清华大学数据库技术与应用

SQL II

授课教师：计算机系王健楠

授课学期：2026年（春季）

课程大纲

单表查询

- 语法
- 常用运算符：DISTINCT、ORDER BY、LIKE
- 处理缺失值：NULL

多表查询

- 外键约束
- 连接
- 集合操作

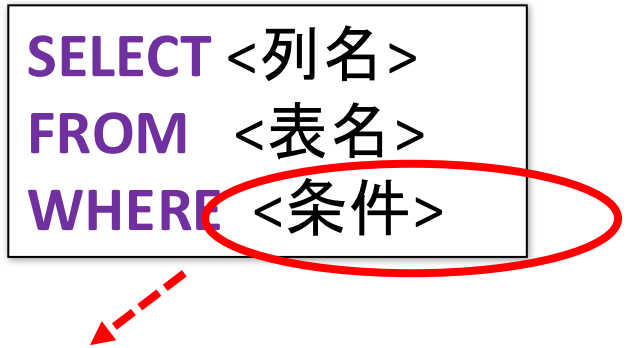
单表查询语法

```
SELECT <列名>
FROM   <表名>
WHERE  <条件>
```

写查询时，请思考以下三个问题：

- 你关注的是哪张表?
- 你关注的是哪些行?
- 你关注的是哪些列?

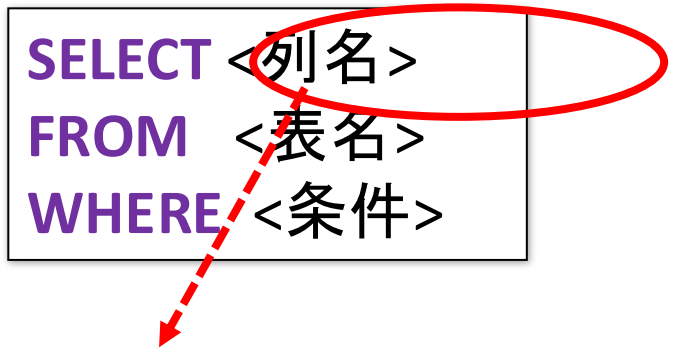
条件（WHERE 子句）



你关注的是哪些行?

- **WHERE** gpa > 3.5
- **WHERE** school = 'THU' **AND** gpa > 3.5
- **WHERE** (school = 'THU' OR school = 'PKU') **AND** gpa > 3.5
- **WHERE** age * 365 > 7500

列的选取（SELECT 子句）



你关注的是哪些列？

- **SELECT** *
- **SELECT** name, age
- **SELECT** name as studentName, age
- **SELECT** name, age * 365 as ageDay

六点注意事项

清華大學 Tsinghua University

1. SQL 关键字不区分大小写：

- 等价：SELECT，Select，select

2. SQL 标识符（表名、列名）的大小写规则取决于具体的 DBMS：

- 表名：Student，student
- 列名：gpa，GPA

3. 值是区分大小写的：

- 不同：'THU'，'thu'

4. SQL 字符串用单引号括起来

- 例如：name = 'Mike'

5. 单引号中的单引号用额外的单引号转义

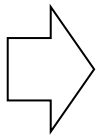
- author = 'Shaq O''Neal'

6. 字符串可以用比较运算符按字母顺序比较

- 例如：'fodder' < 'foo' 结果为 TRUE

DISTINCT：消除重复

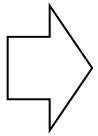
```
SELECT School
FROM   Students
```



School
THU
THU
PKU
HIT
HIT

Versus

```
SELECT DISTINCT School
FROM   Students
```



School
THU
PKU
HIT

ORDER BY：对结果排序

```
SELECT   name, gpa, age
FROM     Students
WHERE    school =   'THU'
ORDER BY gpa DESC, age ASC
```

