

实验三 视图管理和使用

XX 班 100002000001 许愿

【实验目的】

了解如何创建视图及如何在视图中进行数据查询。

【实验环境，主要仪器及软件】

Windows 11, SQL Server 2022 + SQL Server Management Studio 20 / Navicat

【实验内容】

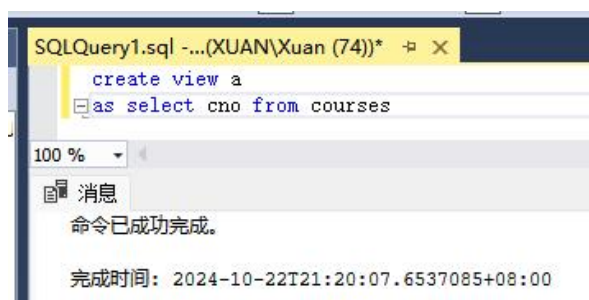
创建视图，使用 SELECT 语句结合视图完成所要求的查询。

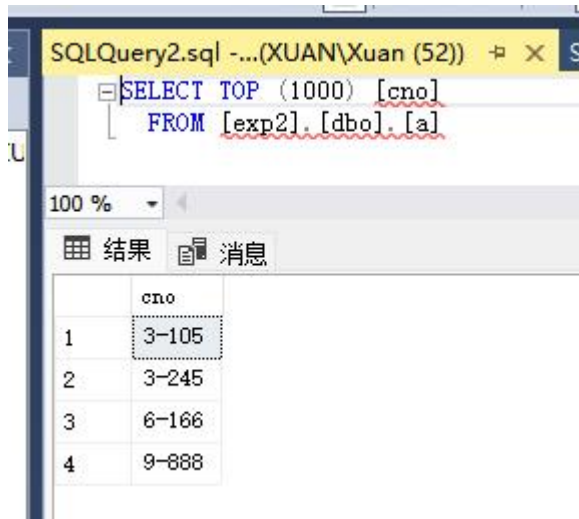
【实验步骤】

一、 创建视图

1. 使用命令创建视图

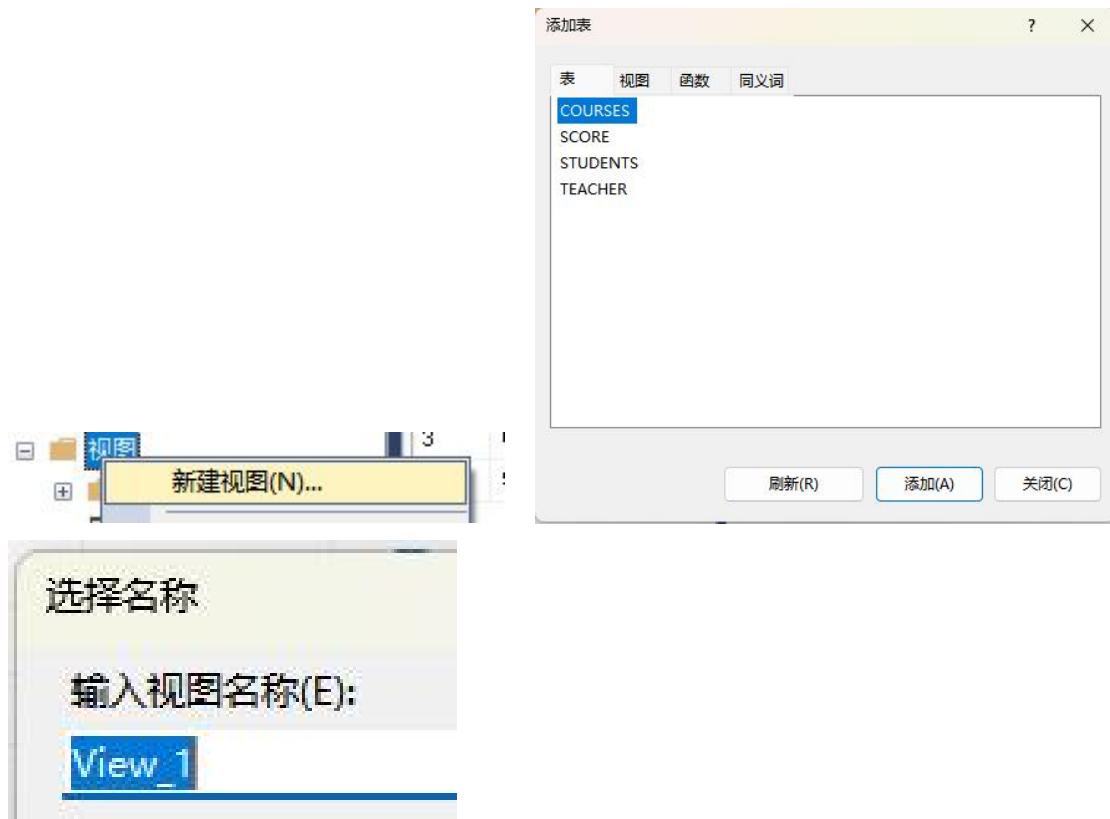
在查询分析器窗口输入 CREATE VIEW 命令，如图所示为创建一个查询所有课程的 CNO 的视图，命名为 a。

