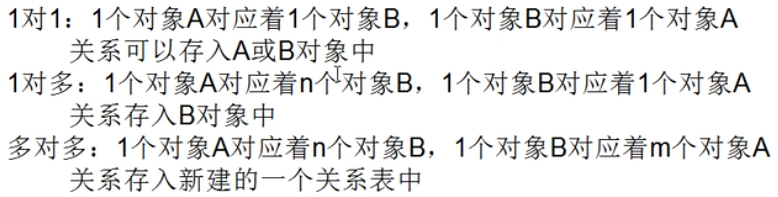
数据库设计

三范式：列不可拆分(到最细)，唯一标识，引用主键(引用唯一标识)

关系及存储：



简单操作：

创建数据库 ：删除数据库 分离，附加数据库 生成脚本(架构和数据)

创建表 ：创建主键--int--设置标识

列名的属性char/varchar/nvarchar/int(11位好像已经越界了，电话号一般用char)/bit(只能存0||1)/datetime(日期时间)/Decimal(一共要n个数字，m个小数)(小数) decimal(5,2) 100-100.00 99-99.00 11199.999-(199.99) date只要日期 time只要时间

约束： 实现数据的有效性检查

主键：(保证值是唯一的) 设置时要注意设置 属性标识主键

非空

唯一 (主键一定唯一，唯一不一定主键)保证这一列的所有行没有重复的，而且主键不允许为空值，唯一键允许空值

索引和键---添加---类型唯一键---列选择所选列

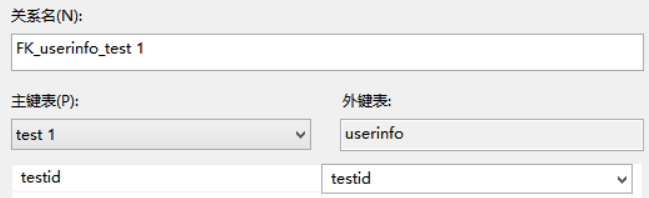
默认 (用户不输入时会有一个默认值) 默认值或绑定---添加 ’ ~~~~ ’

检查 (检查约束) 必须满足约束条件 check约束--添加--表达式 

外键 对关系的合理性做一个有效性检查 (比如一个学生属于3班,但只有2个班) 外键一定是某个表的主键

给关系列做的约束叫外键约束 例如学生表的班级列和班级表 学生表的班级是由班级表的班级决定的 班级表是主键表 学生表是外键表 关系--添加---表和规范

引用的是外键表 被引用的是主键表 外键建立在外键表上 关系1vN时要存入N的一方

此时就会根据主键表的数据要求判断外键表的引用是否规范

脚本操作

不区分大小写，字符串使用单引号，末尾无需分号

注释： -- 单行注释

数据库： 只有创建，删除

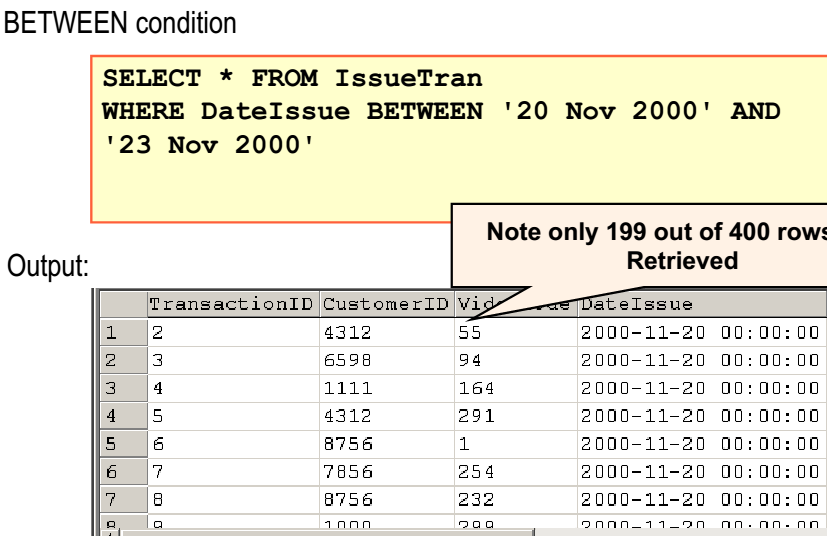
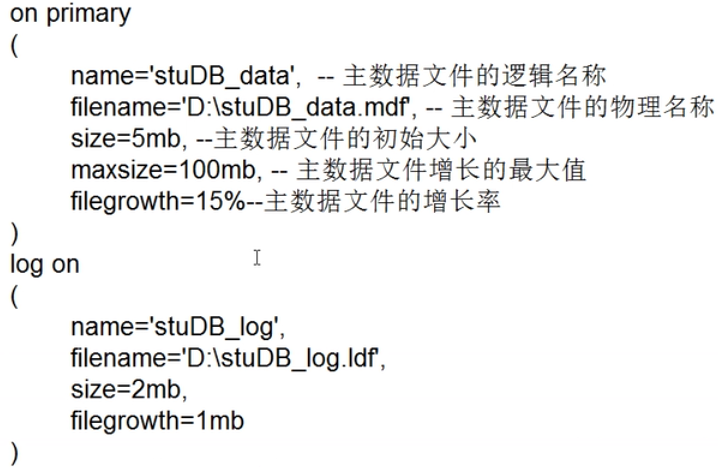
通过查看master数据库中sysdatabases表，了解当前存在的数据库

