

课程介绍

学习目标



数据统计分析

聚合函数、分组查询、HAVING子句



多表连接查询

内连接、外连接，以及多表查询的多种语法



子查询

单行子查询、多行子查询、WHERE子查询、FROM子查询、SELECT子查询



数据操作语言：聚合函数

什么是聚合函数

- ◆ 聚合函数在数据的查询分析中，应用十分广泛。聚合函数可以对数据求和、求最大值和最小值、求平均值等等。
- ◆ 求公司员工的平均月收入是多少？

```
SELECT AVG(sal+IFNULL(comm,0)) FROM t_emp;
```

SUM函数

- ◆ SUM函数用于求和，只能用于数字类型，字符类型的统计结果为0，日期类型统计结果是毫秒数相加。

```
SELECT SUM(ename) FROM t_emp;
```

```
SELECT SUM(sal) FROM t_emp  
WHERE deptno IN (10, 20);
```

- ◆ SUM函数求和会排除NULL值

```
SELECT SUM(comm) FROM t_emp;
```

MAX函数

- ◆ MAX函数用于获得非空值的最大值。

```
SELECT MAX(comm) FROM t_emp;
```

- ◆ 问题1：查询10和20部门中，月收入最高的员工？

```
SELECT MAX(sal+IFNULL(comm,0)) FROM t_emp  
WHERE deptno IN(10,20);
```

- ◆ 问题2：查询员工名字最长的是几个字符？

```
SELECT MAX(LENGTH(ename)) FROM t_emp;
```

MIN函数

- ◆ MIN函数用于获得非空值的最小值。

```
SELECT MIN(empno) FROM t_emp;
```

```
SELECT MIN(hiredate) FROM t_emp;
```

AVG函数

- ◆ AVG函数用于获得非空值的平均值，非数字数据统计结果为0

```
SELECT AVG (sal+IFNULL (comm, 0)) FROM t_emp;
```

```
SELECT AVG (ename) FROM t_emp;
```


COUNT函数

- ◆ COUNT(*)用于获得包含空值的记录数，COUNT(列名)用于获得包含非空值的记录数。

```
SELECT COUNT (*) FROM t_emp;
```

```
SELECT COUNT(comm) FROM t_emp;
```

COUNT函数

- ◆ 查询10和20部门中，底薪超过2000元并且工龄超过15年的员工人数？

```
SELECT COUNT(*) FROM t_emp  
WHERE deptno IN(10,20) AND sal>=2000  
AND DATEDIFF(NOW(),hiredate)/365>=15;
```

- ◆ 查询1985年以后入职的员工，底薪超过公司平均底薪的员工数量？

```
SELECT COUNT(*) FROM t_emp  
WHERE hiredate>="1985-01-01"  
AND sal>AVG(sal);
```

数据操作语言：分组查询

为什么要分组？

- ◆ 默认情况下汇总函数是对全表范围内的数据做统计
- ◆ GROUP BY子句的作用是通过一定的规则将一个数据集划分成若干个小的区域，然后针对每个小区域分别进行数据汇总处理

```
SELECT deptno, AVG(sal) FROM t_emp  
GROUP BY deptno;
```

deptno	AVG(sal)
20	2175.000000
30	1566.666667
10	2916.666667

} 每个部门的平均工资

逐级分组

- ◆ 数据库支持多列分组条件，执行的时候逐级分组。
- ◆ 查询每个部门里，每种职位的人员数量和平均底薪

```
SELECT deptno, job, COUNT (*), AVG (sal)
FROM t_emp GROUP BY deptno, job
ORDER BY deptno;
```

对SELECT子句的要求

- ◆ 查询语句中如果含有GROUP BY子句，那么SELECT子句中的内容就必须遵守规定：SELECT子句中可以包括聚合函数，或者GROUP BY子句的分组列，其余内容均不可以出现在SELECT子句中

```
SELECT deptno, COUNT (*), AVG (sal)
FROM t_emp GROUP BY deptno;
```

```
SELECT deptno, COUNT (*), AVG (sal), sal
FROM t_emp GROUP BY deptno;
```

对分组结果集再次做汇总计算

```
SELECT  
    deptno, COUNT(*), AVG(sal), MAX(sal), MIN(sal)  
FROM t_emp GROUP BY deptno WITH ROLLUP;
```

deptno	COUNT(*)	AVG(sal)	MAX(sal)	MIN(sal)
10	3	2916.666667	5000.00	1300.00
20	5	2175.000000	3000.00	800.00
30	6	1566.666667	2850.00	950.00
(Null)	14	2073.214286	5000.00	800.00