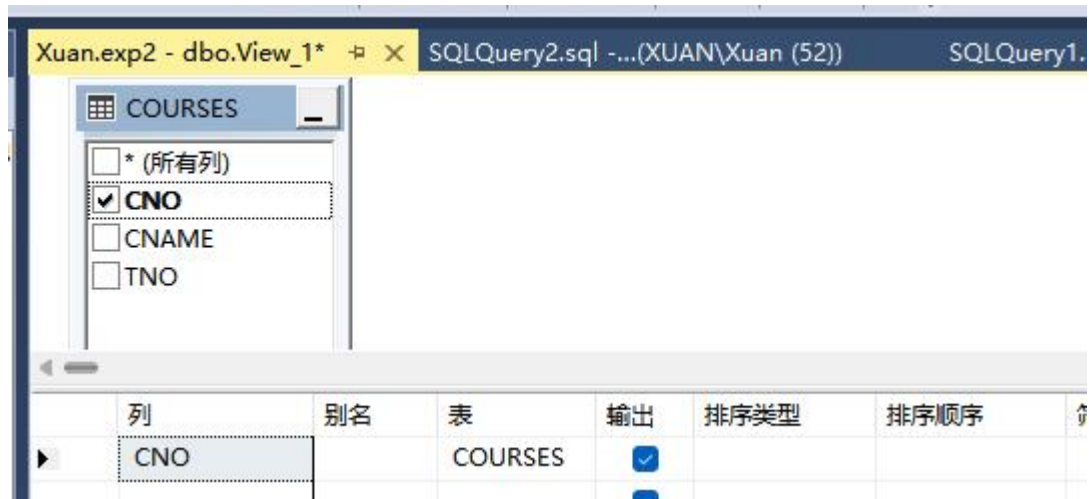




2. 使用菜单创建视图

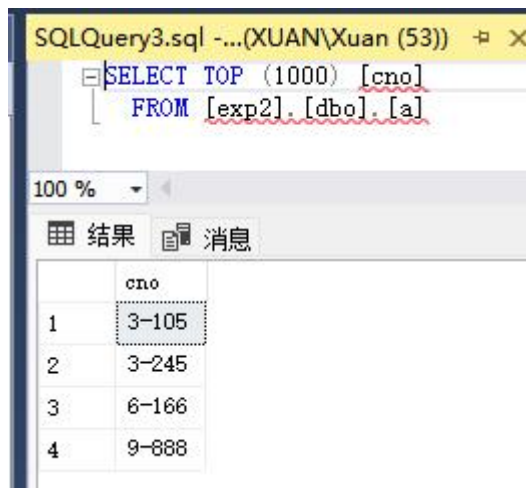
打开 SQL Server Management Studio 窗口，在资源管理器中单击右键视图按右键，选择需要包含的 SELECT 字段，然后右键保存。如图所示为创建一个查询所有课程的 CNO 的视图，命名为 View_1。





二、 查询视图

进入查询分析器窗口，右键选择前 1000 行，即可进行视图查询。



三、 实验内容

1. 设有学生表 S(SNO,SN) (SNO 为学生号, SN 为姓名) 和学生选修课程表 SC(SNO,CNO,CN,CNG) (SNO 为学生号, CNO 为课程号, CN 为课程名, G 为成绩), 试用 SQL 语言完成以下各题: (以下为本题所使用的表格)

对象 SC @exp2.dbo (本地SQL) - 表							
保存 添加字段 删除字段 主键							
字段	索引	外键	唯一键	检查	触发器	选项	存储
名	类型	大小	比例	不是 null	键	注释	
▶ SNO	char	10		<input checked="" type="checkbox"/>	1		
CNO	char	10		<input checked="" type="checkbox"/>	2		
CN	char	20		<input type="checkbox"/>			
G	real			<input type="checkbox"/>			

