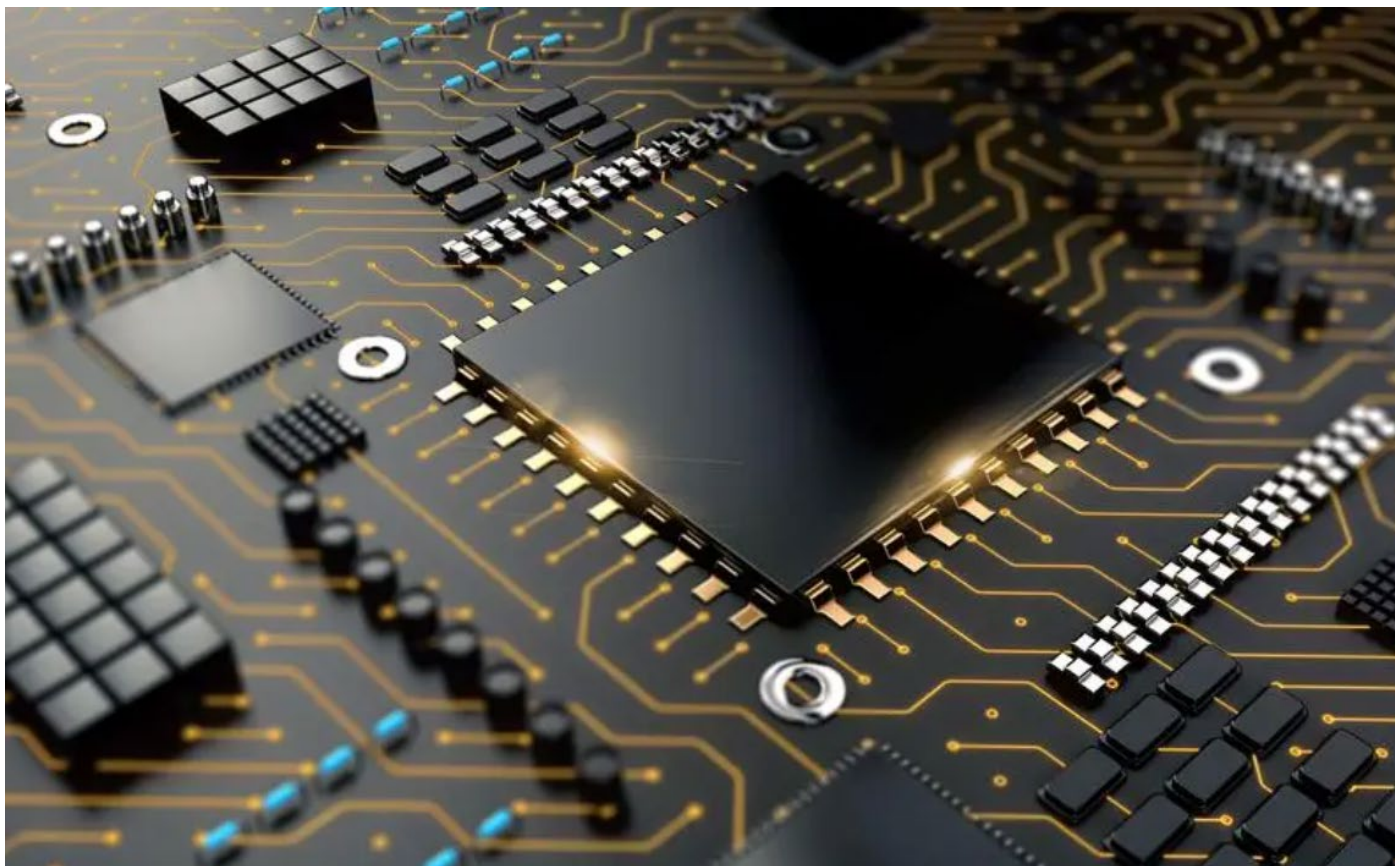




中山大學
SUN YAT-SEN UNIVERSITY



2021级 《数据库原理与应用》 第7周

2024.4.17

Michael Stonebraker



中山大學
SUN YAT-SEN UNIVERSITY

- Michael Stonebraker, 图灵奖获得者。著名的数据库科学家, 他在1992 年提出对象关系数据库模型在加州伯克利分校计算机教授达25年。在此期间他创作了Ingres, Illustra, Cohera, StreamBase Systems 和Vertica等系统。Stonebraker教授也曾担任过Informix的CEO, 目前他是MIT麻省理工学院客席教授。



2024.4.17

从Ingres开始

- 1973 年 IBM 启动了 System R 项目，项目组发表了一系列关于关系数据库的文章。两个伯克莱大学的科学家，[Michael Stonebraker](#) 和 [Eugene Wong](#) 读了这些文章后非常感兴趣，于是决定自己启动一个关系数据的研究项目。他们已经为一个地理数据库系统申请到了资金，他们称为 ingres，意思是交互式图形获取系统
- Ingres的代码是可以免费获得的，到 1980 年止，共分发了 1000 份拷贝，不少公司使用这些代码形成了自己的产品线。Informix 是最早的用户之一，并且其员工完全是 Ingres 项目的人员。他们在 1984 发布了基于 Ingres 的第一版产品，到 1997 年已经成了第二大数据库供应商。然而，一系列管理和财会方面的失误在短短两年内毁了公司的信誉，并于 2000 年被 IBM 收购。

Sybase, MS SQLserver的渊源

- Stonebraker 自己离开了伯克立大学在1982年创建了Ingres 公司，Stonebraker 在这个公司工作到 1991 年，然后公司卖给了 ASK，1994年 ASK/ingres 被 CA Computer Associates 收购。2004 年 CA 在开源许可下发布了 Ingres release 3，并继续开发销售 Ingres。
- Robert Epstein 在伯克立大学项目中的重要程序员，创建了 Britton-Lee，后来的 **Sybase**，Sybase 在 80 到 90 年期间是第二号数据库产品，Sybase 于1992年将产品卖给微软，微软称之为 MS SQLServer。
- 87年左右，Sybase联合了微软，共同开发SQL Server。原始代码的来源与Ingres有些渊源。后来1994年，两家公司合作终止。此时，两家公司都拥有一套完全相同的SQLServer代码。可以认为，Stonebraker教授是目前主流数据库的奠基人。