GROUP_CONCAT函数

- ◆ GROUP_CONCAT函数可以把分组查询中的某个字段拼接成一个字符串
- ◆ 查询每个部门内底薪超过2000元的人数和员工姓名

```
SELECT deptno, GROUP_CONCAT(ename), COUNT(*)  
FROM t_emp WHERE sal >= 2000  
GROUP BY deptno;
```

deptno	GROUP_CONCAT(ename)	COUNT(*)
10	CLARK,KING	2
20	JONES,SCOTT,FORD	3
30	BLAKE	1

} 查询结果集

各种子句的执行顺序

◆ 查询语句中，GROUP BY子句应该第几个执行？

**FROM -> WHERE -> GROUP BY -> SELECT -> ORDER BY
-> LIMIT**

数据操作语言：HAVING子句

分组查询遇到的困难？

- ◆ 查询部门平均底薪超过2000元的部门编号

```
SELECT deptno FROM t_emp  
WHERE AVG(sal) >= 2000  
GROUP BY deptno;
```

- ◆ 因为WHERE子句先于GROUP BY执行，一旦WHERE子句中出现了汇总函数，数据库根本不知道按照什么范围计算汇总值

HAVING子句的用途

- ◆ 查询每个部门中，1982年以后入职的员工超过2个人的部门编号

```
SELECT deptno FROM t_emp  
WHERE hiredate>="1982-01-01"  
GROUP BY deptno HAVING COUNT(*)>=2;  
ORDER BY deptno ASC;
```

HAVING子句的特殊用法

- ◆ 按照数字1分组，MySQL会依据SELECT子句中的列进行分组，HAVING子句也可以正常使用

```
SELECT deptno, COUNT(*) FROM t_emp  
GROUP BY 1;
```

```
SELECT deptno, COUNT(*) FROM t_emp  
GROUP BY 1 HAVING deptno IN (10, 20);
```

数据操作语言：表连接查询（一）

从多张表中提取数据

- ◆ 从多张表中提取数据，必须指定关联的条件。如果不定义关联条件就会出现无条件连接，两张表的数据会交叉连接，产生笛卡尔积。

```
SELECT empno,ename,dname  
FROM t_emp JOIN t_dept;
```

只读	查询时间: 0.057s	第 1 条记录 (共 56 条)
----	--------------	------------------

从多张表中提取数据

- ◆ 规定了连接条件的表连接语句，就不会出现笛卡尔积

```
SELECT e.empno, e.ename, d.dname  
FROM t_emp e JOIN t_dept d  
ON e.deptno=d.deptno;
```

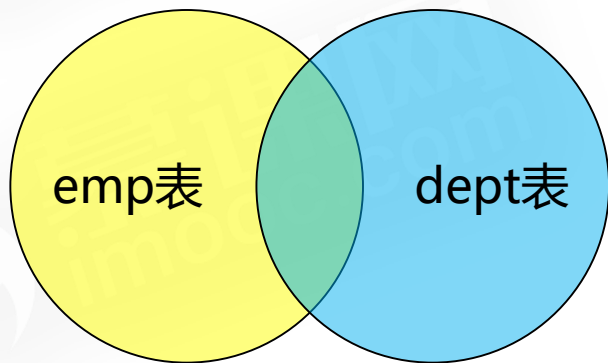
只读	查询时间: 0.113s	第 1 条记录 (共 14 条)
----	--------------	------------------

表连接的分类

- ◆ 表连接分为两种：内连接和外连接
- ◆ 内连接是结果集中只保留符合连接条件的记录
- ◆ 外连接是不管符不符合连接条件，记录都要保留在结果集中

内连接的简介

- ◆ 内连接是最常见的一种表连接，用于查询多张关系表符合连接条件的记录。



```
SELECT ..... FROM 表1  
[INNER] JOIN 表2 ON 条件  
[INNER] JOIN 表3 ON 条件  
.....
```

内连接的多种语法形式

```
SELECT ..... FROM 表1 JOIN 表2 ON 连接条件 ;
```

```
SELECT ..... FROM 表1 JOIN 表2 WHERE 连接条件 ;
```

```
SELECT ..... FROM 表1 , 表2 WHERE 连接条件 ;
```

内连接练习1

- ◆ 查询每个员工的工号、姓名、部门名称、底薪、职位、工资等级？

```
SELECT
    e.empno,e.ename,d.deptno,e.sal,e.job,s.grade
FROM t_emp e
JOIN t_dept d ON e.deptno=d.deptno
JOIN t_salgrade s ON e.sal BETWEEN s.losal AND s.hisal;
```