对象 S @exp2.dbo (本地SQL) - 表							
保存 添加字段 删除字段 主键							
字段	索引	外键	唯一键	检查	触发器	选项	存储
名	类型	大小	比例	不是 null	键	注释	
▶ SNO	char	10		<input checked="" type="checkbox"/>	1		
SN	char	10		<input type="checkbox"/>			

对象 SC @exp2.dbo (本地SQL) - 表				对象 S @exp2.dbo (本地SQL) - 表	
开始事务 文本 筛选 排序 列				开始事务 文本 筛选	
SNO	CNO	CN	G	SNO	SN
101	111	高等数学	93	101	王小明
101	112	物理	90	102	郑浩
102	111	高等数学	88	103	李云
103	114	低等数学	22	104	刘平
▶ 104	112	化学	14	▶ 105	陈加

(1) 建立一个视图 VSSC(SNO,SN,CNO,CN,G)

```
SQLQuery4.sql -...(XUAN\Xuan (54))* SQLQuery1.sql -...(XUAN\Xuan (74))*
create view VSSC
as select S.SNO, SN, CNO, CN, G from S, SC where S.SNO=SC.SNO
```

100 %

消息

命令已成功完成。

完成时间: 2024-10-22T21:43:02.8687460+08:00

SQLQuery4.sql -...(XUAN\Xuan (54))*

```
SELECT *
FROM VSSC
```

100 %

结果 消息

	SNO	SN	CNO	CN	G
1	101	王小明	111	高等数学	93
2	101	王小明	112	物理	90
3	102	郑浩	111	高等数学	88
4	103	李云	114	低等数学	22
5	104	刘平	112	化学	14

(2) 从视图 VSSC 上查询平均成绩在 90 分以上的 SN,CNO,CN 和 G

课程的平均成绩在 90 分以上

SQLQuery7.sql -...(XUAN\Xuan (68))*

```
select SN,CNO,CN,G
from VSSC
where CNO in (
select CNO
from VSSC
group by CNO
having avg(G)>90)
```

100 %

结果 消息

	SN	CNO	CN	G
1	王小明	111	高等数学	93
2	郑浩	111	高等数学	88

个人的平均成绩在 90 分及以上

```
SQLQuery7.sql | -...(XUAN\Xuan (68))*
select SN, CNO, CN, G
from VSSC
where SNO in (
select SNO
from VSSC
group by SNO
having avg(G) > 90)
```

100 %

结果 消息

	SN	CNO	CN	G
1	王小明	111	高等数学	93
2	王小明	112	物理	90

2. 已知三个关系 R、S、T 如图

R

A	B	C
a1	b1	20
a1	b2	22
a2	b1	18
a2	b3	19

S

A	D	E
a1	d1	15
a2	d2	18
a1	d2	24

T

D	F
d2	f2
d3	f3

对象 R @ep.dbo (本地SQL) - 表

开始事务 文本 筛选 排序 列

A	B	C
a1	b1	20
a1	b2	22
a2	b1	18
a2	b3	19

对象 100% -- S @ep.dbo (本地SQL) - 表		
开始事务	文本	筛选
排序		
A	D	E
a1	d1	15
a2	d2	18
a1	d2	24

对象 T @ep.dbo (本地SQL) - 表	
开始事务	文本
筛选	排序
列	
D	F
d2	f2
d3	f3

试用 SQL 语句实现如下操作：

- (1) 将 R、S、T 三个关系按关联属性建立一个视图 RST；

```
SQLQuery5.sql -...(XUAN\Xuan (52))*
create view RST
as
select R.A, B, C, S.D, E, F
from R, S, T
where R.A=S.A and S.D=T.D
```

100 %

消息

命令已成功完成。

完成时间：2024-10-22T21:58:58.6455546+08:00

SQLQuery6.sql -...(XUAN\Xuan (65))*

```
select *
from RST
```

100 %

结果 消息

	A	B	C	D	E	F
1	a1	b1	20	d2	24	f2
2	a1	b2	22	d2	24	f2
3	a2	b1	18	d2	18	f2
4	a2	b3	19	d2	18	f2

(2) 对视图 RST 按属性 A 分组后，求属性 C 和 E 的平均值。

SQLQuery6.sql -...(XUAN\Xuan (65))* SQLQuery5.s

```
select A, AVG(C) as c_avg, AVG(E) as e_avg
from RST
group by A
```