- Stonebraker 教授领导了称为Postgres的后Ingres项目。这个项目的成果非常巨大，在现代数据库的许多方面都做出大量的贡献。Stonebraker 教授还做出了一件造福全人类的事情，那就是把Postgres 放在了BSD 版权的保护下。如今Postgres名字已经变成了PostgreSQL，功能也是日渐强大。
- 在返回 Berkeley 之后，Stonebraker 开始了一项后-Ingres 计划来致力于解决关系模型的数据库管理现有实现的局限性。其中主要的是它们不能让用户定义组合更简单域的新域（或者叫“类型”）（参见关系模型获得对术语“域”的解释。）结果的计划叫做 Postgres，以介入对增加完整的类型支持所需要的最小数目的特征为目标。其中包括定义类型的功能，还有完全描述联系的能力 - 联系至今已经广泛使用但仍由用户完全维护。在 Postgres 中数据库“理解”联系，并能以使用规则的自然方式在有关联的表中检索信息。详情请参见 PostgreSQL 的文章。在 1990 年 Stonebraker 再次离开 Berkeley 去商业化 Postgres，使用了名字 Illustra。Illustra 后来被 Informix 并购，而 Stonebraker 再次回去搞高等教育了。

2024.4.17

PostgreSQL现状

- 最像Oracle的开源数据库，常作为O的替代品
- 生态丰富，以插件形式增强大量功能（例如GIS或索引）
- SQL解析器在开源数据库中较为出色，常被其它开源数据库“借用”
- 由于许可证宽松，所以有很多厂商基于它开发魔改版。比较有名的：Greenplum，华为高斯
- 信创活动与PostgreSQL

- 可以作为实验环境替代品（鼓励大家使用其完成作业）
- 比较与Oracle SQL的差别
- （可能用于）比较与Oracle在体系结构上的区别

Michael Widenius (麦克尔维尔纽斯)



中山大學
SUN YAT-SEN UNIVERSITY

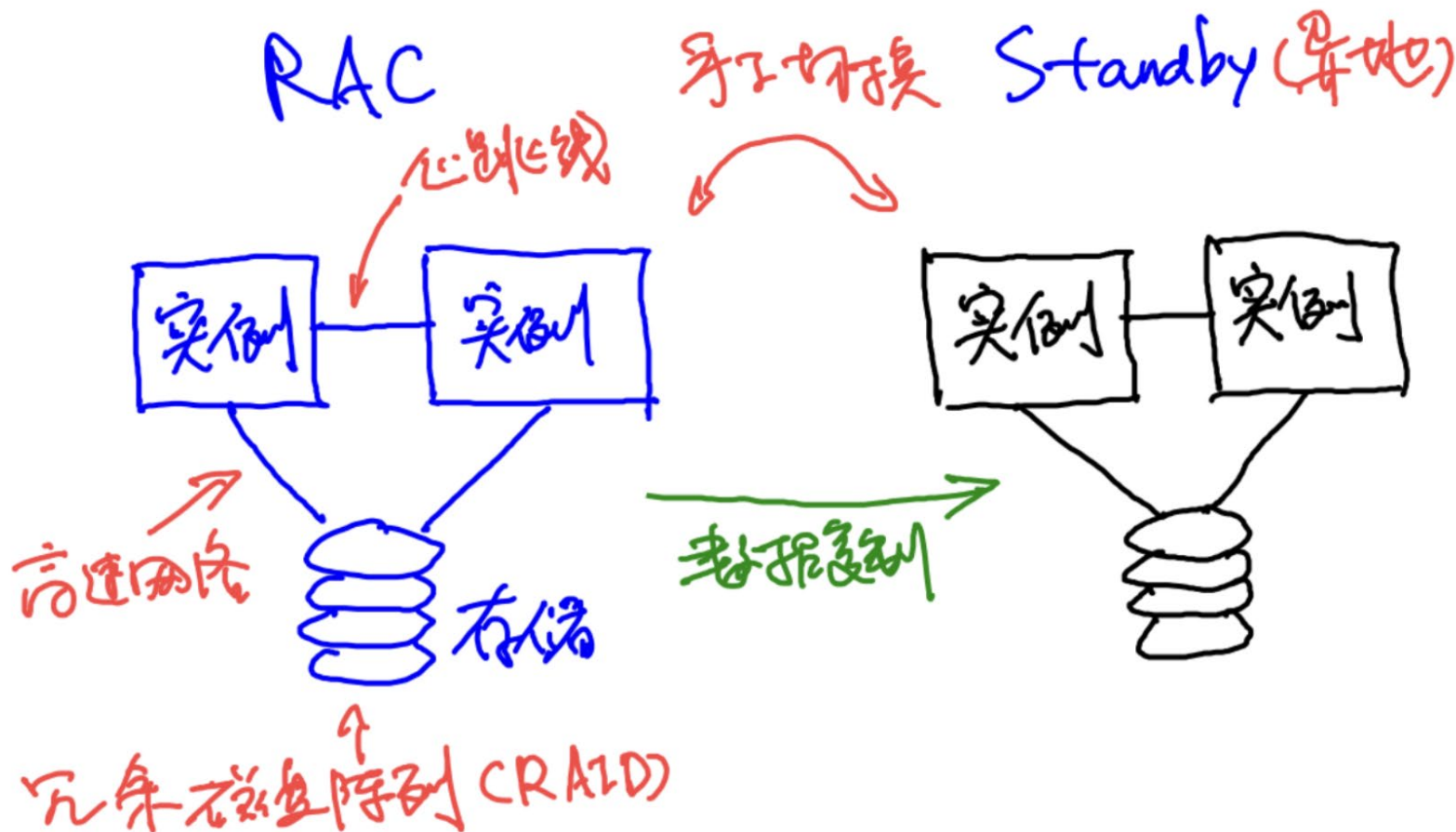


2024.4.17

- 昵称Monty，芬兰人，赫尔辛基理工大学毕业（**中途辍学**）。MySQL数据库和MySQL AB公司（瑞典企业，从事MySQL企业服务）创始人，另一位创始人是David Axmark（大卫阿克马克）
- MySQL官方发音：My-S-Q-L。但My是Monty小女儿名字，按瑞典语发音“米”，因此可能更精准的读法是“米-S-Q-L”
- MySQL后来被Oracle收购，但创始人不适应美国公司的企业文化，很快退出
- 基于MySQL开源代码发布了派生分支“MariaDB”，Maria是Monty另一位女儿的名字，他还有一个叫Max的儿子