- ◆ 内连接的数据表不一定必须有同名字段，只要字段之间符合逻辑关系就可以

内连接练习2

- ◆ 查询与SCOTT相同部门的员工都有谁?

```
SELECT e2.ename  
FROM t_emp e1 JOIN t_emp e2  
ON e1.deptno=e2.deptno  
WHERE e1.ename="SCOTT" AND e2.ename!="SCOTT";
```

- ◆ 相同的数据表也可以做表连接

内连接练习3

- ◆ 查询月薪超过公司平均月薪的员工信息？

```
SELECT  
    e1.ename  
FROM  
    t_emp e1 JOIN  
    (SELECT AVG(sal) avg FROM t_emp) e2  
ON e1.sal > e2.avg;
```

- ◆ 结果集也可以作为一张“表”来跟其他表连接

数据操作语言：表连接查询（二）

内连接查询练习1

- ◆ 查询底薪超过公司平均底薪的员工信息？
- ◆ 查询RESEARCH部门的人数、最高底薪、最低底薪、平均底薪、平均工龄？

内连接查询练习2

- ◆ 查询每种职业的最高工资、最低工资、平均工资、最高工资等级和最低工资等级？
- ◆ 查询每个底薪超过部门平均底薪的员工信息

内连接查询练习3

- ◆ 查询每个底薪超过部门平均底薪的员工信息

```
SELECT
    e.empno, e.ename, e.sal, t.avg
FROM t_emp e JOIN
    (SELECT deptno, AVG(sal) as avg
     FROM t_emp GROUP BY deptno) t
ON e.deptno=t.deptno AND e.sal>=t.avg;
```

数据操作语言：表连接查询（三）

为什么要使用外连接

- ◆ 如果说陈浩是一名临时人员，没有固定的部门编制，那么我们想查询每名员工和他的部门名称，用内连接就会漏掉陈浩，所以要引入外连接的语法才能解决这个问题

7902	FORD	ANALYST	7566	1981-12-03	3000	(Null)	20
7934	MILLER	CLERK	7782	1982-01-23	1300	(Null)	10
8000	陈浩	SALESMAN	7698	1982-07-19	1550	(Null)	(Null)

外连接简介

- ◆ 外连接与内连接的区别在于，除了符合条件的记录之外，结果集中还会保留不符合条件的记录。

```
SELECT
```

```
    e.empno, e.ename, d.dname
```

```
FROM
```

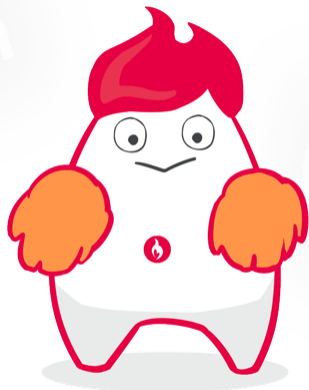
```
    t_emp e
```

```
LEFT JOIN t_dept d ON e.deptno = d.deptno;
```

	7902 FORD	RESEARCH
	7934 MILLER	ACCOUNTIN
▶	8000 陈浩	(Null)

左连接和右连接

- ◆ 左外连接就是保留左表所有的记录，与右表做连接。如果右表有符合条件的记录就与左表连接。如果右表没有符合条件的记录，就用NULL与左表连接。右外连接也是如此。



外连接练习1

- ◆ 查询每个部门的名称和部门的人数？
- ◆ 查询每个部门的名称和部门的人数？如果没有部门的员工，部门名称用NULL代替。
- ◆ UNION关键字可以将多个查询语句的结果集进行合并

(查询语句) UNION (查询语句) UNION (查询语句)

外连接练习2

- ◆ 查询每名员工的编号、姓名、部门、月薪、工资等级、工龄、上司编号、上司姓名、上司部门？

```
SELECT
    e1.empno, e1.ename, d1.dname, e1.sal, t.grade,
    e1.mgr, e2.ename, d2.dname
FROM t_emp e1 LEFT JOIN t_dept d1
ON e1.deptno=d1.deptno
LEFT JOIN t_emp e2 ON e1.mgr=e2.empno
LEFT JOIN t_salgrade t
ON e1.sal BETWEEN t.losal AND t.hisal
LEFT JOIN t_dept d2 ON e2.deptno=d2.deptno;
```


