USE master

IF EXISTS(SELECT NAME FROM SYSDATABASES WHERE name='Demo')

DROP DATABASE Demo

GO

CREATE DATABASE Demo

ON PRIMARY

(

NAME='Demo\_data',

FILENAME='G:\DB\try\Demo\_data.mdf',

SIZE=10MB,

FILEGROWTH = 5MB

)

LOG ON

(

NAME='Demo\_log.ldf',

FILENAME='G:\DB\try\Demo\_log.ldf',

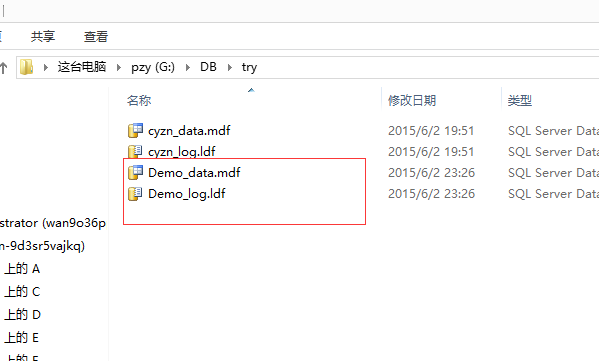
SIZE=5MB,

MAXSIZE=1000MB,

FILEGROWTH=10%

)

GO



USE Demo

IF EXISTS(SELECT NAME FROM SYSOBJECTS WHERE NAME='Department')

DROP TABLE Department

GO

CREATE TABLE Department

(

dp\_id INT IDENTITY(1,1) PRIMARY KEY, --IDENTITY(1,1)自动增长

dp\_name VARCHAR(10) NOT NULL,

--CONSTRAINT Unique\_Name UNIQUE NONCLUSTERED(dp\_name),

--加上唯一约束后会按照该字段排序

)

INSERT INTO Department(dp\_name) VALUES ('地科系')

INSERT INTO Department(dp\_name) VALUES ('工程系')

INSERT INTO Department(dp\_name) VALUES ('仪器系')

INSERT INTO Department(dp\_name) VALUES ('信息系')

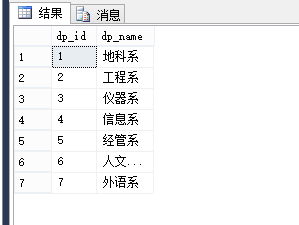
INSERT INTO Department(dp\_name) VALUES ('经管系')

INSERT INTO Department(dp\_name) VALUES ('人文社科系')

INSERT INTO Department(dp\_name) VALUES ('外语系')

SELECT \* FROM Department

GO



USE Demo

IF EXISTS(SELECT NAME FROM SYSOBJECTS WHERE NAME='Teainfo')

DROP TABLE Teainfo

GO

CREATE TABLE Teainfo

(

tea\_id NCHAR(18) PRIMARY KEY,

tea\_name VARCHAR(6) NOT NULL,

tea\_tel NVARCHAR(11) NULL,

tea\_sex CHAR(2) CONSTRAINT tea\_sex CHECK (tea\_sex in ('男','女')),

tea\_age INT NULL,

--CREATE TABLE 时的 SQL FOREIGN KEY 约束:

dp\_id INT FOREIGN KEY REFERENCES Department(dp\_id)

--定义多个列的 FOREIGN KEY 约束

--CONSTRAINT fk\_PerOrders FOREIGN KEY (P\_Id) REFERENCES Persons(P\_Id)

--ALTER TABLE 时的 SQL FOREIGN KEY 约束:

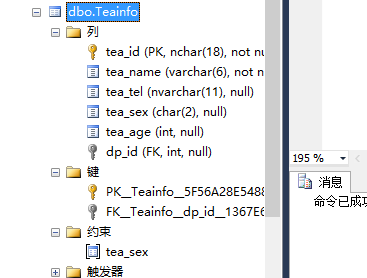
--ALTER TABLE Orders ADD FOREIGN KEY (P\_Id) REFERENCES Persons(P\_Id)