- 遵守GNU/GPL V2许可证
- 在互联网公司被大量使用（例如百度），但因为对事务支持和SQL解析器上的弱点，企业使用较少。有部分企业用于对事务要求不高的场景（例如详单查询）
- 适合构成“单点写-复制-多点读”模式的集群（工行9000节点容器化部署MySQL）
- MyISAM和InnoDB两种存储引擎，后者支持事务
- 与Oracle差别较大，适合用于比较

Oracle的“分布式系统” RAC+Standby



- 高可用
- 负载均衡
- 灾备
- 是否有完善的高可用、负载均衡、灾备能力是企业级数据库软件与开源软件的分水岭
- 信创运动与国产数据库
- 大数据已死?

- 求底层员工（即没有直属下属）中工资最高的员工名字

```
SQL> select ename from emp
      2  where sal=(select max(sal) from emp where empno in
      3    ((select empno from emp) minus (select mgr from emp)))
      4  and empno in ((select empno from emp) minus (select mgr from emp)) ;
```

ENAME

ALLEN

夹带知识点：集合运算

■ Minus, intersect, union, union all

select ename from emp

where sal=(select max(sal) from emp where empno in
((select empno from emp) **minus** (select mgr from
emp)));

Union与union all



■ Union (去除重复元素) , union all (不去除重复元素)

```
SQL> (select empno from emp) union all (select mgr from emp);
```

EMPNO
7369
7499
7521
7566
7654
7698
7782
7839
7844
7900
7902
7934
7902
7698
7698
7839
7698
7839
7839
7698
7698
7566
7782

已选择24行。

- 用1条SQL语句建立以下统计表格，分别统计每个部门，每个年份进入公司，每个工种的人数

```
SQL> (select deptno,count(*) from emp group by deptno) union all  
2  (select to_char(hiredate,'yyyy'),count(*) from emp group by to_char(hiredate,'yyyy')) union all  
3  (select job,count(*) from emp group by job);  
(select deptno,count(*) from emp group by deptno) union all  
*
```

第 1 行出现错误:

ORA-01790: 表达式必须具有与对应表达式相同的数据类型

作业答案

```
SQL> (select dname,count(*) from emp, dept where emp.deptno=dept.deptno group by dname) union all  
2 (select to_char(hiredate,'yyyy'),count(*) from emp group by to_char(hiredate,'yyyy')) union all  
3 (select job,count(*) from emp group by job);
```

DNAME	COUNT(*)
ACCOUNTING	3
RESEARCH	3
SALES	6
1980	1
1982	1
1981	10
CLERK	3
SALESMAN	4
PRESIDENT	1
MANAGER	3
ANALYST	1

已选择11行。

作业答案



```
SQL> select dname,to_char(hiredate,'yyyy'),job,count(*),
2 grouping(dname),grouping(to_char(hiredate,'yyyy')),grouping(job)
3 from emp,dept
4 where emp.deptno=dept.deptno
5 group by rollup (dname,to_char(hiredate,'yyyy'),job);
```

DNAME	TO_C	JOB	COUNT(*)	GROUPING(DNAME)	GROUPING(TO_CHAR(HIREDATE,'YYYY'))	GROUPING(JOB)
SALES	1981	CLERK	1	0	0	0
SALES	1981	MANAGER	1	0	0	0
SALES	1981	SALESMAN	4	0	0	0
SALES	1981		6	0	0	1
SALES			6	0	1	1
RESEARCH	1980	CLERK	1	0	0	0
RESEARCH	1980		1	0	0	1
RESEARCH	1981	ANALYST	1	0	0	0
RESEARCH	1981	MANAGER	1	0	0	0
RESEARCH	1981		2	0	0	1
RESEARCH			3	0	1	1
ACCOUNTING	1981	MANAGER	1	0	0	0
ACCOUNTING	1981	PRESIDENT	1	0	0	0
ACCOUNTING	1981		2	0	0	1
ACCOUNTING	1982	CLERK	1	0	0	0
ACCOUNTING	1982		1	0	0	1
ACCOUNTING			3	0	1	1
			12	1	1	1

作业答案



```
SQL> select  dname,to_char(hiredate,'yyyy'),job,count(*) from emp,dept
2  where emp.deptno=dept.deptno
3  group by cube (dname,to_char(hiredate,'yyyy'),job)
4  having grouping(dname)+grouping(to_char(hiredate,'yyyy'))+grouping(job)=2;
```

DNAME	TO_C	JOB	COUNT(*)
		CLERK	3
		ANALYST	1
		MANAGER	3
		SALESMAN	4
		PRESIDENT	1
	1980		1
	1981		10
	1982		1
SALES			6
RESEARCH			3
ACCOUNTING			3

已选择11行。

- 在学生选修表SC与课程表C放置一些数据，写一条SQL求出选修了C表所列全部课程的学生名单
- 思路：翻译为not exists算法能解决的形式，找出这样的学生，不存在一门课他是没有选修的

建立数据



```
insert into s values ('s1','X','MA',21);
```

```
insert into s values ('s2','Y','PH',20);
```

```
insert into s values ('s3','Z','CS',21);
```

```
insert into s values ('s4','W','MA',25);
```

```
insert into c values ('c1','Algebra',NULL);
```

```
insert into c values ('c2','C Language',NULL);
```

```
insert into c values ('c3','Analysis',NULL);
```

```
insert into sc values ('s1','c1',90);
```

```
insert into sc values ('s1','c2',99);
```

```
insert into sc values ('s2','c1',70);
```

```
insert into sc values ('s2','c2',90);
```

```
insert into sc values ('s2','c3',92);
```

```
insert into sc values ('s3','c3',60);
```

```
SQL> select * from s;
```