数据操作语言：表连接查询（四）

作业练习

- ◆ 查询每名员工的编号、姓名、部门、月薪、工资等级、工龄、上司编号、上司姓名、上司部门？



外连接的注意事项

- ◆ 内连接只保留符合条件的记录，所以查询条件写在ON子句和WHERE子句中的效果是相同的。但是外连接里，条件写在WHERE子句里，不符合条件的记录是会被过滤掉的，而不是保留下来。

ename	dname	deptno
CLARK	ACCOUNTIN	10
KING	ACCOUNTIN	10
MILLER	ACCOUNTIN	10
BLAKE	(Null)	(Null)
CLARK	ACCOUNTIN	10

数据操作语言：子查询（一）

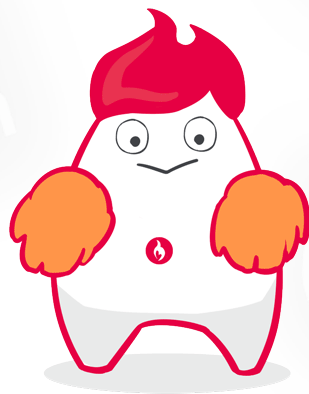
子查询简介

- ◆ 子查询是一种查询中嵌套查询的语句
- ◆ 查询底薪超过公司平均底薪的员工的信息

```
SELECT  
    empno,ename,sal  
FROM t_emp  
WHERE sal>=(SELECT AVG(sal) FROM t_emp);
```

子查询的分类

- ◆ 子查询可以写在三个地方：WHERE子句、FROM子句、SELECT子句，但是只有FROM子句子查询是最可取的



WHERE子查询

- ◆ 这种子查询最简单，最容易理解，但是却是效率很低的子查询
- ◆ 查询底薪超过公司平均底薪的员工的信息

```
SELECT  
    empno,ename,sal  
FROM t_emp  
WHERE sal>=(SELECT AVG(sal) FROM t_emp);
```

比较每条记录都要重新执行子查询



用表连接替代WHERE子查询

- ◆ 表连接的优点是子查询只执行一次，查询效率特别高

```
SELECT e2.empno,e2.ename  
FROM t_emp e1 JOIN t_emp e2  
ON e1.deptno=e2.deptno  
AND e2.ename!="FORD"  
AND e1.ename="FORD";
```


FROM子查询

- ◆ 这种子查询只会执行一次，所以查询效率很高

```
SELECT
    e.empno,e.ename,e.sal,t.avg
FROM t_emp e JOIN
    (SELECT deptno,AVG(sal) as avg
     FROM t_emp GROUP BY deptno) t
ON e.deptno=t.deptno AND e.sal>=t.avg;
```

SELECT子查询

- ◆ 这种子查询每输出一条记录的时候都要执行一次，查询效率很低

```
SELECT
    e.empno,
    e.ename,
    (SELECT dname FROM t_dept WHERE deptno=e.deptno)
FROM t_emp e;
```

相关子查询

- ◆ 查询语句执行的时候要多次的依赖于子查询的结果，这类子查询被称作相关子查询
- ◆ WHERE子查询和SELECT子查询都属于相关子查询
- ◆ 因为相关子查询要反复多次执行，所以应该避免使用

数据操作语言：子查询（二）

单行子查询和多行子查询

- ◆ 单行子查询的结果集只有一条记录，多行子查询结果集有多行记录
- ◆ 多行子查询只能出现在WHERE子句和FROM子句中
- ◆ 如何用子查询查找FORD和MARTIN两个人的同事？

WHERE子句中的多行子查询

- ◆ WHERE子句中，可以使用IN、ALL、ANY、EXISTS关键字来处理多行表达式结果集的条件判断
- ◆ 查询比FORD和MARTIN底薪都高的员工信息？

```
SELECT  ename FROM t_emp
WHERE   sal > ALL
        (SELECT sal FROM t_emp
         WHERE ename IN ("FORD", "MARTIN")) ;
```

EXISTS关键字

- ◆ EXISTS关键字是把原来在子查询之外的条件判断，写到了子查询的里面。

```
SELECT ..... FROM 表名 WHERE [NOT] EXISTS (子查询);
```

EXISTS关键字

- ◆ 查询工资等级3和4级的员工信息?

```
SELECT
    empno,ename,sal,comm
FROM t_emp
WHERE EXISTS
    (SELECT * FROM t_salgrade WHERE grade IN (3,4)
    AND sal BETWEEN lsal AND hisal);
```


课程总结

技能清单

技能知识1



技能知识2



技能知识3



技能知识4



掌握了汇总函数和分组统计



掌握了表连接语法，并熟练运用



掌握了子查询的语法



掌握子查询和表连接的转换