# Dml

## 自定义变量

@ 自定义变量前加@

格式: @变量 (以后@变量统称变量)

例: @a

declare 生成变量

格式：declare 变量 数据类型;

例：declare @haha int;

set 给变量赋值

格式：set 变量=常量;

例: set @haha = 999;

select @n = 999; 也可给变量赋值

go 清除之前赋值记录

例：

declare @n int;

select @n = 999;

select @n;

go

## select 查询变量

格式：select 变量;

例: select @haha; (查询自定义变量)

select id, name from student; (查询表中变量-列名)

### top 查询范围，接数值或百分比

例1: select top 3 id, name from student;

例2:select top 50percent id, name from student;

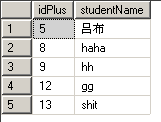
### where 后接查询条件

例1: select id ,name from student where id>5;



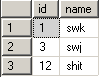
例2: select id+1 id Plus, name student Name from student where id>3 and name>'g' （where 后条件可用and,or相并列）

下图为例2输出结果

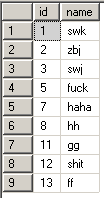


### like 逻辑运算符，后接模糊查询条件

例1：select id,name from student where name like 's%'



例2：select id,name from student where name like '[a-z]%'



例3：select id,name from student where name like 'h\_'

截图03.png

注：like 后：

‘%’ 代表0或多个任意字符

‘-’ 代表一个任意字符

[] 指该集合中任意一个字符

例：[0-9a-q] （0到9，a到q中任意一个）

[^] 指除了该集合中的字符，其他均可

例：[0-9a-q] (除了0到9，a到q外，任意一个字符)

### in 逻辑运算符，后接集合

例1: select id,name from student where id in(1,3,5,7)



例2: select id,name from student where id in(select sid from sc);

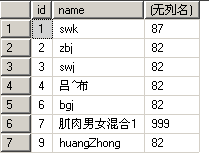


### is 查询是否为null时，最好用is

例: select id,name, sex from student where sex is null (null一般情况不参与运算)

Isnull() 函数,如果第一个变量为null，则它的值为第二个参数

例1：select id,name,isnull(age,999)from student where id<10

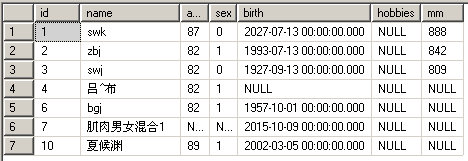


例2：select id,name, age from student where id<10 and isnull(age,999)=999

截图03.png

### exists 逻辑运算，后接表达的式

例：select\*from student where exists(select sid from sc where sc.sid = student.id)



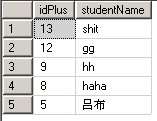
### between 逻辑运算，后接范围

例：select id, name from student where id between 3 and 9



### order by 排序，后接排序变量，可接多个

例1：select id+1 idPlus, name studentName from student where id>3 and name>'g' order by id desc (desc-降序 asc-升序)



例2：select\*from teacher

order by left(certificate,charindex('-',certificate)-1),

convert(int,right(certificate,len(certificate)-charindex('-',certificate)))



### group by 分组，后接分组字段

例1：select sc.cid,max(sc.score)

from sc,

(

select sid,cid

from sc

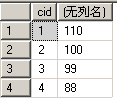
Group by sid,cid

Having count(score)<2

)haha

where(sc.sid=haha.sid and sc.cid=haha.cid )

group by sc.cid



注意： 分组之后，条件是逐组执行，不可以再逐行执行。

分组之后，不可以再选出普通字段，只能选择出统计函数和分组字段。

分组之后，用having作条件查询

### join

### left join……on…… 左边表的数据全部保留

例：select s.\*,sc.score from student s left join sc on id=sid order by s.id

### right join……on…… 右边表的数据全部保留

例：select c.\*,sc.score from sc right join course c on id=sid

### full join……on…… 两边表的数据均全部保留

例：select s.\*,sc.score from student s full join sc on id=sid order by s.id

### inner join……on…… 保留两边表的相交数据，等价于where后接连接条件

例：select s.\*,sc.score from student s inner join sc on id=sid order by s.id

### DISTINCT 查询时不选重复数据

例：select distinct sid,cid from sc

## Insert into 插入数据

例1：insert into student

(name,age,sex,birth)

values('haha',50,1,'1993-9-3')