100 %

结果 消息

	A	c_avg	e_avg
1	a1	21	24
2	a2	18.5	18

3. 已知关系 R 如图

A	B	C
97	b1	84
97	b2	92
97	b3	98
98	b1	72
98	b2	84
98	b3	95
99	b1	88
99	b2	94

对象 R @ep2.dbo (本地SQL) - 表		
A	B	C
97	b1	84
97	b2	92
97	b3	98
98	b1	72
98	b2	84
98	b3	95
99	b1	88
99	b2	94

试用 SQL 语句实现下列操作：

- (1) 按属性 A 分组，求出每组中在属性 C 上的最大值和最小值，且将它们置于视图 RVE 中；

```
SQLQuery8.sql -...(XUAN\Xuan (68))* X SQLQuery10.sql
create view RVE
as
select A,max(C) as max_c,min(C) as min_c
from R
group by A
```

消息
命令已成功完成。
完成时间：2024-10-22T22:10:20.2108573+08:00

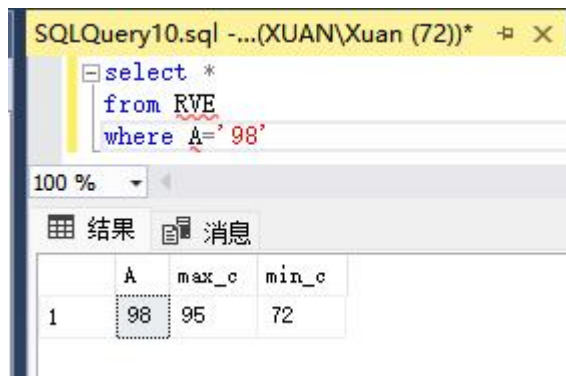
SQLQuery10.sql -...(XUAN\Xuan (72))* S

```
select *
from RVE
```

结果 消息

	A	max_c	min_c
1	97	98	84
2	98	95	72
3	99	94	88

(2) 在视图 RVE 中查询属性 A="98"的记录。



The screenshot shows a SQL query window titled 'SQLQuery10.sql -...(XUAN\Xuan (72))*'. The query is:
`select *
from RVE
where A='98'`
Below the query, the 'Results' tab is active, displaying a table with 4 columns: A, max_c, and min_c. The first row has the value 1 in the first column, 98 in the second, 95 in the third, and 72 in the fourth.

	A	max_c	min_c
1	98	95	72

4、假定一个工厂由不同的车间构成，每个员工的工资的计算方法如下

对每个员工 A，A 的月工资=A 生产的部件的价值×0.1% + A 的月基本工资 + A 所在车间的月平均奖金。假定一个员工只能在一个车间工作。

试设计该工厂的工资信息系统的 ER 图，并导出对应的关系模式。



The screenshot shows the 'Design' view of a table named 'WORKER' in the 'ep3.dbo' database. The table has the following fields: NO (int, primary key), NAME (char, 10), NORMAL_SALARY (real), DEPARTMENT (char, 10), BORN_VALUE (real), AWARD_VALUE (real), and TOTAL_SALARY (real). The 'NO' field is marked as the primary key with a yellow key icon and the number 1.

名	类型	大小	比例	不是 null	键
NO	int			<input checked="" type="checkbox"/>	1
NAME	char	10		<input type="checkbox"/>	
NORMAL_SALARY	real			<input type="checkbox"/>	
DEPARTMENT	char	10		<input type="checkbox"/>	
BORN_VALUE	real			<input type="checkbox"/>	
AWARD_VALUE	real			<input type="checkbox"/>	
TOTAL_SALARY	real			<input type="checkbox"/>	

此处创建一张表格 WORKER，NO 指员工的工号，NAME 指员工的名字，NORMAL_VALUE 指员工的月基本工资，DEPARTMENT 指员工所在的部门，BORN_VALUE 指员工所生产部件的价值，AWARD_VALUE 指员工的月平均奖金。总工资的字在后续视图中添加。

对象

WORKER @ep3.dbo (本地SQL) - 表

WORKER @ep3.dbo (本地SQL) - 表

开始事务

文本

筛选

排序

列

导入

导出

数据生成

创建图表

NO	NAME	NORMAL_SALARY	DEPARTMENT	BORN_VALUE	AWARD_VALUE	TOTAL_SALARY
1	小王	2000	A	40000	2000	(Null)
2	小刘	2000	A	35000	3000	(Null)
3	小张	2000	A	24000	6000	(Null)
4	小郑	3000	B	24000	8000	(Null)
5	小赵	4000	C	24000	2000	(Null)
6	小钱	6000	C	35000	2000	(Null)
7	小孙	4000	C	30000	3200	(Null)
8	小李	3000	B	30000	7600	(Null)
9	小林	2500	B	30000	5701	(Null)
10	小博	7000	B	40000	36726	(Null)

