

【实验】数据查询

DQL (data query language), 数据查询语言, 对应select语句实现。

【实验目的】

1. 掌握select的基本使用;
2. 掌握条件查询、模糊查询、排序的SQL语句实现;
3. 掌握聚合函数、分组查询、限定条件查询实现;
4. 掌握数据表间关系, 及主、外键约束关系;
5. 掌握联合查询, 连接查询 (内连接、外连接) 及其应用;
6. 掌握子查询及其应用;
7. 掌握综合查询应用。

【预备知识】

1. 掌握select的基本使用;
2. 掌握条件查询、模糊查询、排序的SQL语句实现;
3. 掌握聚合函数、分组查询、限定条件查询实现;
4. 掌握数据表间关系, 及主、外键约束关系;
5. 掌握联合查询, 连接查询 (内连接、外连接) 及其应用;
6. 掌握子查询及其应用;
7. 掌握综合查询应用。

【实验内容】

• 实验准备：数据库设计

```
drop database if exists employeedb
create database employeedb default character set = 'utf8mb4';
use employeedb;

CREATE TABLE departments (
  department_id INT PRIMARY KEY,
  department_name VARCHAR(50) );
INSERT INTO departments (department_id, department_name) VALUES
(1, '技术部'),
(2, '市场部'),
(3, '行政部'),
(4, '财务部'),
(5, '产品部'),
(6, '运营部'),
(7, '总经办');

CREATE TABLE employees (
  employee_id INT PRIMARY KEY,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  gender VARCHAR(10),
  age INT,
  job_title VARCHAR(50),
  salary DECIMAL(10,2),
```

```

    department_id INT,
    manager_id INT,
    foreign key (department_id) references departments(department_id) );
INSERT INTO employees (employee_id, first_name, last_name, gender, age,
job_title, salary, department_id, manager_id) VALUES
(101, '张', '三', '男', 23, '程序员', 5000, 1, NULL),
(102, '李', '四', '男', 25, '市场专员', 3500, 2, NULL),
(103, '王', '五', '女', 29, '行政助理', 3800, 3, NULL),
(104, '赵', '六', '女', 30, '财务经理', 8000, 4, NULL),
(105, '陈', '七', '男', 26, '项目经理', 6000, 5, 101),
(106, '刘', '八', '男', 28, '市场经理', 7500, 2, 102),
(107, '张', '九', '女', 27, 'HR经理', 7000, 3, 103),
(108, '赵', '十', '女', 24, '运营经理', 6500, 6, 105),
(109, '林', '十一', '男', 32, '总经理', 15000, 7, NULL),
(110, '张', '十二', '女', 28, '人事助理', 3800, 3, 107),
(111, '李', '十三', '男', 24, '市场专员', 3500, 2, 102),
(112, '王', '十四', '女', 30, '行政主管', 5500, 3, 103),
(113, '陈', '十五', '男', 26, '产品经理', 6000, 5, 101),
(114, '赵', '十六', '男', 30, '测试工程师', 5000, 1, 113),
(115, '张', '十七', '女', 27, 'UI设计师', 4500, 5, 113),
(116, '刘', '十八', '女', 28, '运营专员', 4200, 6, 108),
(117, '张', '十九', '男', 25, '前端开发工程师', 7000, 1, 113),
(118, '李', '二十', '男', 29, '后端开发工程师', 7200, 1, 113),
(119, '谷', '二十一', '男', 30, '项目助理', 3600, 5, 113),
(120, '赵', '二十二', '女', 24, '前端开发工程师', 6700, 1, 113),
(121, '李', '二十三', '男', 32, 'Java开发工程师', 7500, 1, 113),
(122, '王', '二十四', '男', 28, '人事专员', 4000, 3, 107),
(123, '陈', '二十五', '女', 29, '商务专员', 3800, 2, 102),
(124, '张', '二十六', '女', 27, '产品助理', 4200, 5, 113),
(125, '赵', '二十七', '女', 28, '测试主管', 6000, 1, 113),
(126, '陈', '二十八', '男', 30, 'QA工程师', 5000, 1, 125),
(127, '张', '二十九', '男', 27, '系统架构师', 11000, 1, 113),
(128, '赵', '三十', '女', 28, 'UI设计师', 5500, 5, 101);

alter table employees
add column e_name varchar(100);

update employees set e_name=concat(first_name,last_name);

alter table employees
drop column first_name,
drop column last_name;

```

• 实验1：简单查询

- 语法格式： `SELECT column1, column2, ... FROM table_name WHERE condition;`
- 示例：

```

SELECT * FROM employees;
SELECT * FROM employees WHERE salary>5000;

```

- 排序，语法格式：

```
SELECT column1, column2, ... FROM table_name
WHERE condition
ORDER BY column1 ASC|DESC, column2 ASC|DESC, ...;
```

以升序或者降序的方式排序结果。

- 示例：

```
SELECT * FROM employees ORDER BY salary DESC;
```

- 实训任务：

- 将所有员工的工资增加10%。
- 查询薪资大于等于1000元的员工。
- 查询薪资大于等于3000元并且年龄大于等于25岁的员工姓名、年龄和薪资。
- 查询所有男性员工的姓名和工资，要求工资大于等于2000元。
- 查询所有工资大于等于4000元的员工，按照工资从大到小排序。
- 查询部门编号为101的所有员工的姓名、年龄和薪资，按照薪资降序排序。
- 查询所有员工的姓名、性别和薪资，要求薪资不超过5000元。
- 查询所有员工的姓名、年龄和薪资，要求薪资大于等于2000元并且不超过7000元，年龄大于等于22岁并且不超过32岁，按照薪资降序排序。
- 查询员工编号为101的员工的详细信息。
- 查询部门名称为“技术部”的所有员工的姓名和工资。
- 查询年龄大于25岁的员工信息。
- 查询月薪大于20000元的员工信息。
- 查询工资位于20000到30000之间的员工信息。

• 实验2：聚合函数&分组查询

- 聚合函数：

函数	语法	说明
avg	avg(expression)	计算一组值得平均值。计算过程忽略null值
count	count(*)和count(distinct expression)	返回表中的行数
sum	sum(expression)	返回一组值的总和。计算过程忽略null值
min	min(expression)	返回一组值中的最小值
max	max(expression)	返回一组值中的最大值

- 分组查询：将一列或多列数据分组。
- 语法格式：

```
SELECT column1, column2, ... FROM table_name
WHERE condition
GROUP BY column1, column2, ...;
```

- 示例：

```
SELECT COUNT(*) FROM employees GROUP BY department_id;
```

- 实训任务：
 - 查询每种职位的平均薪资和员工数。
 - 查询各个职位的平均薪资，按薪资从高到低排序。
 - 查询所有员工的工资总和、平均工资、最高工资和最低工资。

• 实验3：联合查询

在不同的表中查询所需的数据，并把结果作为单一表输出。

- 语法格式：

```
SELECT column1, column2, ... FROM table_name1
UNION|UNION ALL
SELECT column1, column2, ... FROM table_name2 WHERE condition;
```

- 示例：

```
SELECT first_name, last_name FROM employees
UNION
SELECT first_name, last_name FROM students;
```

- union与union all的区别
 - union是查询两张表中的信息，并去除重复记录；
 - union all是查询两张表信息，并返回所有记录。
- 实训任务：
 - 查询所有员工中年龄最小和最大的员工的姓名和年龄。