--定义多个列的 FOREIGN KEY 约束

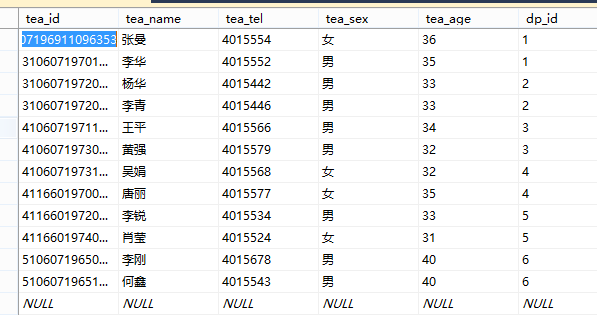
--ALTER TABLE Orders ADD CONSTRAINT fk\_PerOrders FOREIGN KEY (P\_Id) REFERENCES Persons(P\_Id)

)

GO



INSERT INTO Demo.dbo.Teainfo SELECT \* FROM Info.dbo.Director



USE Demo

IF EXISTS(SELECT NAME FROM SYSOBJECTS WHERE NAME='Stuinfo')

DROP TABLE Stuinfo

GO

CREATE TABLE Stuinfo

(

stu\_id CHAR(9) PRIMARY KEY,

stu\_name VARCHAR(6) NOT NULL,

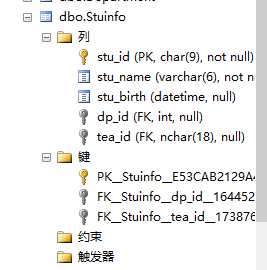
stu\_birth DATETIME NULL,

dp\_id INT FOREIGN KEY (dp\_id) REFERENCES Department (dp\_id),

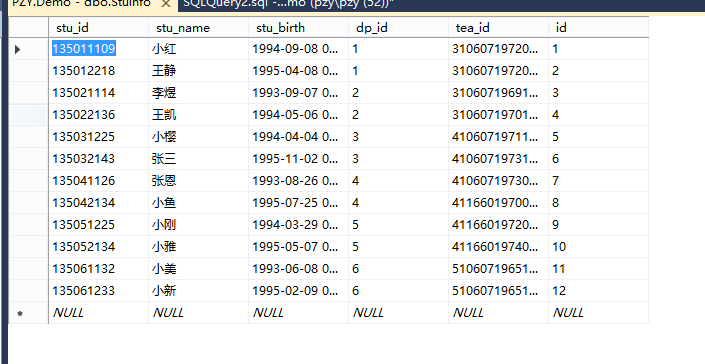
tea\_id NCHAR(18) FOREIGN KEY (tea\_id) REFERENCES Teainfo (tea\_id)

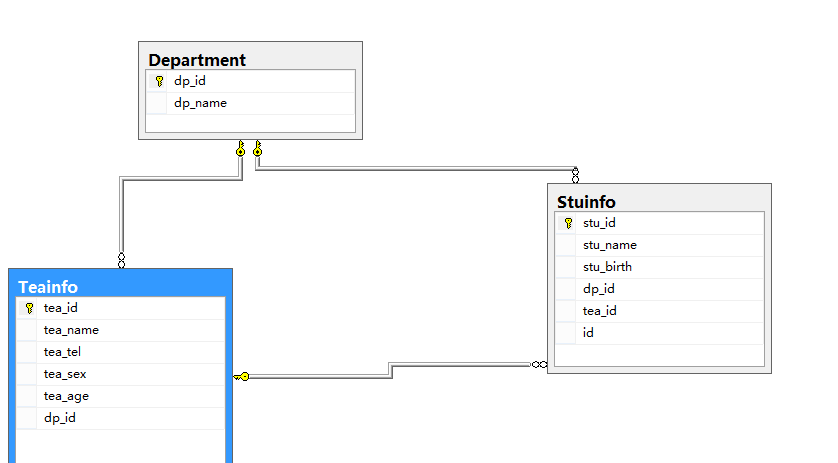
)

GO



INSERT INTO Demo.dbo.Stuinfo SELECT \* FROM Info.dbo.Student





SELECT \* INTO Demo.dbo.StuTest FROM cyzn.dbo.TourNav

USE Demo

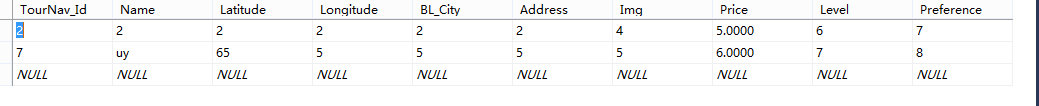
GO

SELECT \* INTO Test1 FROM StuTest

ALTER TABLE Test1

DROP COLUMN Heat --若不加入column，则默认删除约束

GO



更新数据

USE Info

GO

UPDATE Test1

SET Total=StuTest.Total

FROM StuTest

WHERE Test1.Stu\_ID=StuTest.Stu\_ID

GO

修改数据类型

USE Info

GO

ALTER TABLE Test1

ALTER column Total DECIMAL(4,1)