查询数据库 ：点击新建查询

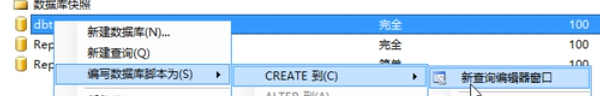
使用某个数据库  查询一个数据库

删除一个数据库

Create database 123 123数据库名 创建数据库



写完后选中所写代码点击执行后 在目标区域就会出现数据库文件 选中哪句执行哪句

忘记代码了 通过这个可以看看应该怎么创建数据库 标准代码

脚本创建表

创建表

Use dbtest ----在这个数据库中创建

Create table classinfo

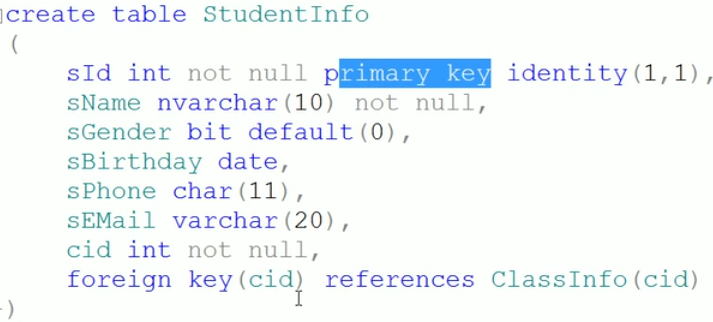
(

cid int not null primary key identity(1,1), ----主键属性

Ctitle nvarchar(10)， ----列属性

~~~ ----用逗号分隔不同行

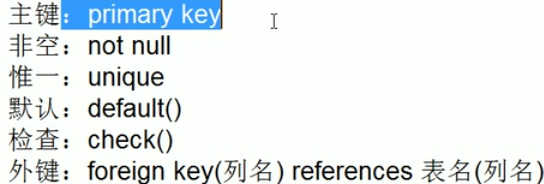
)



不写not null 就是可以空值 default(0)不输入默认值为0

 在本表的cid上有一个外键 引用classinfo里的cid

别忘了最后选中所有点执行

 check() 括号里需要写代码 此代码需要返回一个bool类型的值

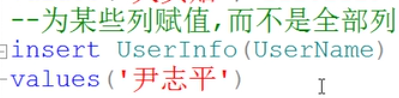
数据插入

Insert 表名(列名) values( ‘ ’ , ’ ’ )

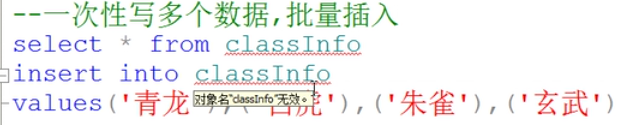


若不写列名 代表为所有列按默认顺序赋值





一次写入多个数据



INSERT INTO 表名称 VALUES (值1, 值2,....) 向表格中插入新的行

INSERT INTO table\_name (列1, 列2,...) VALUES (值1, 值2,....) 指定所要插入数据的列

INSERT INTO table\_name (列1, 列2,...) SELECT列1, 列2 FROM table\_name2 插入数据从一个表到另一个表中

UPDATE 表名称 SET 列名称 = 新值 WHERE 列名称 = 某值 修改表中选择的数据

UPDATE GoodCustomers SET PhoneNumber = 7775588 修改所有行

UPDATE GoodCustomers SET PhoneNumber = 7775588 WHERE CustomerID in

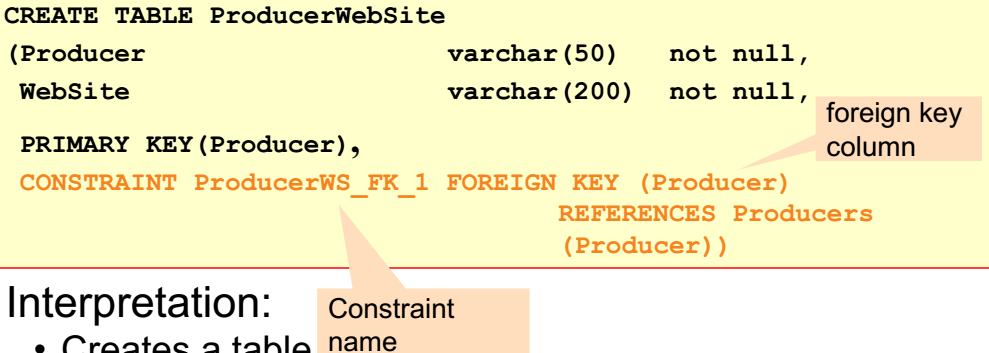
(SELECT CustomerID FROM Customers WHERE MemberCategory = ‘B’) 使用子查询更新

DELETE FROM 表名称 WHERE 列名称 = 值 删除某行

DELETE FROM table\_name 不删除表的情况下删除所有的行，意味着表的结构、属性和索引都是完整的

DELETE FROM GoodCustomers WHERE CustomerID in

(SELECT CustomerID FROM Customers WHERE MemberCategory = ‘A’) 使用子查询删除



