SQL 2014 https://pan.baidu.com/share/init?surl=tgTztj9VydneTXEgJNU

# 1、VS操作技巧

## 快速注释

### （1）、单行注释

使用 //

### （2）、多行注释

/\*

\*

\* \*\*/

小技巧：先敲/\*\*\*/然后在第一个\*后面敲回车键

### （3）、文档注释

鼠标在类上端按三个斜杠就会自动生成：

/// <summary>

///

/// </summary>

### （4）、代码折叠

使用

#region

#end region

把代码进行折叠

### （5）快速屏蔽多行代码

　　注释多行：先按 Ctrl - K 组合键，再按 Ctrl - C 组合键

　　取消注释多行：先按 Ctrl - K 组合键，再按 Ctrl - U 组合键

## 编译和调试

### （1）、编译

Ctrl+F5

### （2）、断点调试

设置断点：鼠标左键单击要加断点的语句左边

F5：设置完断点之后按F5会启动调试并调到第一个断点，在按一下就跳到下一个断点。

F10：逐步调试（不会进入函数）

F11：逐句调试

# 2、C#函数、关键字、语法

## （1）类型转换

1）、隐型转换

Int price=100;

double price 1 = price;

小的类型转换为大的类型，自动转换

2）、强制转换

double price 1 = 10.1;

Int price3=(int) price 1;

大的类型转换为小的类型，需要用到强制转换

3）、将字符串类型转换为int 或者 double

int age = int.Parse(Console.ReadLine()); //注意如果用户错误输入程序会崩溃

int age;

bool result = int.TryParse(Console.ReadLine(),out age);

//用户错误输入程序不会崩溃，只是抛出异常返回0

4）、将任意类型转换为指定类型

Convert.各种成员

## （2）、字符串的常用处理方法

IndexOf();查找某个字符在字符串中的位置（下标）

LastIndexOf();与上面的区别在于这个是从右往左数

SubString();从字符串中截取字符串

ToLower();将字符串转换成小写形式

ToUpper();将字符串转换成大写形式

Equals();比较两个字符串的值是否相等

Join();连接字符串

Split();分割字符串

Trim();去掉字符串两边空格

## （3）、集合、数组、Hashtable

数组：

**初始化方法：**

方法一：Int [] age = new int[10];

方法二：int [] age;

age = new int[5]{1,2,3,4,5};

方法三：int[] age={1,2,3,4,5}

方法四：Int [] age = new int[5]{1,2,3,4,5};

**计算数组长度：**

Int [] age = new int[5]{1,2,3,4,5};

age.Length

集合（相对于数组来说集合有一个好处就是长度可变）：

**应用命名空间：**using System.Collections;

**创建集合：**ArrayList arrayList = new ArrayList();

**往集合里面添加成员：**

方法一**;**arrayList.Add() //单个添加

arrayList.AddRangz()//多个添加

例如：

Student stu=new Student();

arrayList.Add(stu);

方法二：使用初始化器

ArrayList arrayList = new ArrayList()

{

stu

};

**集合长度：**

arrayList.Count //集合里面有几个成员

**取出集合中的成员（由于集合的形参是个object类型所以取出的值要做类型转换）**

例如：

Student stu1 =( Student) arrayList[0];

**修改集合中的成员**

Student stu1 =( Student) arrayList[0]; //修改stu1等于修改 集合的地0个成员

stu1.age=18;

**删除一个成员**

arrayList.Remove(stu);//通过放入的成员名字来删除

arrayList.RemoveAt(0);//通过下标来删除第0个成员

（4）、Hashtable 类型

创建哈希表：

Hashtable ht = new Hashtable();

Student stu1 = new Student（”1001”,”张三”,15,”101班”）;

Student stu2 = new Student（”1002”,”李四”,16,”102班”）;

Student stu3 = new Student（”1003”,”王五”,17,”103班”）;

增加：

ht.Add(“stu1”,stu1);

ht.Add(“stu1”,stu1);

删除：

ht.Remove(“stu2”);

修改：

Student stu = (Student)ht(“stu2”);

Stu.StuName = “xx”

查：

foreach (string key in ht.Keys)

{

Console.WriteLine(key);

}

（5）、List<T>泛型

优点：1、避免装箱拆箱。2、避免类型转换。

命名空间：using System.Collections.Generic

创建：

List<Student> stuList = new List<Student>()