GO

查询：

USE Sample

GO

SELECT DISTINCT Stu\_English FROM StuTest--DISTINCT 关键词用于返回唯一不同的值。

SELECT \* FROM StuTest ORDER BY Total --默认升序 ASC|DESC：升|降

INSERT INTO StuTest VALUES('135042306','彭中耀',99,99,99,297)

UPDATE StuTest SET Stu\_name = 'PTSPZY' WHERE Stu\_Name='彭中耀'

DELETE FROM StuTest WHERE Stu\_Name='PTSPZY'

SELECT TOP 3 \* FROM StuTest--TOP用法

SELECT TOP 80 PERCENT \* FROM StuTest--TOP PERCENT用法 共18条记录 返回15条(14.4)

SELECT \* FROM StuTest WHERE Stu\_ID LIKE '%01'

SELECT \* FROM StuTest WHERE Stu\_ID LIKE '\_\_\_\_\_01'

SELECT \* FROM StuTest WHERE Stu\_ID LIKE '%[1-3]'--'%[!abc]' 最后一位不包含abc

SELECT \* FROM StuTest WHERE Stu\_ID IN (0510301,0510302)

SELECT \* FROM StuTest WHERE Total BETWEEN 250 AND 300--BETWEEN用法

SELECT Stu\_ID AS '学号',Stu\_Name AS '姓名' FROM StuTest--别名（Aliases）

-- INNER JOIN：如果表中有至少一个匹配，则返回行

-- LEFT JOIN：即使右表中没有匹配，也从左表返回所有的行

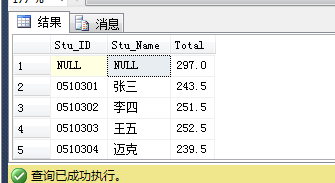
-- RIGHT JOIN：即使左表中没有匹配，也从右表返回所有的行

-- FULL JOIN：只要其中一个表中存在匹配，则返回行

-- 图解SQL Join:http://segmentfault.com/a/1190000000454332

--左表无匹配显示null

SELECT StuInfo.Stu\_ID,StuInfo.Stu\_Name,StuTest.Total FROM StuInfo RIGHT JOIN StuTest ON StuInfo.Stu\_ID = StuTest.Stu\_ID



SELECT StuInfo.Stu\_ID,StuInfo.Stu\_Name,StuTest.Total FROM StuInfo,StuTest WHERE StuInfo.Stu\_ID = StuTest.Stu\_ID

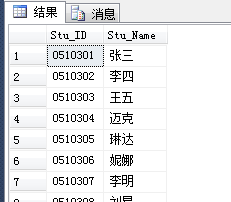
SELECT StuInfo.Stu\_ID,StuInfo.Stu\_Name,StuTest.Total FROM StuInfo FULL OUTER JOIN StuTest ON StuInfo.Stu\_ID = StuTest.Stu\_ID

--UNION 结果集中的列名总是等于 UNION 中第一个 SELECT 语句中的列名

--必须拥有相同数量的列。列也必须拥有相似的数据类型

SELECT Stu\_ID,Stu\_Name FROM StuInfo

UNIONSELECT Stu\_ID,Stu\_Name FROM StuTest;



SELECT Stu\_ID,Stu\_Spe FROM StuInfo UNION ALL--如果允许重复的值，请使用 UNION ALLSELECT Stu\_ID,Stu\_Name FROM StuTest;

USE Demo

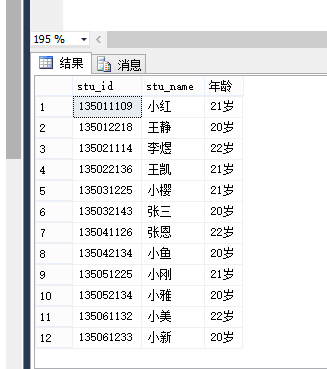
GO

SELECT stu\_id,stu\_name,

(CONVERT(varchar(20),YEAR(GETDATE())-YEAR(stu\_birth))+'岁')

AS '年龄' FROM Stuinfo

GO



USE Demo

GO

DECLARE @a INT ,@b INT,@c VARCHAR(10),@d VARCHAR(100)