CREATE TABLE table\_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,

(op) Primary key ( column1 ) ,

Constrain Constraint\_name Foreign key( column1) references table\_name (column) ); 创建新表

可以选择创建主键外键约束 约束包括：非空，唯一，外键等等

CREATE INDEX index\_name ON table\_name (column\_name) 在表中创建索引 允许使用重复的值

CREATE UNIQUE INDEX index\_name ON table\_name (column\_name) 在表中创建唯一索引 不允许使用重复的值

Drop TABLE table name/INDEX index\_name ON table\_name 删除表/删除索引

ALTER TABLE 语句用于在已有的表中添加、修改或删除列

ALTER TABLE table\_name ADD column\_name datatype 表中添加列

ALTER TABLE table\_name ADD PRIMARY KEY (CountryCode) 添加主键

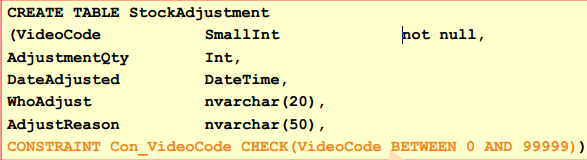
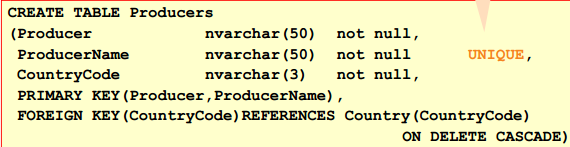
ALTER TABLE table\_name ADD FOREIGN KEY (CustomerId) REFERENCES table\_name2 (CustomerId) 添加外键

ALTER TABLE table\_name DROP COLUMN column\_name 删除表中的列

ALTER TABLE table\_name DROP Primary key constraint name 删除主键

ALTER TABLE tableName ALTER COLUMN columnName VARCHAR(10) 修改列属性

Checking constraint Entity Integrity Constraint

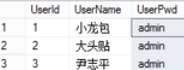
 

videocode 只可以接收范围内的值 此列的每一行都必须是不重复的值(可用主键或unique约束)

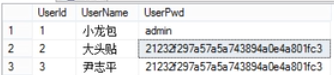
数据修改 update

Update 表名 set 列名= ’ ’

 修改此列的所有行



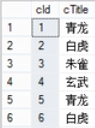
修改此列的指定行



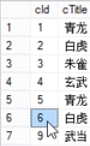
删除数据 delete

Delete 表名 where 注：删除引用外键的列会被报错





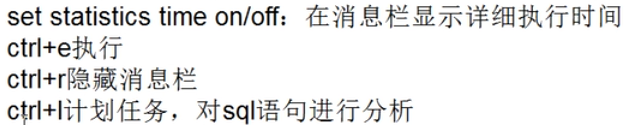
此时再加入数据

 会出现跳号

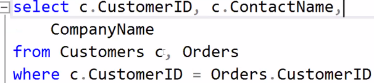
清空

Truncate table userinfo 清空此userinfo表 重置所有 此时再插入数据从1开始 注：删除引用外键的表会被报错