S#	SN	SD	SA
s1	X	MA	21
s2	Y	PH	20
s3	Z	CS	21
s4	W	MA	25

```
SQL> select * from c;
```

C#	CN	PC#
c1	Algebra	
c2	C Language	
c3	Analysis	

```
SQL> select * from sc;
```

S#	C#	G
s1	c1	90
s1	c2	99
s2	c1	70
s2	c2	90
s2	c3	92
s3	c3	60

已选择6行。


```
select sn from s
where not exists ( select * from c where
                    not exists (select * from sc
                                where s#=s.s# and c#=c.c#)
                    );
```

作业答案



中山大學
SUN YAT-SEN UNIVERSITY

```
SQL> select sn from s
      2  where not exists ( select * from c where
      3                        not exists (select * from sc
      4                                    where s#=s.s# and c#=c.c#)
      5                        );
```

SN

Y

SQL>

SQL>

- 王珊《数据库系统概论》第5版第110页例3.62
- 这种运算在关系代数中称为“关系 (table) 的除法”,记为“ $sc \div c$ ”,这是唯一没有被翻译为SQL基本操作的关系代数运算

- N年前一道研究生入学考试题：列出选修了每一门课的同学人数
- 难道不是 “`select c#,count(*) from sc group by c#`” ？

- 怎样用一条SQL语句判断两个集合（假设都没有重复元素）是否相等或是子集关系？
- 列出选修课程与某位指定同学完全一样的同学
- 列出所有选修课程完全一样的同学名单，以学号对的形式输出结果

- 有A, B两张表, 均有C1, C2两列, C1代表商品名称, C2代表商品价格。B中的商品有些是A中的原有商品, 有些是在A中没有的新商品, 要求对A表进行更新修改, B中原有商品用B里的新价格覆盖A的原价格, B中的新商品则插入到A中, 要求1条SQL语句完成

```
SQL> select * from ta;
```

C	C2
X	100
Y	140
Z	30
W	500
U	160
V	230
P	800
Q	320

已选择8行。

```
SQL> select * from tb;
```

C	C2
V	1300
Y	1400
Q	1500

```
SQL> insert into tb values ('A',2400);
```

已创建 1 行。

```
SQL> insert into tb values ('B',3400);
```

已创建 1 行。

```
SQL> commit;
```

2

```
SQL>
```

```
SQL> select * from tb;
```

C	C2
A	2400
B	3400
V	1300
Y	1400
Q	1500

用两条语句完成的方法

- 用已经学过的语句不大可能一条语句完成, update不能insert, insert不能update
- 先insert A表没有的行
- 化归为上周作业的问题
- 问题: 为什么insert进去的行会放在最前面?

```
SQL> insert into ta
2  select c1,c2 from tb
3  where not exists(select * from ta where c1=tb.c1);
```

已创建2行。

```
SQL> select * from ta;
```

C	C2
A	2400
B	3400
X	100
Y	140
Z	30
W	500
U	160
V	230
P	800
Q	320

已选择10行。



夹带知识点: MERGE语句

```
SQL> select * from tb;
```

C	C2
A	2400
B	3400
V	1300
Y	1400
Q	1500

```
SQL> merge into ta using tb on (ta.c1=tb.c1)
  2  when matched then update set ta.c2=tb.c2
  3  when not matched then insert values(tb.c1, tb.c2);
```

5 行已合并。

```
SQL> select * from ta;
```

C	C2
A	2400
B	3400
X	100
Y	1400
Z	30
W	500
U	160
V	1300
P	800
Q	1500

- 在SC表中加入大量数据，然后用pivot函数将它转为宽表SCwide。再用unpivot函数将SCwide转为窄表

```
SQL> create table scwide as  
2 select * from sc pivot (max (g) for c# in ('c1' c1,'c2' c2,'c3' c3));
```

表已创建。

```
SQL> select * from scwide;
```

S#	C1	C2	C3
s2	70	90	92
s1	90	99	
s3			60

```
SQL> select * from scwide  
      2 unpivot (g for c# in (c1, c2, c3));
```

S#	C#	G
s2	C1	70
s2	C2	90
s2	C3	92
s1	C1	90
s1	C2	99
s3	C3	60

已选择6行。

- Oracle数据库特有的功能强大的函数，可以非常简洁地解决复杂的数据统计问题
- Oracle三大难度之巅：with递归（N皇后问题），层次查询（员工信息传递最短路径问题），分析函数
- 可以理解为聚组统计的推广，但功能更加丰富
- 市面上关于分析函数的资料不多（可能太复杂不好理解）

分析函数的一个现实应用例子

- 移动运营商数据：手机号，时间戳，连接基站id
- 求每天每个手机用户连接时间最长的3个基站（工作地点？睡觉地点？其它？）
- 其它运营商应用例子：找出信号不好的基站，找出国外回拨电话
- 号段问题

学习使用分析函数时的常见感受



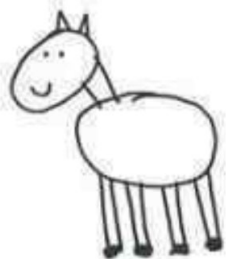
中山大學
SUN YAT-SEN UNIVERSITY



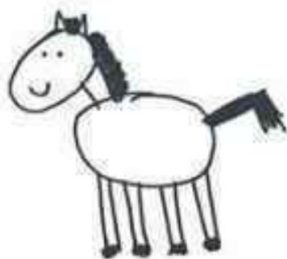
① 画两个圆圈



② 画上脚



③ 画上脸



④ 画上毛发



⑤ 再添加其他细节
就大功告成了!

```
DROP TABLE EMP;
```

```
CREATE TABLE EMP
```

```
(EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
```

```
ENAME VARCHAR2(10),
```

```
JOB VARCHAR2(9),
```

```
MGR NUMBER(4),
```

```
HIREDATE DATE,
```

```
SAL NUMBER(7,2),
```

```
COMM NUMBER(7,2),
```

```
DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT);
```

样板数据



中山大學
SUN YAT-SEN UNIVERSITY

INSERT INTO EMP VALUES

(7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

INSERT INTO EMP VALUES

(7499,'ALLEN','SALESMAN',7698,to_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