SET @b=1

SET @d=''

SELECT @a=COUNT(\*) FROM Stuinfo

WHILE(@b<=@a)

BEGIN

SET @c=(SELECT stu\_name FROM Stuinfo WHERE MONTH(GETDATE())=MONTH(stu\_birth) AND id=@b )

IF (LEN(@c)<>0)

SET @d=@d+@c+''

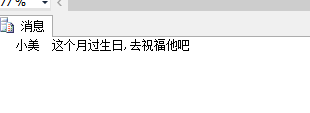
IF (@b=@a)

PRINT @d+'这个月过生日,去祝福他吧'

SET @b=@b+1

END

GO



USE Demo

GO

SELECT stu\_id,stu\_name,YEAR(stu\_birth) AS '出生年份',

YEAR(GETDATE())-YEAR(stu\_birth) AS '年龄'

FROM Stuinfo

WHERE (YEAR(GETDATE())-YEAR(stu\_birth))>20

GO



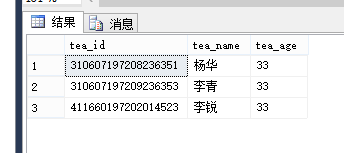
USE Demo

GO

SELECT tea\_id,tea\_name,tea\_age FROM Teainfo

WHERE tea\_age IN(SELECT tea\_age FROM Teainfo WHERE tea\_name='杨华' )

GO



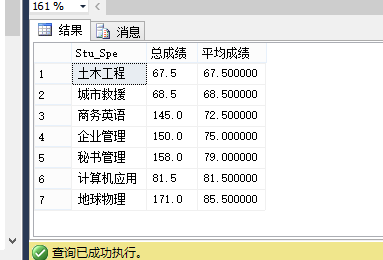
USE Demo

SELECT Stu\_Spe,SUM(Stu\_Test)AS '总成绩',AVG(Stu\_Test)AS '平均成绩'

FROM Test

GROUP BY Stu\_Spe

ORDER BY '平均成绩'



视图：

USE Demo

GO

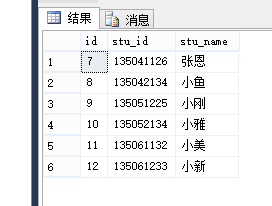
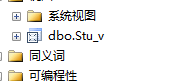
CREATE VIEW Stu\_v

AS SELECT id, stu\_id,stu\_name FROM Stuinfo

WHERE id>6

GO

SELECT \* FROM Stu\_v

USE Demo

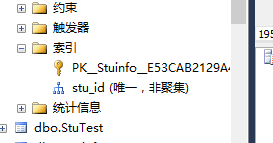
DROP VIEW Stu\_v

USE Demo

GO

CREATE UNIQUE INDEX stu\_id ON Stuinfo(stu\_id)

GO



DROP INDEX Student.Stu\_No

循环结构练习：

--乘法表显示格式优化

DECLARE @i INT,@j INT,@mul INT, @str VARCHAR(1000)

SET @i = 1;

SET @j = 1;

SET @str='';

WHILE(@i<=9)

BEGIN

SET @j = 1;

WHILE(@j<=@i)

BEGIN

SET @mul = @i \* @j;

SET @str =@str + CAST(@i AS VARCHAR(10))+'\*'+CAST(@j AS VARCHAR(10))+'='+CAST(@mul AS VARCHAR(10))+CHAR(9)

SET @j = @j + 1;

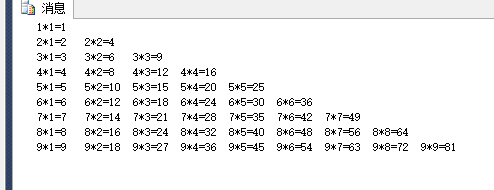
END

PRINT @str;

SET @i = @i + 1;

SET @str='';

END



游标：

DECLARE @studyNum VARCHAR(10),@Name VARCHAR(10)

DECLARE stu\_cur SCROLL CURSOR FOR

SELECT Stu\_ID,Stu\_Name FROM StuInfo

OPEN stu\_cur

--开启游标之后，默认是位于第一行的前面，因此需要FETCH NEXT

FETCH NEXT FROM stu\_cur INTO @studyNum,@Name

WHILE (@@FETCH\_STATUS = 0)--判断是否有数据

BEGIN

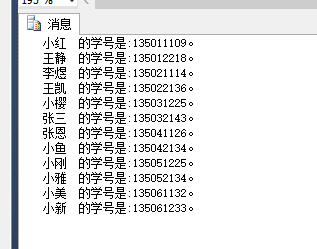
PRINT @Name +'的学号是:'+@studyNum +'。'

