**视图**

**视图一**

CREATE VIEW v\_导师学科学院信息 AS

SELECT DISTINCT

d.导师ID,

d.职称,

a.学科ID,

d.学院ID,

a.学科名称 AS 学科名称,

e.学院名称 AS 学院名称

FROM dbo.导师 d

JOIN dbo.从属 s ON d.导师ID = s.导师ID

JOIN dbo.学科 a ON s.学科ID = a.学科ID

JOIN dbo.属于 c ON d.学院ID = c.学院ID

JOIN dbo.学院 e ON c.学院ID = e.学院ID;



适用范围：

导师与学科信息管理：该视图将导师、学科和学院之间的关系抽象化，简化了查询过程。对于需要查询导师信息及其所对应的学科和学院名称的场景（如导师管理、学科分配、学科所属学院等），该视图非常适用。

数据整合与报表：在教育管理系统中，这样的视图有助于生成导师相关的报表（如导师与学科的分布、导师的学科专长、学院内导师的分布等）。视图通过提供单一的数据来源来简化多表查询。

优化查询性能：通过将复杂的连接操作封装到视图中，可以减少客户端查询的复杂度，从而优化用户查询性能，特别是在需要频繁进行相似查询时，减少了重复的查询代码。

权限控制：在用户只需访问部分字段或仅查看特定关联数据时，视图可以提供一个简化的数据接口，从而控制对敏感字段（如导师详细信息、学院信息等）的访问。

适用的管理场景：

教育信息管理系统中的导师分配。

数据汇总与报表生成（如学科与导师的匹配情况）。

面向教育工作者、学科负责人、学院管理人员的查询和分析。

**视图二**

CREATE VIEW v\_招生目录与条目信息 AS

SELECT

a.招生目录ID,

a.是否通过学校审核,

b.招生目录条目ID,

b.学院ID

FROM

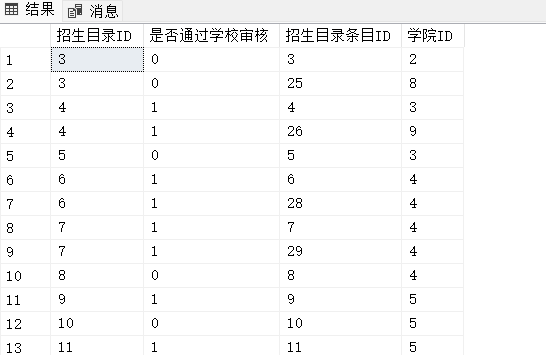
dbo.招生目录 a

JOIN

dbo.包含3 c ON a.招生目录ID = c.招生目录ID

JOIN

dbo.招生目录条目 b ON c.招生目录条目ID = b.招生目录条目ID;

  
适用范围：

招生信息查询与管理：该视图将招生目录、条目和学院之间的关系整理起来，对于招生管理系统中需要快速查询招生条目及其相关学院信息的场景非常有用。通过这个视图，管理人员可以快速查找某个招生目录下的所有条目和对应的学院信息。

数据整合与报表：这对于招生管理部门生成招生计划、招生审核报告以及招生统计数据等非常有帮助。视图简化了从多个表中提取数据的过程，使得数据处理更加高效。

审核与权限控制：在涉及审核流程时，可以使用该视图来生成包含审核状态的招生目录数据，便于审核人员进行查看和处理。同时，视图可以用来简化用户权限管理，限制对复杂查询的访问，只提供所需的数据。

系统性能优化：使用视图可以减少开发过程中多表连接查询的复杂度，提高查询效率，尤其是在涉及到大量数据表连接的场景下。视图的引入有助于优化查询的执行计划，从而提高整体的系统响应速度。

适用的管理场景：

招生信息管理系统中的招生目录和条目的管理。

招生计划生成、招生审核报表和招生分析。

各学院的招生统计与分析（如学院招生情况的查询）。

**试图三**

CREATE VIEW v\_考生与总成绩信息 AS

SELECT

c.考生ID,

c.考生类别,

c.性别,

c.毕业时间,

c.出生日期,

t.总成绩ID,

t.类别 AS 总成绩类别,

t.总分数值 AS 总分

FROM

dbo.考生 c

JOIN

dbo.拥有 h ON c.考生ID = h.考生ID

JOIN

dbo.总成绩 t ON h.总成绩ID = t.总成绩ID;