曾加：

stuList.Add(stu1);

删除：

stuList.Remove(stu1);

修改：

stuList[0].className = “aa”;

查：

for(int i=0;i<stuList.Count;i++)

{

Student s=stuList[i];

Console.WriteLine(s.StuName);

}

（6）、Dictionary 泛型字典集合

创建：

Dictionary<string,Student>dict=new Dictionary<string,Student>();

增加：

dict.Add(“zz”,stu1);

删除：

dict.Remove(“zz”);

查找：

Student stu=dict[“zz”];

# 3、数据库

## （1）、简单的增删改查

--增加

insert into Product (ProductNb, ProductName, ProductImage, Price)

values(1005,'iphone','image/xxx.jpg',9000)

--删除

delete from Product

where ProductNb = 1005

--修改

update Product

set Price=200

where ProductNb = 1004

--查询

select \* from Product

select ProductNb,ProductName from Product

where ProductNb = 1003

## （2）、联表查询

--联表查询

--内联接 inner join

select ProductNb, ProductName, ProductImage, Price, product.typeid , producttype.typeid, typename

from Product inner join ProductType

on(Product.typeid=ProductType.typeid)

select ProductNb, ProductName, ProductImage, Price, typename

from Product , ProductType

where Product.typeid=ProductType.typeid

--使用别名的内联接

select ProductNb, ProductName, ProductImage, Price, typename

from Product a, ProductType b

where a.typeid=b.typeid

--左外联接 left join （以左边为基准，左边有的会全部显示）

select ProductNb, ProductName, ProductImage, Price , typename

from ProductType left join Product

on(Product.typeid=ProductType.typeid)

--左外联接 right join （以右边为基准，左边有的会全部显示）

select ProductNb, ProductName, ProductImage, Price , typename

from ProductType right join Product

on(Product.typeid=ProductType.typeid)

## （3）、聚合函数

--聚合函数（count sum avg max min）

-- count 统计记录条数

--使用指定列查询

select count(Product.Price) as 价格等于600 from Product

where Product.Price=600

--使用任意列查询

select count(\*) from Product

--使用伪列查询（实战中基本上用这个）

select count(1) from Product

--sum（求总和一般只能针对于数字和日期）

select sum(Product.Price) as sum\_price from Product

--avg (求平均值一般只能针对于数字和日期)

select avg(Product.Price) from Product

--max (找出最大值一般只能针对于数字和日期)

select max(Product.Price) from Product

--min (找出最小值一般只能针对于数字和日期)

select min(Product.Price) from Product

（4）、排序和分组

--排序和分组 order by group by

--排序order by [desc]降序 [asc]升序

select \* from Product

order by price desc,typeid desc

--分组 根据产品类型统计每个产品的个数

select ProductType.typename ,count(1) as num

from Product inner join ProductType

on Product.typeid=ProductType.typeid

group by ProductType.typename

order by num

注意，如果同时出现 where、group by、order by 那么order by必须放在最后面 where 应该在greop by前面

## （4）、数据库变量

--数据库变量

--自定义变量

declare @name varchar(50) = '匿名',@id int =10

set @name = '帅哥' --通常赋予固定值 set select 都可以

select @name = '张工',@id = 11 --如果同时给多个变量进行赋值只能用select 不能使用set

select @name=ProductType.typename , @id=ProductType.typeid from ProductType

print @name+'\_'+cast(@id as varchar) --cast类型转换

--查询返回多个值赋给同一个变量set会出错 而select会赋予最后一个返回值给变量

--查询不到信息set会赋予NULL值给变量而select会保持变量之前的值

--系统变量

select @@ERROR --放在语句后面捕获错误类型

--自增

insert into ProductType( typename)

values('我')

select @@identity

--查询当前语言

select @@LANGUAGE

--编译器版本

select @@VERSION

## （5）、数据库事务和程序事务

1）、事务的特性

原子性：不可分割（多个操作）要么一起执行，要么都不执行。

一致性：当数据操作完成，数据保持一致性。

隔离性：每一个事务都是个独立的个体，相互之间是隔离的（不相干）

持久性：事务提交数据会永久保存，永久存储到物理磁盘（数据库文件）

2）、常见的3种的数据库事务

自动提交事务：sqlserver 默认开启而oralce默认关闭；在sqlserver种的insert、delete、update都是自动提交事务。平时 程序sqlconnection关闭的时候也会自动提交。

