传智播客 www.itcast.cn —高级软件人才实作培训专家

SQL

讲师：伍湖

什么是数据库

- 数据库Database：数据的仓库---文件
- 仓库里有排货架(表)，货物会分类存放。比如牙膏、牙刷会放到一个排货架上，面包会单独放在一个排货架上。
- 仓库又分很多种粮仓、武器仓库。
- 仓库中还会有仓库管理员(DBA)对货物进行管理
- 从仓库中拿货物需要凭证，取货和进货的人(程序员)

我们怎么存储数据

- 学校的档案室
- 计算机文件, 文件操作，开发人员需要熟悉操作磁盘文件的函数、必须编写复杂的搜寻算法才能高效的把数据从文件中检索出来、当数据格式发生变化时，需要编写复杂的文件格式升级程序、很难控制并发修改。
- 数据库（其实也是文件），DBMS(数据库管理系统)\RDBMS
- 对于数据不仅仅是需要存储，更重要的是将数据进行存储以后怎么才能方便快捷的查询、修改。
- 数据库特点：海量存储、查找速度快、并发性问题控制、安全性、数据完整性(保存在数据库中的数据是正确的，真实的)。

数据库概述

- DBMS（DataBase Management System，数据库管理系统）和数据库。平时谈到“数据库”可能有两种含义：MSSQLServer、Oracle等某种DBMS；存放一堆数据表的一个分类（Catalog）。
- 不同品牌的DBMS：MYSQL（中型数据库，开源，免费，速度很快，适合对数据要求并不是十分严谨的地方，去掉了很多中小型企业中不常用的功能）、MSSQLServer（大中型数据库，与.net结合很好）、DB2（大型）、Oracle（大型）、Access（文件）、SQLite（极其轻量级数据库）、Sybase等。对于开发人员来讲，大同小异
- SQL(语言)<>SQLServer<>MSSQLServer。最常见的错误。
- 除了Access、SQLServerCE、SQLite等文件型数据库之外，大部分数据库都需要数据库服务器才能运行。学习、开发时是连接本机的数据库，上线运行时是数据库运行在单独的服务器。

数据库运行简易流程

- 查询分析器，查询优化器，查询执行器
- 缓存管理器
- 缓存

数据库中的概念

- 数据库DataBase，不同类的数据应该放到不同的数据库中
  - 便于对各个数据类别的进行个性化管理 （分布式部署）
  - 避免命名冲突
  - 安全性更高
- Table（表）：关系数据库中的关系指的就是表。不同的货物要放到各自的货物架，将这种区域叫做“表” （Table）。不同的表根据放的数据不同进行空间的优化，找起来也方便。---实体类
- 列（Column）
- 用表格格式化数据：即便是引入了自动识别设备也很容易识别。

2003年5月入职，是产品开发部的，姓名马虎

王二小，技术支持部，入职是2005年7月

姓名	马虎
部门	开发部
入职时间	2008.06.06

姓名	部门	入职时间
马虎	开发	2003
王二小	技术支持	2005
马虎	开发	2003

主键（PrimaryKey）

工号	姓名	部门	入职时间
001	风姐	员工培训部	2010年7月5日
002	瘦瘦	公关部	2010年8月2日
003	憨憨	开发部	2009年3月5日

主键就是数据行的唯一标识。不会出现重复数据的列才能当主键。一个表可以没有主键，但是会非常难以处理，因此没有特殊理由表都要设定主键

主键有两种选用策略：***业务主键*和逻辑主键**。业务主键是使用有业务意义的字段做主键，比如身份证号、银行账号等；逻辑主键是使用没有任何业务意义的字段做主键，完全给程序看的，业务人员不会看的数据。因为很难保证业务主键不会重复（身份证号重复）、不会变化（帐号升位），因此**推荐用逻辑主键**。

表间关联、外键（ForeignKey）

商品名	价格	生产厂家	厂家地址	厂家电话
大大香瓜子	5.00	大大食品厂	恰恰大街300号	010-123456
大大开心果	15.00	大大食品厂	恰恰大街300号	010-123456
苦咖啡	2	伊利食品厂	内蒙古伊利路1号	400400400
随变	3	伊利食品厂	内蒙古伊利路1号	400400400
冰工厂	1	伊利食品厂	内蒙古伊利路1号	400400400