**适用范围**：主要用于需要显示考生及其总成绩的查询，比如生成成绩单或考生成绩概况的报表。

**试图四**

CREATE VIEW v\_成绩考试科目与考生信息 AS

SELECT

c.考生ID,

g.考试科目ID,

g.类别 AS 科目类别,

g.名称 AS 科目名称,

g.考试时间,

g.考试地点,

s.数值 AS 科目成绩

FROM

dbo.成绩 s

JOIN

dbo.考试科目 g ON s.考试科目ID = g.考试科目ID

JOIN

dbo.考生 c ON s.考生ID = c.考生ID;



**适用范围**：适用于生成包含考试科目与成绩的报表，帮助分析考生在各科目上的表现。

**试图五**

CREATE VIEW v\_考生复试志愿与复试结果 AS

SELECT DISTINCT

c.考生ID,

r.志愿ID,

r.学科ID,

rs.综合评价,

rs.录取状态,

q.导师ID

FROM

dbo.考生 c

JOIN

dbo.填报 t ON c.考生ID = t.考生ID

JOIN

dbo.复试志愿 r ON t.考生ID = r.考生ID

JOIN

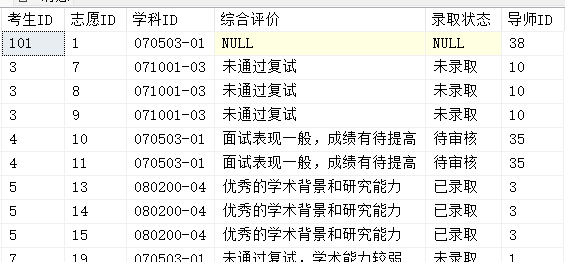
dbo.复试结果 rs ON c.考生ID = rs.考生ID

JOIN

dbo.确定 q ON rs.复试结果ID = q.复试结果ID

JOIN

dbo.导师 d ON q.导师ID = q.导师ID;



这个视图通过连接 考生表、填报表、复试志愿表、复试结果表 和 导师表，来展示考生的复试志愿和复试结果信息。

适用范围：适用于需要展示考生复试志愿、复试结果以及分配导师等信息的场景，特别是在录取和复试流程中。

**索引**

索引1：IX\_导师\_学院ID

目的：优化基于 学院ID 的查询性能，尤其是涉及到学院相关信息的查询。

索引：

CREATE NONCLUSTERED INDEX IX\_导师\_学院ID

ON dbo.导师 (学院ID);

IX\_招生目录条目\_学科ID

目的：优化基于 学科ID 查询招生目录条目的性能，尤其是在查看特定学科的招生信息时。

CREATE NONCLUSTERED INDEX IX\_招生目录条目\_学科ID

ON dbo.招生目录条目 (学科ID);

-- 在 `从属` 表上创建复合索引

目的：优化查询导师与学科的关联信息。

CREATE NONCLUSTERED INDEX IX\_从属\_导师ID\_学科ID

ON dbo.从属 (导师ID, 学科ID);

-- 创建复合索引，提高查询性能（如果查询经常基于 `学科ID` 和 `导师ID` 进行过滤）

CREATE NONCLUSTERED INDEX IX\_从属\_学科ID\_导师ID

ON dbo.从属 (学科ID, 导师ID);

-- 创建复合索引，提高基于 `招生目录ID` 和 `招生目录条目ID` 的查询性能

CREATE NONCLUSTERED INDEX IX\_包含3\_招生目录ID\_招生目录条目ID

ON dbo.包含3 (招生目录ID, 招生目录条目ID);

--查询经常根据 考生ID 进行筛选或连接操作，因此可以为 考生ID 创建索引：

CREATE NONCLUSTERED INDEX IX\_考生\_考生ID

ON dbo.考生 (考生ID);

--为了优化查询基于 考生ID 的连接，也可以为 考生ID 字段创建索引：

CREATE NONCLUSTERED INDEX IX\_复试结果\_考生ID

ON dbo.复试结果 (考生ID);