INSERT INTO EMP VALUES

(7521,'WARD','SALESMAN',7698,to_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

INSERT INTO EMP VALUES

(7566,'JONES','MANAGER',7839,to_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

INSERT INTO EMP VALUES

(7654,'MARTIN','SALESMAN',7698,to_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

INSERT INTO EMP VALUES

(7698,'BLAKE','MANAGER',7839,to_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

INSERT INTO EMP VALUES

(7782,'CLARK','MANAGER',7839,to_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

INSERT INTO EMP VALUES

(7788,'SCOTT','ANALYST',7566,to_date('13-JUL-87')-85,3000,NULL,20);

样板数据



中山大學
SUN YAT-SEN UNIVERSITY

```
INSERT INTO EMP VALUES
```

```
(7839,'KING','PRESIDENT',NULL,to_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);
```

```
INSERT INTO EMP VALUES
```

```
(7844,'TURNER','SALESMAN',7698,to_date('8-9-1981','dd-mm-yyyy'),1500,0,30);
```

```
INSERT INTO EMP VALUES
```

```
(7876,'ADAMS','CLERK',7788,to_date('13-JUL-87')-51,1100,NULL,20);
```

```
INSERT INTO EMP VALUES
```

```
(7900,'JAMES','CLERK',7698,to_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);
```

```
INSERT INTO EMP VALUES
```

```
(7902,'FORD','ANALYST',7566,to_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);
```

```
INSERT INTO EMP VALUES
```

```
(7934,'MILLER','CLERK',7782,to_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);
```

样板数据



中山大學
SUN YAT-SEN UNIVERSITY

INSERT INTO EMP1 VALUES

(4369,'SAM','CLERK',7902,to_date('27-12-1985','dd-mm-yyyy'),1800,NULL,20);

INSERT INTO EMP VALUES

(4499,'ALEXANDER','SALESMAN',7698,to_date('20-12-1983','dd-mm-yyyy'),1500,500,30);

INSERT INTO EMP VALUES

(4421,'WATSON','SALESMAN',7698,to_date('25-8-1984','dd-mm-yyyy'),1350,400,30);

INSERT INTO EMP VALUES

(4566,'JOHN','ANALYST',7788,to_date('23-4-1983','dd-mm-yyyy'),3500,NULL,20);

INSERT INTO EMP VALUES

(4854,'LARRY','SALESMAN',7698,to_date('25-9-1982','dd-mm-yyyy'),1350,1600,30);

INSERT INTO EMP VALUES

(4695,'BOB','SALESMAN',7698,to_date('1-9-1983','dd-mm-yyyy'),3050,1300,30);

INSERT INTO EMP VALUES

(4682,'MAY','CLERK',7782,to_date('19-10-1981','dd-mm-yyyy'),2050,NULL,10);

INSERT INTO EMP VALUES

(4288,'WENDY','ANALYST',7566,to_date('18-03-87','dd-mm-yyyy '),3000,NULL,20);

INSERT INTO EMP VALUES

(4845,'PETER','SALESMAN',7698,to_date('10-9-1985','dd-mm-yyyy'),1600,0,30);

INSERT INTO EMP VALUES

(4877,'CASSANDRA','CLERK',7788,to_date('13-12-1985','dd-mm-yyyy'),1000,NULL,20);

INSERT INTO EMP VALUES

(4903,'JADE','CLERK',7698,to_date('03-11-1981','dd-mm-yyyy'),1950,NULL,30);

INSERT INTO EMP VALUES

(4932,'IRENE','ANALYST',7566,to_date('04-10-1981','dd-mm-yyyy'),2900,NULL,20);

INSERT INTO EMP VALUES

(4921,'ROSE','CLERK',7782,to_date('03-11-1982','dd-mm-yyyy'),1350,NULL,10);

INSERT INTO EMP VALUES

(5559,'NED','MANAGER',7839,to_date('11-12-1981','dd-mm-yyyy'),2800,NULL,40);

INSERT INTO EMP VALUES

(5599,'ATARI','CLERK',5559,to_date('21-6-1981','dd-mm-yyyy'),1650,NULL,40);

INSERT INTO EMP VALUES

(5521,'ZEN','CLERK',5559,to_date('22-12-1983','dd-mm-yyyy'),1250,NULL,40);

INSERT INTO EMP VALUES

(5566,'SNOW','CLERK',5559,to_date('22-4-1982','dd-mm-yyyy'),2975,NULL,40);

样板数据



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-12月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2月-81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月-81	1250	500	30
7566	JONES	MANAGER	7839	02-4月-81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5月-81	2850		30
7782	CLARK	MANAGER	7839	09-6月-81	2450		10
7839	KING	PRESIDENT		17-11月-81	5000		10
7844	TURNER	SALESMAN	7698	08-9月-81	1500	0	30
7900	JAMES	CLERK	7698	03-12月-81	950		30
7902	FORD	ANALYST	7566	03-12月-81	3000		20
7934	MILLER	CLERK	7782	23-1月-82	1300		10
4499	ALEXANDER	SALESMAN	7698	20-12月-83	1500	500	30
4421	WATSON	SALESMAN	7698	25-8月-84	1350	400	30
4566	JOHN	ANALYST	7788	23-4月-83	3500		20
4854	LARRY	SALESMAN	7698	25-9月-82	1350	1600	30
4695	BOB	SALESMAN	7698	01-9月-83	3050	1300	30
4682	MAY	CLERK	7782	19-10月-81	2050		10
4845	PETER	SALESMAN	7698	10-9月-85	1600	0	30
4877	CASSANDRA	CLERK	7788	13-12月-85	1000		20
4903	JADE	CLERK	7698	03-11月-81	1950		30
4932	IRENE	ANALYST	7566	04-10月-81	2900		20
4921	ROSE	CLERK	7782	03-11月-82	1350		10
5559	NED	MANAGER	7839	11-12月-81	2800		40
5599	ATARI	CLERK	5559	21-6月-81	1650		40
5521	ZEN	CLERK	5559	22-12月-83	1250		40
5566	SNOW	CLERK	5559	22-4月-82	2975		40