查询结果可以按指定列进行排序：

- 支持按任意数量的列排序
- 支持升序或降序排序

默认排序方式为升序

- 在列名后添加 **ASC** 或 **DESC** 关键字指定排序方向

LIKE：简单字符串模式匹配

```
SELECT *
FROM   Students
WHERE  name LIKE 'Sm_t%'
```

SQL 通过 **LIKE** 运算符支持字符串模式匹配，使用两种通配符：

- % 代表零个或多个任意字符
- _ 代表恰好一个任意字符
- 可以用 \ 对 % 和 _ 进行转义
  - 例如：name **LIKE** 'Michael_Jordan'

练习

以下哪些名字会被返回?

```
SELECT *
FROM   Students
WHERE  name LIKE 'Sm_t%'
```

1. Smit
2. SMIT
3. Smart
4. Smith
5. Smythe
6. Smut
7. Smeath
8. Smt

练习（答案）

以下哪些名字会被返回?

1. Smit ✓（第 1 个）
2. SMIT —（LIKE 不区分大小写，实际上也匹配；取决于数据库配置）
3. Smart —（第 3 个字母 a 不满足 _t 匹配）
4. Smith ✓（第 4 个）
5. Smythe ✓（第 5 个；Smyth 匹配 Sm_t，% 匹配 e）
6. Smut ✓（第 6 个）
7. Smeath —（Smeat 不满足 Sm_t）
8. Smt —（不足 4 个字符）

答案：1、4、5、6

练习

以下查询会返回所有学生吗?

```
SELECT *
FROM   Students
WHERE  gpa <= 3.5 OR gpa > 3.5
```

name	age	gpa
Mike	20	4.0
Joe	18	NULL
Alice	21	3.8

SQL 中的 NULL（空值）

清華大學 Tsinghua University

当某个字段没有值时，可以用 NULL 表示

NULL 可能代表多种含义：

- 该值不存在
- 该值存在但未知
- 该值不适用
- 等等……

NULL 约束（示例）

```
CREATE TABLE Students (
        name CHAR(20) NOT NULL,
        age CHAR(20) NOT NULL,
        gpa FLOAT
)
```

练习

以下查询会返回所有学生吗?

```
SELECT *
FROM   Students
WHERE  gpa <= 3.5 OR gpa > 3.5 OR gpa is NULL
```

name	age	gpa
Mike	20	4.0
Joe	18	NULL
Alice	21	3.8

课程大纲

清華大學 Tsinghua University

单表查询

- SFW 查询
- 常用运算符：DISTINCT、LIKE、ORDER BY
- NULL 空值

多表查询

- 外键约束
- 连接
- 集合操作

外键约束

外键约束

- student_id 引用 sid

Students

sid	name	gpa
101	Bob	3.2
123	Mary	3.8

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

外键约束

外键约束

- student_id 引用 sid

Students

sid	name	gpa
101	Bob	3.2
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

声明外键

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

```
CREATE TABLE Enrolled(
        student_id CHAR(20),
        cid        CHAR(20),
        grade      CHAR(10),
        PRIMARY KEY (student_id, cid),
        FOREIGN KEY (student_id) REFERENCES Students(sid)
)
```

插入操作

若向 Enrolled 中插入一条记录，但对应的学生不在 Students 表中，会怎样?

- 插入操作将被拒绝

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A
190	354	A

删除操作

若删除一个已选课的学生（该学生在 Enrolled 表中有对应记录），会怎样?

- 方式一：拒绝删除（ON DELETE RESTRICT）

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

ON DELETE RESTRICT

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

```
CREATE TABLE Enrolled(
        student_id   CHAR(20),
        cid                  CHAR(20),
        grade          CHAR(10),
        PRIMARY KEY (student_id, cid),
        FOREIGN KEY (student_id) REFERENCES Students(sid)
        ON DELETE RESTRICT
)
```

效果：若被删除的学生在 Enrolled 中有记录，则删除操作被拒绝。

删除操作

若删除一个已选课的学生，会怎样?

- 方式二：级联删除（ON DELETE CASCADE）

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

ON DELETE CASCADE

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

```
CREATE TABLE Enrolled(
        student_id CHAR(20),
        cid                CHAR(20),
        grade      CHAR(10),
        PRIMARY KEY (student_id, cid),
        FOREIGN KEY (student_id) REFERENCES Students(sid)
        ON DELETE CASCADE
)
```

删除操作

若删除一个已选课的学生，会怎样?