显示事务：

开始事务 begin transaction

编写事务脚本

commit transaction提交事务 或者 rollback transaction回滚事务

eg:

--转账业务

begin transaction –开始事务

declare @errorsum int = 0 –默认没有错误

update Account

set UserMoney= UserMoney-2000 –如果值是负数会报错

where CardID=1001 --张三

set @errorsum=@errorsum+@@error –只要update报错error就不等于0

update Account

set UserMoney= UserMoney+2000

where CardID=1002 –李四

set @errorsum=@errorsum+@@error –只要update报错error就不等于0

if @errorsum<>0 --其中<>是不等于的意思

begin

print ‘转账失败’

rollback; --回滚

end

else

begin

print ‘转账成功’

commit; --提交

end

隐式事务：sqlserver 默认是关闭状态而oralce是开启状态。隐式事务开启的时候自动提交事务就不会自动提交。（有自带开启事务、没有提交事务）

--开启隐式事务 set implicit\_transaction on

--关闭隐式事务 set implicit\_transaction off

## （6）、触发器

### 1）、增删改查会分别产生两个临时表

inserted 临时表 :在执行insert动作的时候会产生

deleted 临时表 ：在执行delet动作时会产生

注意：在执行update 动作底层会执行delete 和 insert动作

2）、触发器实例

--insert

create trigger insert\_Recorlog

on product

for insert -- for 是受约束限制（比如之前的主表与子表关联），此处也可以使用delete update

as

begin

declare @uerid int --自定义一个用户编号

select @userid=typeid form inserted

insert into logs(logContent,createuserid)

values('添加动作',@userid)

end

--delete删除

create trigger delect\_Recorlog

on ProductType

instead of delete -- instead of 不受约束限制这是和之前for的区别，此处也可以使用delete update

as

begin

declare @uerid int --自定义一个用户编号

select @userid=typeid form inserted

delete from product where typeid=@userid

delete form producttype where typeid=@user

end

注意：触发器第一次创建用create 修改的时候是使用alter

## （7）、用命令行操作数据库

1）、连接SQL：进入数据库的bin路径：mysql -hlocalhost -uroot –proot

查看里面有哪些数据库：show databases

删除指定的数据库：drop database web1

2）、创建一个数据库：create database web;

3）、使用这个数据库：use database; 或者 use web

4）、查看数据库里面所有的表 : show tables;

查看单表admin里面的内容 desc admin

5）、创建表

管理员表admin(ID、登入名、密码)：

create table admin(

id int primary key auto\_increment,

username varchar(20),

password char(32)

);

注意：

primary key 主键

auto\_increment自动增长

单页模板表 board（ID、模块名、内容）

create table board(

id int primary key auto\_increment,

boardname varchar(20),

content text

);

注意：

text 文本

新闻表：news（ID、标题、分类、作者、内容、头图、发布时间、点击率）

create table news(

id int primary key auto\_increment,

title varchar(50),

cate\_id int default 0,

author varchar(30),

content text,

img varchar(50),

intime timestamp default current\_timestamp,

hlts int default 0

);

注意：

时间轴：timestamp

当前时间：current timestamp

产品表product（ID、名称、编号、分类、详细说明、产品分类、产品图、发布时间、点击率）

create table product(

id int primary key auto\_increment,

productname varchar(40),

pro\_no varchar(30),

cate\_id int default 0,

content text,

img varchar(50),

intime timestamp default current\_timestamp,

hlts int default 0

);

分类表category(ID、分类名、排序号)

create table category(

id int primary key auto\_increment,

catename varchar(30),

orderno tinyint unsigned default 5

);

注意

tinyint 是小数字的意思 0-255

友情链接表 flink(ID 、链接名、网址、说明、时间)

create table flink(

id int primary key auto\_increment,

title varchar(20),

link\_url varchar(50),

content varchar(50),

intime timestamp default current\_timestamp

);

留言表questbook(ID、用户名、内容、留言时间)

create table questbook(

id int primary key auto\_increment,

username varchar(30),

content text,

intime timestamp default current\_timestamp

);

6）、删除表admin

drop table admin