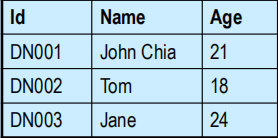
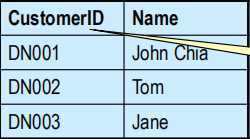
辅助键



***查询***

别名 当两个表格中共同存在同一键时 需要xxx.xx

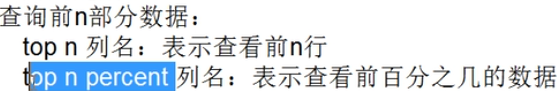
别名SELECT Id AS CustomerID, Name FROM Customers

..as.. 别名 将…改名为… SELECT CustomerID, Name FROM Customers显示这两列

Select \* from xxx 查询显示xxx中所有列 \*代表所有列

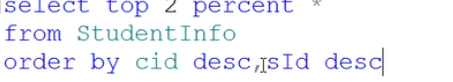
查询ui 指定列 userinfo ui 将其改名为ui

 select top 1 from xxx 查第一行数据

百分会向上取整 （往多取整）

SELECT \* FROM Customers ORDER BY Age 按年龄排序 默认升序

 asc升序 desc降序

取前百分之二的数据 先用cid的降序，当cid相同时用sid降序

 select distinct cid from xxx 代表选择cid列后消除重复项 此时只会显示cid列

条件查询 where

查询sid为1



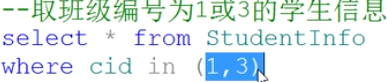
只想要一个属性的话



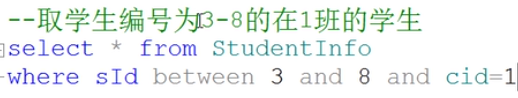
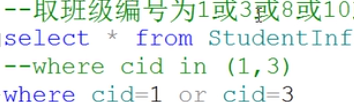
Between…and.. 查询一个连续的范围(闭区间)



In 在一个非连续的范围

也可以（1,2,3）

And or not



一般not会跟（）表示非（）

模糊查询

Like关键字

%代表零至多个任意字符 \_表示一个任意字符 []表示一位在一个范围内的字符[1,2,3]表示此位上只能是1/2/3 ^写在[]里开头表示非的含义，代表不在这个范围内

%和\_写在[]里表示本身的含义 在[]里写一个连续的范围的话可以使用- 如[0-9]

 结果1(1-4)45664878798979

Sub query

SELECT \* FROM a WHERE CustId IN (SELECT CustId FROM b) 选择b中的custid在a中的 （）里的称为sub query

Sub query 会先执行 返回一系列行 注意最后的结果是a表中的符合条件行

检索null值

SELECT \* FROM Customers WHERE Email IS NULL 检索出Email为null的

Count

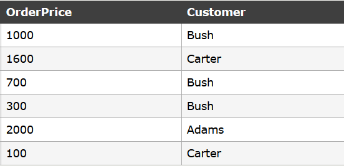
SELECT Count(\*) AS Total FROM Customers 表示返回一个叫total的表 里面有表中记录数

SELECT Count(列名) AS Total FROM Customers 返回一个叫total的表 里面有指定列中值的数目 null值不计入

Average

