为了更完整地支持特教机构的需求，可以在现有表结构的基础上进一步扩展和改进。以下是针对学员信息和其他关联信息的详细改进方案：

---

### 1. \*\*学员信息表 (`students`)\*\*

在原有基础上，增加家长信息、残疾信息、成长档案等字段。

```sql

CREATE TABLE students (

student\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

gender ENUM('男', '女', '其他') NOT NULL,

birthdate DATE NOT NULL, -- 生日

age INT, -- 年龄（可计算，无需手动输入）

id\_card VARCHAR(20), -- 身份证号

disability\_card VARCHAR(50), -- 残疾证号

disability\_type VARCHAR(100), -- 残疾类型

enrollment\_date DATE NOT NULL, -- 入学时间

status ENUM('在读', '毕业', '退学') NOT NULL,

parent\_name VARCHAR(100), -- 家长姓名

parent\_relationship VARCHAR(50), -- 家长关系（如父亲、母亲、监护人）

parent\_phone VARCHAR(20), -- 家长电话

sales\_consultant VARCHAR(100), -- 销售顾问

recommender VARCHAR(100), -- 推荐人

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

---

### 2. \*\*课程信息表 (`courses`)\*\*

存储课程的基本信息。

```sql

CREATE TABLE courses (

course\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_name VARCHAR(100) NOT NULL,

description TEXT,

total\_sessions INT, -- 总课时数

price DECIMAL(10, 2), -- 课程价格

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

---

### 3. \*\*学员与课程关联表 (`student\_courses`)\*\*

记录学员报读的课程信息，包括剩余课程、交费金额、剩余金额等。

```sql

CREATE TABLE student\_courses (

student\_course\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

enrolled\_date DATE NOT NULL, -- 报读日期

total\_sessions INT NOT NULL, -- 总课时数

remaining\_sessions INT NOT NULL, -- 剩余课时数

total\_amount DECIMAL(10, 2) NOT NULL, -- 总交费金额

remaining\_amount DECIMAL(10, 2) NOT NULL, -- 剩余金额

sales\_consultant VARCHAR(100), -- 销售顾问

recommender VARCHAR(100), -- 推荐人

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

```

---

### 4. \*\*交费订单表 (`payment\_orders`)\*\*

记录学员的缴费订单信息。

```sql

CREATE TABLE payment\_orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

amount DECIMAL(10, 2) NOT NULL, -- 交费金额

payment\_date DATE NOT NULL, -- 交费日期

payment\_method ENUM('现金', '银行卡', '支付宝', '微信') NOT NULL, -- 支付方式

order\_status ENUM('已支付', '未支付', '已退款') NOT NULL, -- 订单状态

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (course\_id) REFERENCES courses(course\_id)

);

```

---

### 5. \*\*上课记录表 (`class\_records`)\*\*

记录学员的上课情况。

```sql

CREATE TABLE class\_records (

record\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

course\_id INT NOT NULL,

schedule\_id INT NOT NULL, -- 关联课程安排表

attendance\_status ENUM('出席', '缺席', '请假') NOT NULL, -- 出勤状态

notes TEXT, -- 备注

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (course\_id) REFERENCES courses(course\_id),

FOREIGN KEY (schedule\_id) REFERENCES course\_schedule(schedule\_id)

);

```

---

### 6. \*\*课程安排表 (`course\_schedule`)\*\*

存储课程的具体安排信息。

```sql

CREATE TABLE course\_schedule (

schedule\_id INT AUTO\_INCREMENT PRIMARY KEY,

course\_id INT NOT NULL,

teacher\_id INT NOT NULL,

start\_time DATETIME NOT NULL, -- 课程开始时间

end\_time DATETIME NOT NULL, -- 课程结束时间

location VARCHAR(100), -- 上课地点

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (course\_id) REFERENCES courses(course\_id),

FOREIGN KEY (teacher\_id) REFERENCES teachers(teacher\_id)

);

```

---

### 7. \*\*成长档案表 (`student\_profiles`)\*\*

存储学员的成长档案信息。

```sql

CREATE TABLE student\_profiles (

profile\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

profile\_date DATE NOT NULL, -- 档案记录日期

content TEXT, -- 档案内容

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id)

);

