVER()` 中添加 `PARTITION BY` ，基本的语法如下：

```
<window_function> OVER (PARTITION BY column1, column2 ... column_n)
```

- `PARTITION BY` 的作用与 `GROUP BY` 类似：将数据按照传入的列进行分组，与 `GROUP BY` 的区别是， `PARTITION BY` 不会改变结果的行数

2 PARTITION BY与GROUP BY区别

① group by是分组函数，partition by是分析函数

② 在执行顺序上：

```
from > where > group by > having > order by
```

而partition by应用在以上关键字之后，可以简单理解为就是在执行完select之后，在所得结果集之上进行partition by分组

③ partition by相比较于group by，能够在保留全部数据的基础上，只对其中某些字段做分组排序（类似excel中的操作），而group by则只保留参与分组的字段和聚合函数的结果（类似excel中的pivot透视表）

我们看下面的例子：

不同型号火车基本信息表（train）

id	model	max_speed	production_year	first_class_places	second_class_places
1	CR200J	160	2000	30	230
2	CR200J	160	2000	40	210
3	CRH1A	200	2005	40	180
4	CRH2A	250	2010	45	150
5	CRH2B	250	2010	50	250
6	CRH3A	250	2013	60	250

> first_class_places一等座，second_class_places二等座

运营线路表（ROUTE）

id	name	from_city	to_city	distance
1	Manchester Express	Sheffield	Manchester	60
2	GoToLeads	Manchester	Leeds	70
3	StudentRoute	London	Oxford	90
4	MiddleEnglandWay	London	Leicester	160
5	BeatlesRoute	Liverpool	York	160
6	NewcastleDaily	York	Newcastle	135
7	ScotlandSpeed	Newcastle	Edinburgh	200

票价表（TICKET）

id	price	class	journey_id
1	200	2	24
2	76	1	12
3	102	2	6
4	126	2	11
5	80	1	17
6	74	1	5
……	……	……	……

时刻表（JOURNEY）

id	train_id	route_id	date
1	1	1	2016-01-03
2	1	2	2016-01-04
3	1	3	2016-01-05
4	1	4	2016-01-06
5	2	2	2016-01-03
6	2	3	2016-01-04
7	2	4	2016-01-05
8	2	5	2016-01-06
……	……	……	……

- 需求：查询每种类型火车的ID，型号，一等座数量，同型号一等座数量总量

id	model	first_class_places	sum
1	CR200J	30	70
2	CR200J	40	70
3	CRH1A	40	40
4	CRH2A	45	45
5	CRH2B	50	110
6	CRH2B	60	110

- 使用`PARTITION BY`，可以轻松计算整个组的统计信息，并保留各个行的详细信息
- 下面的SQL就可以实现我们的需求：

```
SELECT
  id,
  model,
  first_class_places,
  SUM(first_class_places) OVER (PARTITION BY model)
FROM train;
```

- 如果用`GROUP BY`实现上面相同的需求，我们需要通过子查询将结果与原始表`JOIN`，对比起来，使用窗口函数的实现更加简洁

```
-- GROUP BY实现
SELECT
    id,
    train.model,
    first_class_places,
    c.sum
FROM
    train
    JOIN ( SELECT model, SUM( first_class_places ) AS sum FROM
train GROUP BY model ) AS c
    ON train.model = c.model
```

练习11

- 需求：查询每天的车次数量
- 查询结果包括：时刻表车次ID，日期，每天的车次数量

```
SELECT
  id,
  date,
  COUNT(id) OVER(PARTITION BY date) as count
FROM journey;
```

查询结果

id	date	count
1	2016/1/3	7
5	2016/1/3	7
9	2016/1/3	7
17	2016/1/3	7
21	2016/1/3	7
25	2016/1/3	7
29	2016/1/3	7
2	2016/1/4	8
6	2016/1/4	8
10	2016/1/4	8
13	2016/1/4	8
14	2016/1/4	8
22	2016/1/4	8
26	2016/1/4	8
30	2016/1/4	8
3	2016/1/5	8
7	2016/1/5	8
11	2016/1/5	8
15	2016/1/5	8
18	2016/1/5	8
19	2016/1/5	8
23	2016/1/5	8
27	2016/1/5	8
4	2016/1/6	6

id	date	count
8	2016/1/6	6
12	2016/1/6	6
16	2016/1/6	6
24	2016/1/6	6
28	2016/1/6	6
20	2016/3/6	1