SELECT AVG(Age) AS AvgAge FROM Customers 返回一个叫avgage的表 表中有age的平均数

例 找到OrderPrice 值高于 OrderPrice 平均值的客户



SELECT Customer FROM Orders

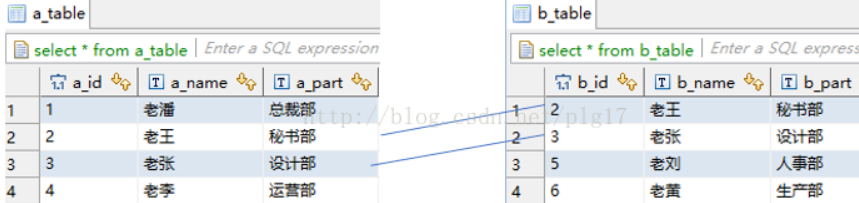
WHERE OrderPrice>(SELECT AVG(OrderPrice) FROM Orders)

Is null 和is not null 查询null值是能用is null 形式 不可以查询xxx=null

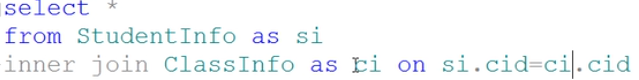
设置表中第5和6的数据为null



连接查询：当需要的结果需要从多张表中取时



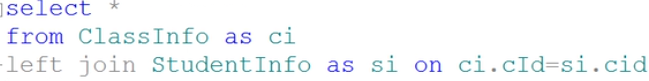
内连接 inner join (两表中完全匹配的数据)

结果为2个表中所有数据

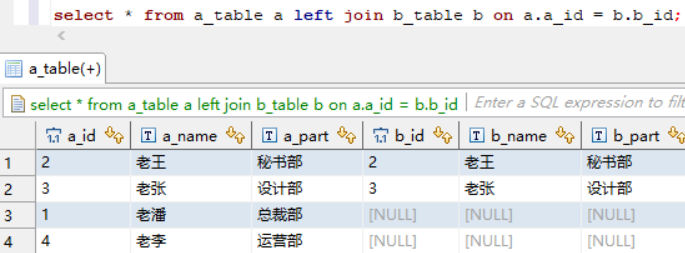
 会有重复数据

展现si和ci内连接的所有，通过cid=cid的内连接 xxx inner join yyy on 111=222

左连接 left join (两表中完全匹配的数据+左表特有的)

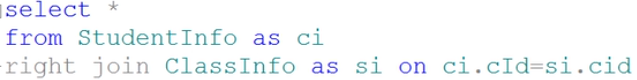


结果多了左表特有的

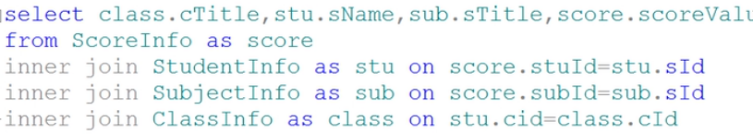


xxx left join yyy on 111=222

右连接 right join (两表中完全匹配的数据+右表特有的)

结果多了右表特有的

完全外链接：full outer join 两表中完全匹配的数据+左表和右表特有的



多表共同查询，想插入一个表 找这个表和已知表的共同列就可以插入了

聚合函数

Count： Select count(\*) as count1 from student 查询这个表中一共有多少行

Select count(\*) as count1 from student where cid=1 查询cid=1的一共有多少行

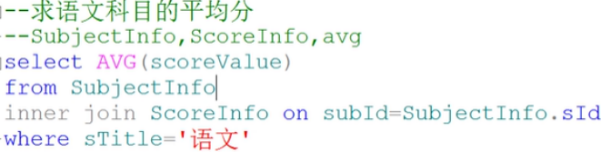
若其中一行所有列都为空 那么其不计数 或者只count(xx)列 其中xx里的null行不计数

Max ： 查询某列的最大值

Min同理 1号同学的最低分

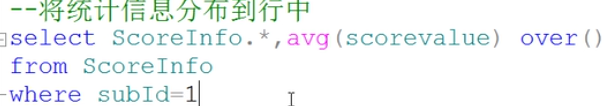
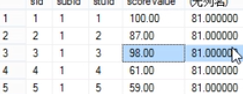
Top SELECT TOP 2 CustomerID FROM Customers 取前两行

Avg

 SELECT AVG(Age) AS AvgAge FROM Customers

开窗函数 将统计信息分布到行中 over() 写在聚合函数后

将聚合函数的值还原到原来的每一行中显示

分组 group by 列名1，列名2 根据指定列进行分组

分组后条件筛选 having

聚合函数一般结合分组使用，进行分组内的数据进行统计

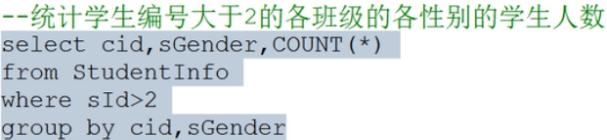


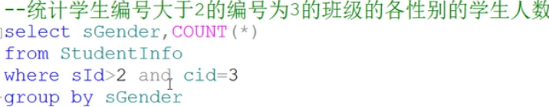
