## 数据查询语法（DQL）

　　DQL就是数据查询语言，数据库执行DQL语句不会对数据进行改变，而是让数据库发送结果集给客户端。

语法：

SELECT selection\_list /\*要查询的列名称\*/

FROM table\_list /\*要查询的表名称\*/

WHERE condition /\*行条件\*/

GROUP BY grouping\_columns /\*对结果分组\*/

HAVING condition /\*分组后的行条件\*/

ORDER BY sorting\_columns /\*对结果分组\*/

LIMIT offset\_start, row\_count /\*结果限定\*/

创建数据库：study1015

创建名：

* 学生表：stu

|  |  |  |
| --- | --- | --- |
| **字段名称** | **字段类型** | **说明** |
| **sid** | char(6) | 学生学号 |
| **sname** | varchar(50) | 学生姓名 |
| **age** | int | 学生年龄 |
| **gender** | varchar(50) | 学生性别 |

INSERT INTO stu VALUES('S\_1001', 'liuYi', 35, 'male');

INSERT INTO stu VALUES('S\_1002', 'chenEr', 15, 'female');

INSERT INTO stu VALUES('S\_1003', 'zhangSan', 95, 'male');

INSERT INTO stu VALUES('S\_1004', 'liSi', 65, 'female');

INSERT INTO stu VALUES('S\_1005', 'wangWu', 55, 'male');

INSERT INTO stu VALUES('S\_1006', 'zhaoLiu', 75, 'female');

INSERT INTO stu VALUES('S\_1007', 'sunQi', 25, 'male');

INSERT INTO stu VALUES('S\_1008', 'zhouBa', 45, 'female');

INSERT INTO stu VALUES('S\_1009', 'wuJiu', 85, 'male');

INSERT INTO stu VALUES('S\_1010', 'zhengShi', 5, 'female');

INSERT INTO stu VALUES('S\_1011', 'xxx', NULL, NULL);

* 雇员表：emp

|  |  |  |
| --- | --- | --- |
| **字段名称** | **字段类型** | **说明** |
| **empno** | int | 员工编号 |
| **ename** | varchar(50) | 员工姓名 |
| **job** | varchar(50) | 员工工作 |
| **mgr** | int | 领导编号 |
| **hiredate** | date | 入职日期 |
| **sal** | decimal(7,2) | 月薪 |
| **comm** | decimal(7,2) | 奖金 |
| **deptno** | int | 部分编号 |

INSERT INTO emp values(7369,'SMITH','CLERK',7902,'1980-12-17',800,NULL,20);

INSERT INTO emp values(7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);

INSERT INTO emp values(7521,'WARD','SALESMAN',7698,'1981-02-22',1250,500,30);

INSERT INTO emp values(7566,'JONES','MANAGER',7839,'1981-04-02',2975,NULL,20);

INSERT INTO emp values(7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30);

INSERT INTO emp values(7698,'BLAKE','MANAGER',7839,'1981-05-01',2850,NULL,30);

INSERT INTO emp values(7782,'CLARK','MANAGER',7839,'1981-06-09',2450,NULL,10);

INSERT INTO emp values(7788,'SCOTT','ANALYST',7566,'1987-04-19',3000,NULL,20);

INSERT INTO emp values(7839,'KING','PRESIDENT',NULL,'1981-11-17',5000,NULL,10);

INSERT INTO emp values(7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);

INSERT INTO emp values(7876,'ADAMS','CLERK',7788,'1987-05-23',1100,NULL,20);

INSERT INTO emp values(7900,'JAMES','CLERK',7698,'1981-12-03',950,NULL,30);

INSERT INTO emp values(7902,'FORD','ANALYST',7566,'1981-12-03',3000,NULL,20);

INSERT INTO emp values(7934,'MILLER','CLERK',7782,'1982-01-23',1300,NULL,10);

* 部门表：dept

|  |  |  |
| --- | --- | --- |
| **字段名称** | **字段类型** | **说明** |
| **deptno** | int | 部门编码 |
| **dname** | varchar(50) | 部门名称 |
| **loc** | varchar(50) | 部门所在地点 |

INSERT INTO dept values(10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO dept values(20, 'RESEARCH', 'DALLAS');