截图02.png

例2：insert into student

values('hayhay',50,1,'1993-9-3','play game') --不指定字段时值应一一与表中格式对应，不可缺失+

截图03.png

例3：insert into student

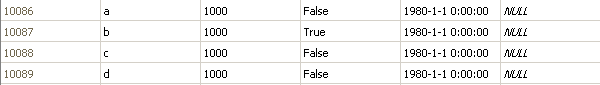
(name,sex)

select'a',0 union

select'b',1 union

select'c',0 union

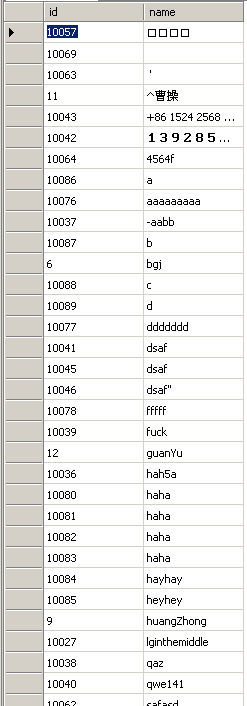
select'd',0;



表中age，birth字段有数据是因为设有默认值

例4：insert into kk

select id,name from student



## update……set…… 更新表数据

例1：update student set age=age+1;

例2：update student setname=replace(name,'a','shit')wherenamelike'%a'

例3：update sc set score=score+5 from student

where student.id=sc.cid and student.age<15

## delete……from…… 删除表数据

例1：delete a where id>100

例2：delete a

from student s

where a.id=s.id and s.name='blm'

## Select into 把查询结果生成新表

例1：select name,sex into haha from student

例2：select IDENTITY(int,1,1) haha, name,sex into shit from student where id<20

注：IDENTITY，标识函数，只能在生成表时使用

# Ddl

## Create 创建

### Create table 创建表

例1：create table fuck(

name nvarchar,

birth datetime,

age int

)

例2：create table jj(

id int not null,

name nvarchar(20),

age int,

sex bit,

primary key(id)

)

例3：create table tb (

id int IDENTITY(1,1),

name nvarchar(50),

age int

)

注：IDENTITY，标识函数，只能在生成表时使用

### Create view 创建视图

例：

create view sjc

as

select s.\*, sc.cid, sc.score from student s left join sc on s.id=sc.sid where s.id<20

### 其他

Create 可用于创建存储过程，触发器等

## Drop 删除

例1：drop table fuck

例2：drop view sjc

例3：drop procedure fuckshit

## Alter 修改，可用于修改表结构，重定义存储过程及触发器

例1：alter table student add height int, weight int; --添加列名

例2：alter table student alter column height decimal(4,1) --修改列属性

例3：alter table student drop column birth --删除例

# 流程控制

### While 循环语句

例：declare @n int;

set @n = 0;

while @n<20

begin

print @n

set @n = @n+1

end

### if 条件语句

例1：

declare @name nvarchar;

set @name = 'fjsldkjsdkjfjs'

if len(@name)>20

print 'That is a long name!'

else

print 'That is a short name!'

例2：

declare @n int

set @n = 72

if @n%2=0

begin

print @n

print '这是一个偶数！'

end

else

begin

print @n

print '这是一个基数！'

end

### case 条件语句

例1：

select id,

case

when len(name)<5 then'你的名字少于5个字符'

when len(name)>=5 then'你的名字大于或等于5个字符'

end haha, age from student

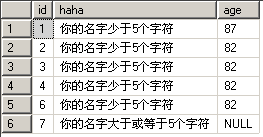
where

case

when id>7 then 'haha'

when id<=7 then 'heyhey'

end= 'heyhey'



例2：

update student set sex=

case

when avgScore>80 then 1

when avgScore>=70 then 0

else null

end

from(select sid,avg(score) avgScore from sc groupby sid)a

where student.id=a.sid and student.id<20

# Procedure 存储过程

创建例：

create procedure fuckshit @n int,@t int,@sum int output

as

select @sum=0

while @n<=@t

begin

select @sum=@sum+@n

select @n=@n+1

print @sum

end

修改例：

alter procedure fuckshit @n int,@t int,@sum int output

as

select @sum=0

while @n<=@t

begin

select @sum=@sum+@n

select @n=@n+1

print @sum

end

# trigger 触发器

例：

create trigger errorCheck on ex

for delete,update

as

begin transaction

declare @n int

select @n=score from deleted

if @n<60

begin

raiserror ('no can do',16,1)

rollback

return

end

commit transaction

go

格式：

create trigger trigger\_ame

　　on {table\_name | view\_name}

　　{for | After | Instead of }

　　[ insert, update,delete ]

　　as

sql\_statement

注：delete时会有一个叫deleted的临时表，insert时有一个inserted的临时表，update时有一个deleted和一个inserted的临时表