在此关系模式的基础上，利用 SQL 命令，进行以下计算：

- (1) 求工厂的基本工资总和；

使用 SUM 函数合并 WORKER 表中所有 NORMAL_SALARY 即可。

SQLQuery11.sql -...(XUAN\Xuan (53))*	
<pre>select sum(NORMAL_SALARY) as total_normal_salary from WORKER</pre>	
100 %	
结果	消息
total_normal_salary	
1	35500

- (2) 求每个员工所在车间的月平均奖金；

使用 AVG 函数计算每个车间的月平均奖金,由此创建出视图 DEP_AVG_AWARD。

SQLQuery11.sql -...(XUAN\Xuan (53))*	
<pre>create view DEP_AVG_AWARD as select DEPARTMENT, AVG(AWARD_VALUE) as AVG_AWARD_VALUE from WORKER group by DEPARTMENT</pre>	
100 %	
消息	
命令已成功完成。	
完成时间: 2024-10-22T22:39:56.9040880+08:00	

SQLQuery12.sql -...(XUAN\Xuan (54))*

```
select *
from DEP_AVG_AWARD
```

100 %

结果 消息

	DEPARTMENT	AVG_AWARD_VALUE
1	A	3666.66666666667
2	B	14506.75
3	C	2400

生成一份工资表 SALARY_TABLE 的视图，将车间月平均奖金左连接至员工表 WORKER 中。

SQLQuery11.sql -...(XUAN\Xuan (53))* SQLQuery12.sql -...(XUAN\Xuan (54))*

```
create view SALARY_TABLE
as
select NO, NAME, NORMAL_SALARY, W.DEPARTMENT, BORN_VALUE, AVG_AWARD_VALUE, TOTAL_SALARY
from WORKER W
left join DEP_AVG_AWARD D
on W.DEPARTMENT=D.DEPARTMENT
```

100 %

消息

命令已成功完成。

完成时间: 2024-10-22T22:50:17.0204027+08:00

SQLQuery13.sql -...(XUAN\Xuan (67))* SQLQuery11.sql -...(XUAN\Xuan (53))*

```
select *
from SALARY_TABLE
```

100 %

结果 消息

	NO	NAME	NORMAL_SALARY	DEPARTMENT	BORN_VALUE	AVG_AWARD_VALUE	TOTAL_SALARY
1	1	小王	2000	A	40000	3666.66666666667	NULL
2	2	小刘	2000	A	35000	3666.66666666667	NULL
3	3	小张	2000	A	24000	3666.66666666667	NULL
4	4	小郑	3000	B	24000	14506.75	NULL
5	5	小赵	4000	C	24000	2400	NULL
6	6	小钱	6000	C	35000	2400	NULL
7	7	小孙	4000	C	30000	2400	NULL
8	8	小李	3000	B	30000	14506.75	NULL
9	9	小林	2500	B	30000	14506.75	NULL
10	10	小傅	7000	B	40000	14506.75	NULL

(3) 求所有车间每个员工的月工资。

```
SQLQuery12.sql | -...(XUAN\Xuan (54))*  SQLQuery13.sql | -...(XUAN\Xuan (67))*  SQL
update SALARY_TABLE
set TOTAL_SALARY = NORMAL_SALARY + BORN_VALUE * 0.001 + AVG_AWARD_VALUE
```

100 %

消息

(10 行受影响)

完成时间: 2024-10-22T22:50:55.7918995+08:00

```
SQLQuery12.sql | -...(XUAN\Xuan (54))*  SQLQuery13.sql | -...(XUAN\Xuan (67))*  SQL
select *
from SALARY_TABLE
```

100 %

结果 消息

	NO	NAME	NORMAL_SALARY	DEPARTMENT	BORN_VALUE	AVG_AWARD_VALUE	TOTAL_SALARY
1	1	小王	2000	A	40000	3666.66666666667	5706.667
2	2	小刘	2000	A	35000	3666.66666666667	5701.667
3	3	小张	2000	A	24000	3666.66666666667	5690.667
4	4	小郑	3000	B	24000	14506.75	17530.75
5	5	小赵	4000	C	24000	2400	6424
6	6	小钱	6000	C	35000	2400	8435
7	7	小孙	4000	C	30000	2400	6430
8	8	小李	3000	B	30000	14506.75	17536.75
9	9	小林	2500	B	30000	14506.75	17036.75
10	10	小傅	7000	B	40000	14506.75	21546.75

若需要将总工资精确到分，可以使用 ROUND()函数进行四舍五入。