已选择27行。

2024.4.17


```
select empno,ename,deptno,hiredate,sal,  
avg(sal) over (partition by deptno order by hiredate) avg_sal,  
sum(sal) over (partition by deptno order by hiredate) sum_sal,  
max(sal) over (partition by deptno order by hiredate) max_sal,  
count(sal) over (partition by deptno order by hiredate) count_sal  
from emp;
```

结果以及解释



中山大學
SUN YAT-SEN UNIVERSITY

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	AVG_SAL	SUM_SAL	MAX_SAL	COUNT_SAL
7782	CLARK	10	09-6月 -81	2450	2450	2450	2450	1
4682	MAY	10	19-10月 -81	2050	2250	4500	2450	2
7839	KING	10	17-11月 -81	5000	3166.66667	9500	5000	3
7934	MILLER	10	23-1月 -82	1300	2700	10800	5000	4
4921	ROSE	10	03-11月 -82	1350	2430	12150	5000	5
7369	SMITH	20	17-12月 -80	800	800	800	800	1
7566	JONES	20	02-4月 -81	2975	1887.5	3775	2975	2
4932	IRENE	20	04-10月 -81	2900	2225	6675	2975	3
7902	FORD	20	03-12月 -81	3000	2418.75	9675	3000	4
4566	JOHN	20	23-4月 -83	3500	2635	13175	3500	5
4877	CASSANDRA	20	13-12月 -85	1000	2362.5	14175	3500	6
7499	ALLEN	30	20-2月 -81	1600	1600	1600	1600	1
7521	WARD	30	22-2月 -81	1250	1425	2850	1600	2
7698	BLAKE	30	01-5月 -81	2850	1900	5700	2850	3
7844	TURNER	30	08-9月 -81	1500	1800	7200	2850	4
7654	MARTIN	30	28-9月 -81	1250	1690	8450	2850	5
4903	JADE	30	03-11月 -81	1950	1733.33333	10400	2850	6
7900	JAMES	30	03-12月 -81	950	1621.42857	11350	2850	7
4854	LARRY	30	25-9月 -82	1350	1587.5	12700	2850	8
4695	BOB	30	01-9月 -83	3050	1750	15750	3050	9
4499	ALEXANDER	30	20-12月 -83	1500	1725	17250	3050	10
4421	WATSON	30	25-8月 -84	1350	1690.90909	18600	3050	11
4845	PETER	30	10-9月 -85	1600	1683.33333	20200	3050	12
5599	ATARI	40	21-6月 -81	1650	1650	1650	1650	1
5559	NED	40	11-12月 -81	2800	2225	4450	2800	2
5566	SNOW	40	22-4月 -82	2975	2475	7425	2975	3
5521	ZEN	40	22-12月 -83	1250	2168.75	8675	2975	4

2024.4.17

- $\langle \text{分析函数名} \rangle (\text{参数1}, \text{参数2} \dots) \text{ over}$ (分析短语)
- 使用 “over” 关键字是分析函数的标志

- 分区短语 (partition by)
- 排序短语 (order by)
- 开窗短语



实验：不同的分区条件混合

```
select empno,ename,deptno,hiredate,sal,  
avg(sal) over (partition by deptno order by hiredate) avg_sal,  
sum(sal) over (partition by deptno order by hiredate) sum_sal,  
job,  
count(sal) over (partition by job order by hiredate) count_sal  
from emp;
```

结果



EMPNO	ENAME	DEPTNO	HIREDATE	SAL	AVG_SAL	SUM_SAL	JOB	COUNT_SAL
7782	CLARK	10	09-6月 -81	2450	2450	2450	MANAGER	3
4682	MAY	10	19-10月 -81	2050	2250	4500	CLERK	3
7839	KING	10	17-11月 -81	5000	3166.66667	9500	PRESIDENT	1
7934	MILLER	10	23-1月 -82	1300	2700	10800	CLERK	6
4921	ROSE	10	03-11月 -82	1350	2430	12150	CLERK	8
7369	SMITH	20	17-12月 -80	800	800	800	CLERK	1
7566	JONES	20	02-4月 -81	2975	1887.5	3775	MANAGER	1
4932	IRENE	20	04-10月 -81	2900	2225	6675	ANALYST	1
7902	FORD	20	03-12月 -81	3000	2418.75	9675	ANALYST	2
4566	JOHN	20	23-4月 -83	3500	2635	13175	ANALYST	3
4877	CASSANDRA	20	13-12月 -85	1000	2362.5	14175	CLERK	10
7499	ALLEN	30	20-2月 -81	1600	1600	1600	SALESMAN	1
7521	WARD	30	22-2月 -81	1250	1425	2850	SALESMAN	2
7698	BLAKE	30	01-5月 -81	2850	1900	5700	MANAGER	2
7844	TURNER	30	08-9月 -81	1500	1800	7200	SALESMAN	3
7654	MARTIN	30	28-9月 -81	1250	1690	8450	SALESMAN	4
4903	JADE	30	03-11月 -81	1950	1733.33333	10400	CLERK	4
7900	JAMES	30	03-12月 -81	950	1621.42857	11350	CLERK	5
4854	LARRY	30	25-9月 -82	1350	1587.5	12700	SALESMAN	5
4695	BOB	30	01-9月 -83	3050	1750	15750	SALESMAN	6
4499	ALEXANDER	30	20-12月 -83	1500	1725	17250	SALESMAN	7
4421	WATSON	30	25-8月 -84	1350	1690.90909	18600	SALESMAN	8
4845	PETER	30	10-9月 -85	1600	1683.33333	20200	SALESMAN	9
5599	ATARI	40	21-6月 -81	1650	1650	1650	CLERK	2
5559	NED	40	11-12月 -81	2800	2225	4450	MANAGER	4
5566	SNOW	40	22-4月 -82	2975	2475	7425	CLERK	7
5521	ZEN	40	22-12月 -83	1250	2168.75	8675	CLERK	9

已选择27行。

2024.4.17

```
select empno,ename,deptno,hiredate,sal,  
job,  
count(sal) over (partition by job order by hiredate) count_sal  
from emp;
```