外键

商品名	价格	厂家编号
大大香瓜子	5.00	1
大大开心果	15.00	1
苦咖啡	2	2
随变	3	2
冰工厂	1	2

主键

编号	名称	地址	电话
1	大大食品厂	恰恰大街300号	010-123456
2	伊利食品厂	内蒙古伊利路1号	400400400

SQLServer的管理

- 需要安装SQLServer2012或者SQLServer2008，若要使用SQLServer管理工具进行开发还要安装SQL Server Management Studio，还可以使用VisualStudio进行管理
- 使用免费的SQLServerExpress版本，Express版本的服务器名称. \SQLEXPRESS，对于开发人员来讲和其他版本没有区别。. .\sqlexpress 127.0.0.1 localhost 计算机名称
- SQLServer的两种验证方式：用户名验证和Windows验证，开发时用Windows验证就行。
- 开发人员关注点在开发上，而不是配置、备份等之上，那是DBA做的事情。

创建数据库

- 创建数据库，创建表，设置主键
- 数据库的分离和附加（以及 脱机联机）
- MS SQLServer的每个数据库包含：
  - 1个主数据文件(.mdf)必须。
  - 1个事务日志文件（.ldf）必须。
- 可以包含：
  - 任意多个次要数据文件(.ndf)
  - 多个事务日志文件
- 文件组：可将多个数据文件逻辑的分到一组，方便日后管理维护（备份、将表建在指定的文件组上等等。）
- 创建
  - <部门表>：部门Id,部门名称
  - <员工表>：员工Id,身份证号，姓名，性别，入职日期，年龄，地址，手机号，所属部门、Email

分类	备注和说明	类型	说明
二进制数据类型	存储非子符和文本的数据	Image	可用来存储图像
文本数据类型	字符数据包括任意字母、符号或数字字符的组合	Char,8000	固定长度的非 Unicode 字符数据。固定长度的字符串相对于可变长度的字符串来说效率要高一些，在数据长度固定的情况下优先选用固定长度，省去了计算长度的过程，提高效率
		Varchar,8000	可变长度非 Unicode 数据
		Nchar,4000	固定长度的 Unicode 数据
		Nvarchar,4000	可变长度 Unicode 数据
		Text 2^31-1 varchar(max)	存储长文本信息(指针,2G) varchar(max)，大字符串类型可以保存非常多的字符，但是对于这种类型的数据DBMS经常将它们保存到单独的空间中，这就导致了数据的保存和加载速度比较慢，因此除非必要，否则不要使用。
		Ntext nvarchar(max)	Nvarchar(max)代替
日期和时间	日期和时间在单引号内输入	Datetime	日期和时间
数字数据	该数据仅包含数字，包括正数、负数以及分数	int smallint	整数
		float real	数字
货币数据类型	用于十进制货币值,money 和 smallmoney 数据类型精确到它们所代表的货币单位的万分之一。	Money(C#:double)	

练习

- 创建一个HeiMaBlog数据库。
- 创建一个班级表：
  - Class：Id （班级编号，自动编号，主键）、Name（班级名称）、Descr（班级简介）。
  - 创建一个学生信息表：
    - Student：Id（学生编号，自动编号，主键）、Name（学生姓名）、Gender（性别）、Address（家庭地址）、Phone（电话）、Age（年龄）、Birthday（出生日期）CardId（身份证号）、CId（班级Id）
- 分离数据库
  - 在需要分离的数据库上点右键-任务-分离
- 附加数据库(在其他计算机上，亲自测试！)
  - 在数据库节点上点右键-附加
- 打开数据之前，要打开数据库服务

SQL语句入门(脚本、命令)

- **SQL** 全名是结构化查询语言（Structured Query Language），是关系数据库管理系统的标准语言
- SQL语句是和DBMS“交谈”专用的语句，不同DBMS都认SQL语法。
- SQL语句中字符串用**单引号**、**单等号**。
- select *、SeLeCT *:
  - SQL语句是大小写不敏感的，不敏感指的是SQL关键字，字符串值还是大小写敏感的
- 建库、删除数据库、创建表、删除表不仅可以手工完成，还可以执行SQL语句完成，在自动化部署、数据导入中用的很多
- 简单的Insert语句。
- （*） SQL主要分**DDL（数据定义语言,建表、建库等语句。）**、DML（数据操作语言multipulation）和DCL （数据库控制语言）。Create Table、Drop Table、Alter Table等属于DDL，Select、Insert、Update、Delete等属于DML， GRANT 授权、REVOKE 取消授权属于DCL

使用sql语句创建数据库和表

- 使用SQL语句创建School数据库、TblClass表、TblStudent表。
- Go：
  - 将T-SQL语句分批发送到数据库实例执行。

T-SQL创建数据库的语法：

```
CREATE  DATABASE  数据库名
 ON [PRIMARY]
 (
  <数据文件参数> [，…n]
 )
 [LOG ON]
 (
  <日志文件参数> [，…n]
 )
```