- 方式三：将外键置为 NULL（ON DELETE SET NULL)

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
NULL	354	A

注意：虽然此操作满足外键约束，但由于 student_id 是主键的一部分，主键不能为 NULL，因此该删除操作仍会被拒绝。

ON DELETE SET NULL

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

```
CREATE TABLE Enrolled(
        student_id CHAR(20),
        cid                CHAR(20),
        grade      CHAR(10),
        PRIMARY KEY (student_id, cid),
        FOREIGN KEY (student_id) REFERENCES Students(sid)
        ON DELETE SET NULL
)
```

课程大纲

清華大學 Tsinghua University

单表查询

- SFW 查询
- 常用运算符：DISTINCT、LIKE、ORDER BY
- NULL 空值

多表查询

- 外键约束
- **连接**
- 集合操作

为什么要使用多张表?

Tsinghua University

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A
123	454	A+
156	354	A

VS.

EnrolledStudents

student_id	name	gpa	cid	grade
123	Mary	3.8	354	A
123	Mary	3.8	454	A+
156	Mike	3.7	354	A

多表存储 vs. 单表存储

清華大學 Tsinghua University

多表存储的优点

- 更新数据更简单（例如：将 Mary 的 GPA 改为 3.9）
- 查询单个实体信息更快（例如：查询 Mary 的 GPA）

单表存储的优点

- 数据共享更方便（例如：可以直接将一张表发送给他人）
- 避免多表连接的额外开销（例如：查询 Mary 所修的所有课程）

连接：示例

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

查找在 354 课程中获得 A+ 的学生，返回其姓名和 GPA。

```
SELECT name, gpa
FROM   Students, Enrolled
WHERE  sid = student_id AND
       cid = 354 AND grad = 'A+'
```

如何连接2张表?

哪些行感兴趣?

练习

清華大學 Tsinghua University

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND grade >= 'A'
```

name
Mary

(A)

name
Mary
Mike

(B)

name
Mary
Mike
Mary

(C)

name
Mary
Mary
Mike

(D)

元组变量的必要性

清華大學 Tsinghua University

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	name	grade
123	354	DB I	A+
123	454	DB II	A+
156	354	DB I	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id
```

元组变量

清華大學 Tsinghua University

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	name	grade
123	354	DB I	A+
123	454	DB II	A+
156	354	DB I	A

```
SELECT Students.name
FROM   Students, Enrolled
WHERE  sid = student_id
```

```
SELECT S.name
FROM   Students S, Enrolled
WHERE  sid = student_id
```

连接的其他写法

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND
       cid = 354 AND grad = 'A+'
```

```
SELECT name
FROM   Students
JOIN   Enrolled ON sid = student_id
WHERE cid = 354 AND grad = 'A+'
```

```
SELECT name
FROM   Students
JOIN   Enrolled ON sid = student_id
            AND cid = 354 AND grad = 'A+'
```

连接类型

```
SELECT name, course
FROM   Student INNER JOIN Enroll ON
           name = stdName
```

```
SELECT name, course
FROM   Student FULL OUTER JOIN Enroll ON          name =
stdName
```

```
SELECT name
FROM   Student LEFT OUTER JOIN Enroll ON          name =
stdName
```

```
SELECT name
FROM   Student RIGHT OUTER JOIN Enroll ON         name =
stdName
```

连接类型

```
SELECT name, course
FROM   Student INNEI                ıroll ON
            name = stdName
```

```
SELECT name, course
FROM   Student FULL OUTER .         oll ON          name =
stdName
```

```
SELECT name
FROM   Student LEFT OUTER .         ɔll ON          name =
stdName
```

```
SELECT name
FROM   Student RIGHT OUTER          ·oll ON         name =
stdName
```

左外连接

清華大學 Tsinghua University

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

```
SELECT name, course
FROM   Student JOIN | LEFT JOIN Enroll ON
           name = stdName
```

如何查询所有学生，无论是否选课?

左外连接：查询结果