结果



EMPNO	ENAME	DEPTNO	HIREDATE	SAL	JOB	COUNT_SAL
4932	IRENE	20	04-10月-81	2900	ANALYST	1
7902	FORD	20	03-12月-81	3000	ANALYST	2
4566	JOHN	20	23-4月-83	3500	ANALYST	3
7369	SMITH	20	17-12月-80	800	CLERK	1
5599	ATARI	40	21-6月-81	1650	CLERK	2
4682	MAY	10	19-10月-81	2050	CLERK	3
4903	JADE	30	03-11月-81	1950	CLERK	4
7900	JAMES	30	03-12月-81	950	CLERK	5
7934	MILLER	10	23-1月-82	1300	CLERK	6
5566	SNOW	40	22-4月-82	2975	CLERK	7
4921	ROSE	10	03-11月-82	1350	CLERK	8
5521	ZEN	40	22-12月-83	1250	CLERK	9
4877	CASSANDRA	20	13-12月-85	1000	CLERK	10
7566	JONES	20	02-4月-81	2975	MANAGER	1
7698	BLAKE	30	01-5月-81	2850	MANAGER	2
7782	CLARK	10	09-6月-81	2450	MANAGER	3
5559	NED	40	11-12月-81	2800	MANAGER	4
7839	KING	10	17-11月-81	5000	PRESIDENT	1
7499	ALLEN	30	20-2月-81	1600	SALESMAN	1
7521	WARD	30	22-2月-81	1250	SALESMAN	2
7844	TURNER	30	08-9月-81	1500	SALESMAN	3
7654	MARTIN	30	28-9月-81	1250	SALESMAN	4
4854	LARRY	30	25-9月-82	1350	SALESMAN	5
4695	BOB	30	01-9月-83	3050	SALESMAN	6
4499	ALEXANDER	30	20-12月-83	1500	SALESMAN	7
4421	WATSON	30	25-8月-84	1350	SALESMAN	8
4845	PETER	30	10-9月-85	1600	SALESMAN	9

已选择27行。

2024.4.17



开窗短语: rows

```
select empno,ename,deptno,hiredate,sal,  
sum(sal) over (partition by deptno order by hiredate  
rows between 1 preceding and current row) sum_sal  
from emp;
```

用于指出在分区内的统计范围（基于数前后行数）

无论怎样分区，以及怎样制定开窗，分析函数的统计范围都不超出所在分区以外

结果解读



中山大學
SUN YAT-SEN UNIVERSITY

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	2450
4682	MAY	10	19-10月 -81	2050	4500
7839	KING	10	17-11月 -81	5000	7050
7934	MILLER	10	23-1月 -82	1300	6300
4921	ROSE	10	03-11月 -82	1350	2650
7369	SMITH	20	17-12月 -80	800	800
7566	JONES	20	02-4月 -81	2975	3775
4932	IRENE	20	04-10月 -81	2900	5875
7902	FORD	20	03-12月 -81	3000	5900
4566	JOHN	20	23-4月 -83	3500	6500
4877	CASSANDRA	20	13-12月 -85	1000	4500
7499	ALLEN	30	20-2月 -81	1600	1600
7521	WARD	30	22-2月 -81	1250	2850
7698	BLAKE	30	01-5月 -81	2850	4100
7844	TURNER	30	08-9月 -81	1500	4350
7654	MARTIN	30	28-9月 -81	1250	2750
4903	JADE	30	03-11月 -81	1950	3200
7900	JAMES	30	03-12月 -81	950	2900
4854	LARRY	30	25-9月 -82	1350	2300
4695	BOB	30	01-9月 -83	3050	4400
4499	ALEXANDER	30	20-12月 -83	1500	4550
4421	WATSON	30	25-8月 -84	1350	2850
4845	PETER	30	10-9月 -85	1600	2950
5599	ATARI	40	21-6月 -81	1650	1650
5559	NED	40	11-12月 -81	2800	4450
5566	SNOW	40	22-4月 -82	2975	5775
5521	ZEN	40	22-12月 -83	1250	4225

2024.4.17

测试: n preceding



```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between 1 preceding and 3 preceding) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	
4682	MAY	10	19-10月 -81	2050	
7839	KING	10	17-11月 -81	5000	
7934	MILLER	10	23-1月 -82	1300	
4921	ROSE	10	03-11月 -82	1350	
7369	SMITH	20	17-12月 -80	800	
7566	JONES	20	02-4月 -81	2975	
4932	IRENE	20	04-10月 -81	2900	
7902	FORD	20	03-12月 -81	3000	
4566	JOHN	20	23-4月 -83	3500	
4877	CASSANDRA	20	13-12月 -85	1000	
7499	ALLEN	30	20-2月 -81	1600	
7521	WARD	30	22-2月 -81	1250	
7698	BLAKE	30	01-5月 -81	2850	
7844	TURNER	30	08-9月 -81	1500	
7654	MARTIN	30	28-9月 -81	1250	

2024.4.17

测试: n preceding



```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between 3 preceding and 1 preceding) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	
4682	MAY	10	19-10月-81	2050	2450
7839	KING	10	17-11月-81	5000	4500
7934	MILLER	10	23-1月 -82	1300	9500
4921	ROSE	10	03-11月-82	1350	8350
7369	SMITH	20	17-12月-80	800	
7566	JONES	20	02-4月 -81	2975	800
4932	IRENE	20	04-10月-81	2900	3775
7902	FORD	20	03-12月-81	3000	6675
4566	JOHN	20	23-4月 -83	3500	8875
4877	CASSANDRA	20	13-12月-85	1000	9400
7499	ALLEN	30	20-2月 -81	1600	
7521	WARD	30	22-2月 -81	1250	1600
7698	BLAKE	30	01-5月 -81	2850	2850
7844	TURNER	30	08-9月 -81	1500	5700
7654	MARTIN	30	28-9月 -81	1250	5600
4903	JADE	30	03-11月-81	1950	5600