- 在上一小节我们介绍过，OVER() 实在Where 之后执行，看下面的SQL：

```
SELECT
  id,
  model,
  max_speed,
  COUNT(id) OVER (PARTITION BY max_speed)
FROM train
WHERE production_year != 2013;
```

- 这里会把生产日期是2013年的数据过滤掉

练习12

- 需求：按车型分组，每组中满足一等座>30,二等座>180的有几条记录
  - 查询结果包括如下字段：`id`，`model`，`first_class_places`，`second_class_places`，`count` 满足条件的记录数量

```
SELECT
  id,
  model,
  first_class_places,
  second_class_places,
  COUNT(id) OVER (PARTITION BY model)
FROM train
WHERE first_class_places > 30
  AND second_class_places > 180;
```

查询结果

id	model	first_class_places	second_class_places	count
2	CR200J	40	210	1
5	CRH2B	50	250	2
6	CRH2B	60	250	2

3 PARTITION BY传入多列

- 需求：查询每天，每条线路速的最快车速
- 查询结果包括如下字段：线路ID,日期，车型，每天相同线路的最快车速

```
SELECT
  journey.id,
  journey.date,
  train.model,
  train.max_speed,
  MAX(max_speed) OVER(PARTITION BY route_id, date)
FROM journey
JOIN train
  ON journey.train_id = train.id;
```

查询结果

id	date	model	max_speed	max
1	2016/1/3	CR200J	160	160
26	2016/1/4	CRH2B	250	250
18	2016/1/5	CRH2B	250	250
23	2016/1/5	CRH2B	250	250
20	2016/3/6	CRH2B	250	250
5	2016/1/3	CR200J	160	250
17	2016/1/3	CRH2B	250	250
2	2016/1/4	CR200J	160	160
9	2016/1/3	CRH1A	200	250
21	2016/1/3	CRH2B	250	250
6	2016/1/4	CR200J	160	250
22	2016/1/4	CRH2B	250	250
3	2016/1/5	CR200J	160	250
19	2016/1/5	CRH2B	250	250
27	2016/1/5	CRH1A	200	250
29	2016/1/3	CRH1A	200	200
13	2016/1/4	CRH2A	250	250
7	2016/1/5	CR200J	160	160
4	2016/1/6	CR200J	160	250
24	2016/1/6	CRH2B	250	250
25	2016/1/3	CR200J	160	160
10	2016/1/4	CRH1A	200	250
14	2016/1/4	CRH2A	250	250
11	2016/1/5	CRH1A	200	200

id	date	model	max_speed	max
8	2016/1/6	CR200J	160	160
30	2016/1/4	CRH2B	250	250
15	2016/1/5	CRH2A	250	250
12	2016/1/6	CRH1A	200	250
28	2016/1/6	CRH2A	250	250
16	2016/1/6	CRH2A	250	250

练习13

- 需求：查询时刻表中的车次ID，运营车辆的生产日期（`production_year`），同一种车型的车次数量，同一线路的车次数量

```
SELECT
  journey.id,
  production_year,
  COUNT(journey.id) OVER(PARTITION BY train_id) as
count_train,
  COUNT(journey.id) OVER(PARTITION BY route_id) as
count_journey
FROM train
JOIN journey
  ON train.id = journey.train_id;
```

查询结果

id	production_year,	count_train	count_journey
1	1905/6/22	5	5
18	1905/7/2	5	5
20	1905/7/2	5	5
23	1905/7/5	5	5
26	1905/7/5	5	5
2	1905/6/22	5	3
5	1905/6/22	4	3
17	1905/7/2	5	3
3	1905/6/22	5	7
6	1905/6/22	4	7
9	1905/6/27	6	7
27	1905/6/27	6	7
19	1905/7/2	5	7
21	1905/7/5	5	7
22	1905/7/5	5	7
4	1905/6/22	5	5
7	1905/6/22	4	5
29	1905/6/27	6	5
13	1905/7/2	5	5
24	1905/7/5	5	5
25	1905/6/22	5	5
8	1905/6/22	4	5
10	1905/6/27	6	5
11	1905/6/27	6	5

id	production_year,	count_train	count_journey
14	1905/7/2	5	5
12	1905/6/27	6	4
15	1905/7/2	5	4
28	1905/7/2	5	4
30	1905/7/2	5	4
16	1905/7/2	5	1

练习14

- 需求：查询票价表，返回`id`，`price`，`date` ，并统计每天的在售车票的平均价格，每天的在售车票种类。（不统计运营车辆id为5的数据）

```
SELECT
  ticket.id,
  date,
  price,
  AVG(price) OVER(PARTITION BY date),
  COUNT(ticket.id) OVER(PARTITION BY date)
FROM ticket
JOIN journey
  ON ticket.journey_id = journey.id
WHERE train_id != 5;
```