FETCH NEXT FROM stu\_cur INTO @studyNum,@Name--游标前进

END

CLOSE stu\_cur

DEALLOCATE stu\_cur



存储过程：

1.不含参数：

CREATE PROCEDURE pass1\_xp

AS

SELECT \* FROM StuTest WHERE Total>=250

EXEC pass1\_xp

2.含参数：

CREATE PROCEDURE pass2\_xp @pass INT

AS

SELECT \* FROM StuTest WHERE Total>=@pass

EXEC pass2\_xp 250

3.含默认参数

CREATE PROCEDURE pass3\_xp @pass INT = NULL

AS

IF @pass IS NULL

BEGIN

SELECT \* FROM StuTest WHERE Total > = 250

END

ELSE

BEGIN

SELECT \* FROM StuTest WHERE Total > = @pass

END

EXEC pass3\_xp

4.带输入参数的存储过程

CREATE PROCEDURE pass4\_xp

@pass INT = NULL,

@studyNum VARCHAR(10) OUT,

@Name VARCHAR(10) OUTPUT

AS

IF @pass IS NULL

BEGIN

SELECT \* FROM StuTest WHERE Total > = 250

END

ELSE

BEGIN

SELECT @Name=Stu\_Name,@studyNum=Stu\_ID FROM StuTest WHERE Total = @pass

END

DECLARE @pass INT,

@studyNum VARCHAR(10),

@Name VARCHAR(10)

SET @pass=297

EXEC pass4\_xp @pass,@studyNum OUT,@Name OUTPUT

SELECT @Name +'的学号是：'+@studyNum

IF EXISTS(SELECT \* FROM SYSOBJECTS WHERE name='pass4\_xp')

DROP PROC pass4\_xp

GO

USE Sample

GO

CREATE PROC pass4\_xp

@TITLE VARCHAR(40) , @SUM DECIMAL OUTPUT

AS

SELECT 'Stu\_Name' = Stu\_Name FROM t

WHERE Stu\_Name LIKE @TITLE +'%'

SELECT @SUM = SUM(Total) FROM t

WHERE Stu\_Name LIKE @TITLE +'%'

DECLARE @SUM DECIMAL

EXEC pass4\_xp '李',@SUM OUTPUT

SELECT @SUM

Demo(模糊查询查询系名并输出该系的总分)

IF EXISTS(SELECT \* FROM SYSOBJECTS WHERE name='pass4\_xp')

DROP PROC pass4\_xp

GO

USE Sample

GO

CREATE PROC pass4\_xp

@TITLE VARCHAR(40) , @SUM INT OUTPUT,@SPE VARCHAR(10) OUTPUT

AS

SELECT @SUM = SUM(Stu\_PS) FROM StuInfo

WHERE Stu\_Spe LIKE @TITLE + '%'

SELECT DISTINCT @SPE = Stu\_Spe FROM StuInfo

WHERE Stu\_Spe LIKE @TITLE + '%'

DECLARE @TITLE VARCHAR(40),@SUM INT,@SPE VARCHAR(10)

SET @TITLE='信息'

EXEC pass4\_xp @TITLE,@SUM OUTPUT,@SPE OUTPUT

PRINT @SPE +'总分为：'+ CAST(@SUM AS VARCHAR(10))

--DISTINCT尝试

DECLARE @SPE VARCHAR(10)

SELECT DISTINCT @SPE = Stu\_Spe FROM StuInfo

WHERE Stu\_Spe LIKE '信息%'

PRINT @SPE

触发器：

--触发随机数次序修改

CREATE TRIGGER tgr\_Reply\_update\_stu\_order

ON Reply

FOR UPDATE

AS

DECLARE @MaxNum INT--获取最大次序

IF (UPDATE(isOk))--判断更新

BEGIN

--每次更新，在最大值的基础上+1，stu\_order字段默认值必须为0

SET @MaxNum=((SELECT MAX(stu\_order) FROM Reply)+1)

--获取当前被更新的行，进行更新操作（利用副本表inserted ）

UPDATE Reply SET stu\_order = @MaxNum FROM Reply INNER JOIN inserted ON Reply.id = inserted.id

END