2024.4.17

测试: following



```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between 1 preceding and 1 following) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	4500
4682	MAY	10	19-10月-81	2050	9500
7839	KING	10	17-11月-81	5000	8350
7934	MILLER	10	23-1月 -82	1300	7650
4921	ROSE	10	03-11月-82	1350	2650
7369	SMITH	20	17-12月-80	800	3775
7566	JONES	20	02-4月 -81	2975	6675
4932	IRENE	20	04-10月-81	2900	8875
7902	FORD	20	03-12月-81	3000	9400
4566	JOHN	20	23-4月 -83	3500	7500
4877	CASSANDRA	20	13-12月-85	1000	4500
7499	ALLEN	30	20-2月 -81	1600	2850
7521	WARD	30	22-2月 -81	1250	5700
7698	BLAKE	30	01-5月 -81	2850	5600
7844	TURNER	30	08-9月 -81	1500	5600
7654	MARTIN	30	28-9月 -81	1250	4700

2024.4.17

测试: unbounded preceding



```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between unbounded preceding and 1 following) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	4500
4682	MAY	10	19-10月-81	2050	9500
7839	KING	10	17-11月-81	5000	10800
7934	MILLER	10	23-1月 -82	1300	12150
4921	ROSE	10	03-11月-82	1350	12150
7369	SMITH	20	17-12月-80	800	3775
7566	JONES	20	02-4月 -81	2975	6675
4932	IRENE	20	04-10月-81	2900	9675
7902	FORD	20	03-12月-81	3000	13175
4566	JOHN	20	23-4月 -83	3500	14175
4877	CASSANDRA	20	13-12月-85	1000	14175
7499	ALLEN	30	20-2月 -81	1600	2850
7521	WARD	30	22-2月 -81	1250	5700
7698	BLAKE	30	01-5月 -81	2850	7200
7844	TURNER	30	08-9月 -81	1500	8450
7654	MARTIN	30	28-9月 -81	1250	10400
4903	JADE	30	03-11月-81	1950	11350
7900	JAMES	30	03-12月-81	950	12700

2024.4.17

测试: unbounded following



```
SQL> select empno, ename, deptno, hiredate, sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between 2 preceding and unbounded following) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	12150
4682	MAY	10	19-10月 -81	2050	12150
7839	KING	10	17-11月 -81	5000	12150
7934	MILLER	10	23-1月 -82	1300	9700
4921	ROSE	10	03-11月 -82	1350	7650
7369	SMITH	20	17-12月 -80	800	14175
7566	JONES	20	02-4月 -81	2975	14175
4932	IRENE	20	04-10月 -81	2900	14175
7902	FORD	20	03-12月 -81	3000	13375
4566	JOHN	20	23-4月 -83	3500	10400
4877	CASSANDRA	20	13-12月 -85	1000	7500
7499	ALLEN	30	20-2月 -81	1600	20200
7521	WARD	30	22-2月 -81	1250	20200
7698	BLAKE	30	01-5月 -81	2850	20200
7844	TURNER	30	08-9月 -81	1500	18600
7654	MARTIN	30	28-9月 -81	1250	17350

2024.4.17



没有开窗短语时的缺省

```
select empno,ename,deptno,hiredate,sal,  
sum(sal) over (partition by deptno order by hiredate  
rows between unbounded preceding and current row) sum_sal  
from emp;
```

```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 rows between unbounded preceding and current row) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	2450
4682	MAY	10	19-10月-81	2050	4500
7839	KING	10	17-11月-81	5000	9500
7934	MILLER	10	23-1月 -82	1300	10800
4921	ROSE	10	03-11月-82	1350	12150
7369	SMITH	20	17-12月-80	800	800
7566	JONES	20	02-4月 -81	2975	3775
4932	IRENE	20	04-10月-81	2900	6675

测试：在开窗短语中使用子查询

```
SQL> select empno, ename, deptno, hiredate, sal,
2  sum(sal) over (partition by deptno order by hiredate
3  rows between 1 preceding and (select count(*) from emp)/10 following) sum_sal
4  from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	9500
4682	MAY	10	19-10月-81	2050	10800
7839	KING	10	17-11月-81	5000	9700
7934	MILLER	10	23-1月 -82	1300	7650
4921	ROSE	10	03-11月-82	1350	2650
7369	SMITH	20	17-12月-80	800	6675
7566	JONES	20	02-4月 -81	2975	9675
4932	IRENE	20	04-10月-81	2900	12375
7902	FORD	20	03-12月-81	3000	10400
4566	JOHN	20	23-4月 -83	3500	7500
4877	CASSANDRA	20	13-12月-85	1000	4500
7499	ALLEN	30	20-2月 -81	1600	5700
7521	WARD	30	22-2月 -81	1250	7200
7698	BLAKE	30	01-5月 -81	2850	6850
7844	TURNER	30	08-9月 -81	1500	7550
7654	MARTIN	30	28-9月 -81	1250	5650
4903	JADE	30	03-11月-81	1950	5500

2024.4.17



开窗短语: range

```
select empno,ename,deptno,hiredate,sal,  
sum(sal) over (partition by deptno order by hiredate  
range between unbounded preceding and 365 following) sum_sal  
from emp;
```

从本分区最开始的行一直累加到hiredate+365（值范围）的行

结果



```
SQL> select empno,ename,deptno,hiredate,sal,  
2 sum(sal) over (partition by deptno order by hiredate  
3 range between unbounded preceding and 365 following) sum_sal  
4 from emp;
```

EMPNO	ENAME	DEPTNO	HIREDATE	SAL	SUM_SAL
7782	CLARK	10	09-6月 -81	2450	10800
4682	MAY	10	19-10月 -81	2050	10800
7839	KING	10	17-11月 -81	5000	12150
7934	MILLER	10	23-1月 -82	1300	12150
4921	ROSE	10	03-11月 -82	1350	12150
7369	SMITH	20	17-12月 -80	800	9675
7566	JONES	20	02-4月 -81	2975	9675
4932	IRENE	20	04-10月 -81	2900	9675
7902	FORD	20	03-12月 -81	3000	9675
4566	JOHN	20	23-4月 -83	3500	13175
4877	CASSANDRA	20	13-12月 -85	1000	14175
7499	ALLEN	30	20-2月 -81	1600	11350
7521	WARD	30	22-2月 -81	1250	11350
7698	BLAKE	30	01-5月 -81	2850	11350
7844	TURNER	30	08-9月 -81	1500	11350
7654	MARTIN	30	28-9月 -81	1250	12700

2024.4.17



中山大學
SUN YAT-SEN UNIVERSITY

Thanks

FAQ时间