多列分组



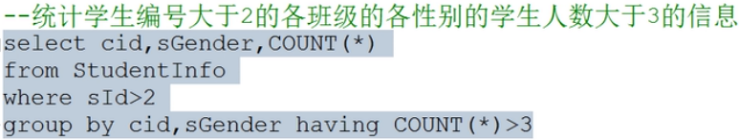
先用subid(性别)进行分组 之后再使用stuid(地区)进行在分组 显示每个再细分组后的各个平均值

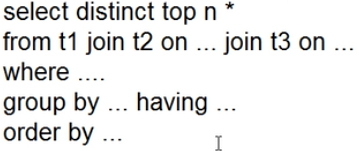
相当于 性别和地区一样的分一组 计算各组平均值

先从总数据中拿出满足where条件的数据，再将数据进行分组，再计数



若想再分组之后再想进行条件筛选 需要在group by之后 加having



 顺序 from---where---group by---having---distinct---order by---top---得到你想要的列\*

联合查询 对多个结果进行再处理 将每个结果集进行合并---行的连接

联合要求 各个结果集列数一致 对应列的类型要一致---就是联合后同一列不能类型不同

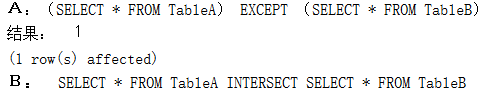
Union 将多个结果集数据进行合并 并消除重复行 并且按第一列进行升序排列

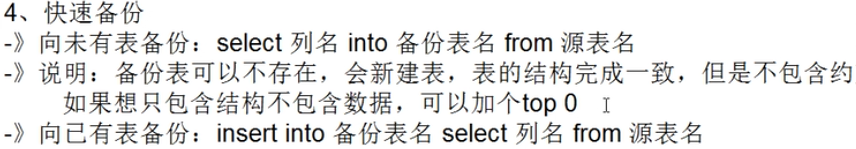
Union all 只会将多个结果集数据进行合并

Except 将第一个结果集的内容从第二个结果集进行排除，剩下的那些内容

Intersect 交集 两个结果集一样的地方

A 12345 b 2345

结果2345

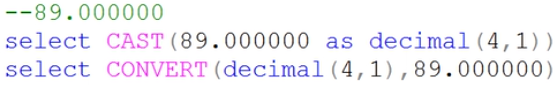
不包含约束 主键，外键 像已有表备份时不能\*全插



类型转换

Cast (表达式as 转换类型 (长度) ) int 无长度 字符串有长度的

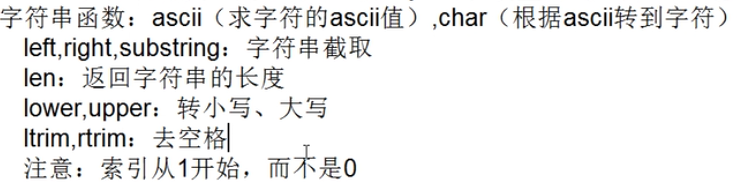
Convert (目标类型(长度) ，表达式 )

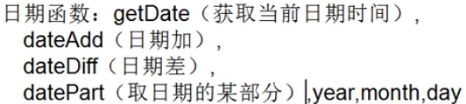


C#中 +号有重载 将1与 ’ 1 ’相连接时 会自动将非字符串值转换为字符串进行拼接

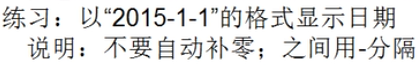
Sql中 直接会得出结果为2 将非int值转换为int



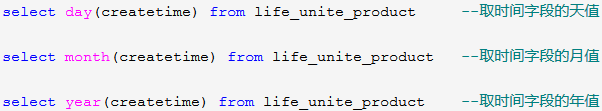




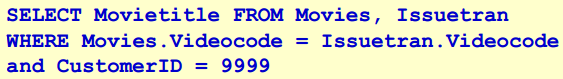
 可以不写双引号 返回值为int

原本为2015-01-01

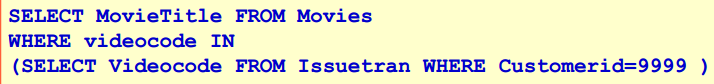
select [CONVERT](https://www.baidu.com/s?wd=CONVERT&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)(varchar(6) , getdate(), 112 )   
varchar里面放长度 输出就行  
只输出年就varchar(4)  
只输出年月就varchar(6)  
只输出年月日就varchar(8)



Inner join 可以简写为



Sub query



结果相同

SELECT \* FROM [table name] 表格名中有空格的需要[]括起来