INSERT INTO dept values(30, 'SALES', 'CHICAGO');

INSERT INTO dept values(40, 'OPERATIONS', 'BOSTON');

###### 1　基础查询

1.1　查询所有列

1.2　查询指定列

###### 2　条件查询

2.1　条件查询介绍

条件查询就是在查询时给出WHERE子句，在WHERE子句中可以使用如下运算符及关键字：

* =、!=、<>、<、<=、>、>=；
* BETWEEN…AND；
* IN(set)；
* IS NULL；
* AND；
* OR；
* NOT；

2.2　查询性别为女，并且年龄50的记录

2.3　查询学号为S\_1001，或者姓名为liSi的记录

2.4　查询学号为S\_1001，S\_1002，S\_1003的记录

2.5　查询学号不是S\_1001，S\_1002，S\_1003的记录

2.6　查询年龄为null的记录

2.7　查询年龄在20到40之间的学生记录

2.8　查询性别非男的学生记录

2.9　查询姓名不为null的学生记录

###### 3　模糊查询

当想查询姓名中包含a字母的学生时就需要使用模糊查询了。模糊查询需要使用关键字LIKE。

3.1　查询姓名由5个字母构成的学生记录

3.2　查询姓名由5个字母构成，并且第5个字母为“i”的学生记录

3.3　查询姓名以“z”开头的学生记录

3.4　查询姓名中第2个字母为“i”的学生记录

3.5　查询姓名中包含“a”字母的学生记录

###### 4　字段控制查询

4.1　去除重复记录

去除重复记录（两行或两行以上记录中系列的上的数据都相同），例如emp表中sal字段就存在相同的记录。当只查询emp表的sal字段时，那么会出现重复记录，那么想去除重复记录，需要使用DISTINCT：

4.2　查看雇员的月薪与佣金之和

4.3　给列名添加别名

在上面查询中出现列名为sal+IFNULL(comm,0)，这很不美观，现在我们给这一列给出一个别名，为total：

###### 5　排序

5.1　查询所有学生记录，按年龄升序排序

5.2　查询所有学生记录，按年龄降序排序

5.3　查询所有雇员，按月薪降序排序，如果月薪相同时，按编号升序排序

###### 6　聚合函数

聚合函数是用来做纵向运算的函数：

* COUNT()：统计指定列不为NULL的记录行数；
* MAX()：计算指定列的最大值，如果指定列是字符串类型，那么使用字符串排序运算；
* MIN()：计算指定列的最小值，如果指定列是字符串类型，那么使用字符串排序运算；
* SUM()：计算指定列的数值和，如果指定列类型不是数值类型，那么计算结果为0；
* AVG()：计算指定列的平均值，如果指定列类型不是数值类型，那么计算结果为0；

6.1　COUNT

当需要纵向统计时可以使用COUNT()。

* 查询emp表中记录数：
* 查询emp表中有佣金的人数：
* 查询emp表中月薪大于2500的人数：
* 统计月薪与佣金之和大于2500元的人数：
* 查询有佣金的人数，以及有领导的人数：

6.2　SUM和AVG

当需要纵向求和时使用sum()函数。

* 查询所有雇员月薪和：
* 查询所有雇员月薪和，以及所有雇员佣金和：
* 查询所有雇员月薪+佣金和：
* 统计所有员工平均工资：

6.3　MAX和MIN

* 查询最高工资和最低工资：

###### 7　分组查询

当需要分组查询时需要使用GROUP BY子句，例如查询每个部门的工资和，这说明要使用部分来分组。

7.1　分组查询

* 查询每个部门的部门编号和每个部门的工资和：
* 查询每个部门的部门编号以及每个部门的人数：
* 查询每个部门的部门编号以及每个部门工资大于1500的人数：

7.2　HAVING子句

* 查询工资总和大于9000的部门编号以及工资和：

###### 8　LIMIT

LIMIT用来限定查询结果的起始行，以及总行数。

8.1　查询5行记录，起始行从0开始

8.2　查询10行记录，起始行从3开始

8.3　分页查询

如果一页记录为10条，希望查看第3页记录应该怎么查呢？

* 第一页记录起始行为0，一共查询10行；
* 第二页记录起始行为10，一共查询10行；
* 第三页记录起始行为20，一共查询10行；