```

---

### 8. \*\*残疾信息表 (`disability\_info`)\*\*

存储学员的残疾信息。

```sql

CREATE TABLE disability\_info (

disability\_id INT AUTO\_INCREMENT PRIMARY KEY,

student\_id INT NOT NULL,

disability\_type VARCHAR(100) NOT NULL, -- 残疾类型

disability\_level VARCHAR(50), -- 残疾等级

description TEXT, -- 描述

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (student\_id) REFERENCES students(student\_id)

);

```

---

### 9. \*\*销售顾问表 (`sales\_consultants`)\*\*

存储销售顾问信息。

```sql

CREATE TABLE sales\_consultants (

consult\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone VARCHAR(20),

email VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

---

### 10. \*\*推荐人表 (`recommenders`)\*\*

存储推荐人信息。

```sql

CREATE TABLE recommenders (

recommender\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone VARCHAR(20),

email VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

```

---

### 11. \*\*通知表 (`notifications`)\*\*

存储系统通知或提醒信息。

```sql

CREATE TABLE notifications (

notification\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL, -- 接收通知的用户

message TEXT NOT NULL, -- 通知内容

is\_read ENUM('已读', '未读') DEFAULT '未读',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

```

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(100) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

role ENUM('管理员', '老师', '财务', '教务') NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP);

---

### 总结

通过以上改进，数据库可以更全面地支持特教机构的需求，包括：

1. \*\*学员信息\*\*：增加家长信息、残疾信息、成长档案等。

2. \*\*课程信息\*\*：记录课程的基本信息和价格。

3. \*\*学员与课程关联\*\*：记录报读课程、剩余课程、交费金额、剩余金额等。

4. \*\*交费订单\*\*：记录学员的缴费订单。

5. \*\*上课记录\*\*：记录学员的出勤情况。

6. \*\*成长档案\*\*：记录学员的成长信息。

7. \*\*销售顾问和推荐人\*\*：记录销售顾问和推荐人信息。

这些表结构可以根据实际需求进一步调整和优化，确保系统能够高效地支持特教机构的日常运营和管理。

1. 创建新数据库

如果你想创建一个新的数据库，可以运行以下命令：

sql

复制

CREATE DATABASE mydatabase;

然后再次运行 SHOW DATABASES;，你会看到新创建的数据库。

2. 使用数据库

运行以下命令选择要使用的数据库（例如 mydatabase）：

sql

复制

USE mydatabase;

3. 创建表

在数据库中创建一个新表（例如 mytable）：

sql

复制

CREATE TABLE mytable (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL

);

4. 插入数据

向表中插入一些数据：

sql

复制

INSERT INTO mytable (name, email) VALUES ('Alice', 'alice@example.com');

INSERT INTO mytable (name, email) VALUES ('Bob', 'bob@example.com');

5. 查询数据

查询表中的数据：

sql

复制

SELECT \* FROM mytable;

6. 退出 MySQL

运行以下命令退出 MySQL 命令行：

sql

复制

EXIT;

与 PHP 连接

如果你需要将 PHP 与 MySQL 连接，可以使用以下代码：

php

复制

<?php

$host = 'localhost'; // MySQL 服务器地址

$user = 'root'; // MySQL 用户名

$password = '你的密码'; // MySQL 密码

$database = 'mydatabase'; // 数据库名称

// 创建连接

$conn = new mysqli($host, $user, $password, $database);

// 检查连接

if ($conn->connect\_error) {

die("连接失败: " . $conn->connect\_error);

}

echo "连接成功";

// 查询数据

$sql = "SELECT \* FROM mytable";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

// 输出数据

while ($row = $result->fetch\_assoc()) {

echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " . $row["email"] . "<br>";

}

} else {

echo "0 结果";

}

// 关闭连接

$conn->close();

?>

总结

你已经成功连接到 MySQL 服务器，并查看了默认的数据库。

可以创建新的数据库和表，插入和查询数据。

如果需要将 PHP 与 MySQL 连接，可以参考上面的 PHP 代码。