```
SELECT name, course
FROM   Student LEFT JOIN Enroll ON
           name = stdName
```

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

输出结果

name	course
Mary	354
Tom	354
Tom	454
Jack	NULL

右外连接：查询结果

```
SELECT name, course
FROM   Student RIGHT JOIN Enroll ON
           name = stdName
```

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

输出结果

name	course
Mary	354
Tom	354
Tom	454
NULL	354

全外连接：查询结果

```
SELECT name, course
FROM   Enroll FULL JOIN Student ON
           name = stdName
```

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

输出结果

name	course
Mary	354
Tom	354
Tom	454
Jack	NULL
NULL	354

练习题 1

```
SELECT name, course
FROM   Student LEFT JOIN Enroll ON
            name = stdName AND course = 354
```

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

name	course
Mary	354
Tom	354
Jack	NULL

(A)

name	course
Mary	354
Tom	354

(B)

练习题 2

```
SELECT name, course
FROM   Student LEFT JOIN Enroll ON
              name = stdName
WHERE  course = 354
```

Student

name	gpa
Mary	3.8
Tom	3.6
Jack	3.7

Enroll

stdName	course
Mary	354
Tom	354
Tom	454
Bob	354

name	course
Mary	354
Tom	354
Jack	NULL

(A)

name	course
Mary	354
Tom	354

(B)

课程大纲

单表查询

- SFW 查询
- 常用运算符：DISTINCT、LIKE、ORDER BY
- NULL 空值

多表查询

- 外键约束
- 连接
- **集合操作**

集合操作

清華大學 Tsinghua University

SQL 支持三种集合操作：

- UNION（并集）
- INTERSECT（交集）
- EXCEPT（差集）

这些操作要求参与运算的两个Table**具有相容的列结构**（列数相同、类型兼容）。

注意事项：

- 虽然 SQL 标准支持以上三种操作，各数据库系统的实现可能有所差异
- EXCEPT 在部分系统中称为 MINUS

查询修过某门课程的学生

查找修过 354 或 454 课程的学生（用 OR 实现）：

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND (cid = 354 OR cid = 454)
```

查询修过某门课程的学生 — 使用 UNION

使用 UNION 查找修过 354 或 454 课程的学生:

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 354
UNION
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 454
```

查询同时修了两门课的学生

查找同时修了 354 和 454 课程的学生:

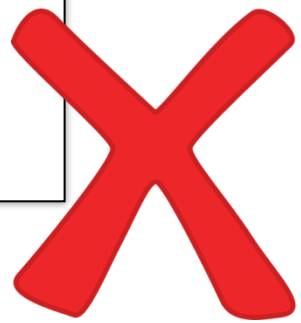
Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students S, Enrolled E
WHERE  sid = student_id AND
     (E.cid = 354 AND E.cid = 454)
```



查询同时修了两门课的学生

正确做法：对 Enrolled 表进行自连接（使用两个别名 E1、E2）：

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students S, Enrolled E1, Enrolled E2
WHERE  S.sid = E1.student_id AND S.sid = E2.student_id
       AND (E1.cid = 354 AND E2.cid = 454)
```

两门都修 — 使用 INTERSECT

同时修了 354 和 454 课程的学生:

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 354
INTERSECT
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 454
```

只修其中一门（差集）

查找修了 354 课程但未修 454 课程的学生：

Students

sid	name	gpa
123	Mary	3.8
156	Mike	3.7

Enrolled

student_id	cid	grade
123	354	A+
123	454	A+
156	354	A

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 354
EXCEPT
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 454
```

EXCEPT 返回第一个查询有而第二个查询没有的记录（差集）

在部分数据库系统中，EXCEPT 写作 MINUS

集合操作与重复值

与普通 SQL 查询不同，UNION、INTERSECT 和 EXCEPT 默认消除重复值。

使用 ALL 关键字保留重复值（多重集操作）：

```
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 354
INTERSECT ALL
SELECT name
FROM   Students, Enrolled
WHERE  sid = student_id AND cid = 454
```

三种多重集操作：UNION ALL、INTERSECT ALL、EXCEPT ALL