查询结果

id	date	price	avg	count
6	2016/1/3	74	119.3333	21
7	2016/1/3	200	119.3333	21
102	2016/1/3	168	119.3333	21
136	2016/1/3	126	119.3333	21
149	2016/1/3	100	119.3333	21
101	2016/1/3	80	119.3333	21
34	2016/1/3	200	119.3333	21
23	2016/1/3	62	119.3333	21
87	2016/1/3	200	119.3333	21
45	2016/1/3	130	119.3333	21
142	2016/1/3	102	119.3333	21
120	2016/1/3	108	119.3333	21
32	2016/1/3	83	119.3333	21
29	2016/1/3	67	119.3333	21
112	2016/1/3	53	119.3333	21
135	2016/1/3	150	119.3333	21
50	2016/1/3	150	119.3333	21
110	2016/1/3	124	119.3333	21
145	2016/1/3	92	119.3333	21
71	2016/1/3	69	119.3333	21
……	……	……	……	……

小结

- OVER（PARTITION BY x）的工作方式与GROUP BY类似，将x列中，所有值相同的行分到一组中
- PARTITON BY 后面可以传入一列数据，也可以是多列（需要用逗号隔开列名）

练习15

- 使用上一小节的员工，部门表
- 需求：统计每一个员工的姓名，所在部门，薪水，该部门的最低薪水，该部门的最高薪水

```
SELECT
  first_name,
  last_name,
  department.name AS `department`,
  salary,
  MIN(salary) OVER(PARTITION BY department.name) `min_salary`,
  MAX(salary) OVER(PARTITION BY department.name) `max_salary`
FROM employee
Join department
on department.id = employee.department_id
```

查询结果

first_name	last_name	department	salary	min_salary	max_salary
Karen	Morris	Accounting	6347	6286	6347
Kathy	Sanders	Accounting	6286	6286	6347
Joe	Thompson	Help Desk	5639	2076	5639
Barbara	Clark	Help Desk	3232	2076	5639
Todd	Bell	Help Desk	4653	2076	5639
Ronald	Butler	Help Desk	2076	2076	5639
Larry	Lee	Human Resources	2796	2796	6419
Willie	Patterson	Human Resources	4771	2796	6419
Janet	Ramirez	Human Resources	3782	2796	6419
Doris	Bryant	Human Resources	6419	2796	6419
Amy	Williams	Human Resources	6261	2796	6419
Keith	Scott	Human Resources	4928	2796	6419
Diane	Turner	IT	5330	3617	5330
Clarence	Robinson	IT	3617	3617	5330
Eugene	Phillips	IT	4877	3617	5330
Philip	Mitchell	IT	5259	3617	5330
Ann	Wright	Management	2094	2094	6923
Charles	Wilson	Management	5167	2094	6923
Russell	Johnson	Management	3762	2094	6923
Jacqueline	Cook	Management	6923	2094	6923

练习16

- 需求：统计每名员工在所在部门的工资占比
- 返回如下字段
  - 姓名，所在部门名称，薪水，薪水占该部门比例

```
SELECT
  first_name,
  last_name,
  department.name as `department`,
  salary,
  salary / sum(salary) OVER(partition by department.name) *100
as percent
FROM employee
join department
on employee.department_id = department.id
```

查询结果

first_name	last_name	department	salary	percent
Karen	Morris	Accounting	6347	50.2414
Kathy	Sanders	Accounting	6286	49.7586
Joe	Thompson	Help Desk	5639	36.1474
Barbara	Clark	Help Desk	3232	20.7179
Todd	Bell	Help Desk	4653	29.8269
Ronald	Butler	Help Desk	2076	13.3077
Larry	Lee	Human Resources	2796	9.6557
Willie	Patterson	Human Resources	4771	16.4762
Janet	Ramirez	Human Resources	3782	13.0607
Doris	Bryant	Human Resources	6419	22.1674
Amy	Williams	Human Resources	6261	21.6217
Keith	Scott	Human Resources	4928	17.0183
Diane	Turner	IT	5330	27.9306
Clarence	Robinson	IT	3617	18.954
Eugene	Phillips	IT	4877	25.5568
Philip	Mitchell	IT	5259	27.5586
Ann	Wright	Management	2094	11.6683
Charles	Wilson	Management	5167	28.7919
Russell	Johnson	Management	3762	20.9629
Jacqueline	Cook	Management	6923	38.5768