创建数据库示例 1

```
CREATE DATABASE HeiMaBlog
   ON  PRIMARY  --默认就属于PRIMARY主文件组，可省略
(
 NAME=  'HeiMaBlog',  --主数据文件的逻辑名
 FILENAME='D:\ HeiMaBlog_data.mdf',  --主数据文件的物理名
 SIZE=3mb,  --主数据文件初始大小
 MAXSIZE=10mb,  --主数据文件最大的值
 FILEGROWTH=15%   --主数据文件的增长率
)
LOG ON - – 日志文件
(
  NAME='HeiMaBlog',
  FILENAME='D:\HeiMaBlog_log.ldf',
  SIZE=3mb,  --日志文件初始大小
  MaxSize=20mb,
  FILEGROWTH=1MB
)
GO
```

建表

```
USE HeiMaBlog--将当前数据库设置为HeiMaBlog
GO
CREATE  TABLE  Score
(
   ScoreId INT IDENTITY(1,1),
   SId  INT  NOT  NULL ,
   English  INT  NOT  NULL,
   Math  INT  NOT  NULL
   --Name Varchar(50)  not null
)
GO
```

创建表练习

- 创建数据库TestSchool
- 创建学生成绩表ScoreScoreId（成绩id,主键,自动编号）、SId（学生编号）、English（英语成绩）、Math（数学成绩）

- 创建老师表Teacher
- Id、Name、Gender、Age、Salary、Birthday

约束-保证数据完整性（数据检查）

- 先用设计器创建约束、再用代码创建约束。
- 数据库约束是为了保证数据的完整性(正确性)而实现的一套机制
- 见文件Employee.sql
- 非空约束
- 主键约束(PK) primary key constraint 唯一 且 不为空
- 唯一约束 (UQ)unique constraint 唯一，允许为空，但只能出现一次
- 默认约束 (DF)default constraint 默认值
- 检查约束 (CK)check constraint 范围以及格式限制
- 外键约束 (FK)foreign key constraint 表关系：保证外键值来源于主键
- 增加外键约束时，设置级联更新、级联删除：
  - [**ON DELETE** { NO ACTION | **CASCADE** | SET NULL | SET DEFAULT }]
  - [**ON UPDATE** { NO ACTION | **CASCADE** | SET NULL | SET DEFAULT }]

练习

- Teacher表中
  - Gender 控制只能是男 女，默认女
  - Age 在30-40之间 默认30
  - 唯一键
  - 默认值
- Score表中
  - studentId 是外键 先要把Student表中的sId设置为主键
  - 测试外键约束：
  - 1：在学生表（主表）中删除在成绩表中被引用的学生记录。
  - 2：成绩表中添加一条新成绩，studentId在 学生表中没有。
- 保存SQL脚本。再次打开即可执行。

数据插入

- 向表中插入一行（该行的每一列都有数据）
  - insert into 表【列名1，列名2】 values(值1，值2)
  - insert语句可以省略表名后的列名，但是不推荐。
    - Insert into 表 values(值1，值2)
- 插入数据时，只向某些列插入数据：如果插入的行中有些字段的值不确定，那么Insert的时候不指定那些列即可。
  - Insert into 表(列1) values(值1)
- 自动编号列不需要手动插入。【SET IDENTITY_INSERT 表名 ON 】
- 注意：主键不能有重复值。
- **插入数据时的单引号问题。**
- Insert into 表（列） select 列1，列2 union select 列1，列2 from 表
- Select 列 into 新表名 from 旧表
- N前缀：N’字符串’，在服务器上执行的代码中（例如在存储过程和触发器中）显示的 **Unicode 字符串常量**必须以大写字母 N 为前缀。即使所引用的列已定义为 Unicode 类型，也应如此。如果不使用 N 前缀，可能导致不识别某些字符。

数据更新(数据修改)

- 更新一个列：update Student set sSex = ‘男’
- 更新多个列： update Student set sSex ='女',sAge = 18,sBirthday='1989-8-8'
- 更新一部分数据： update Student set sClassId= 4 where sClassId = 1，用where语句表示只更新Name是’ tom’ 的行，注意SQL中等于判断用单个=，而不是==。
- Where中还可以使用复杂的逻辑判断update Student set sAge=30 where sName='华佗' or sAge<25 ， or相当于C#中的||（或者）
- 所有学生的年龄加1 update Student set sAge = sAge + 1
- update Student set sClassId=6
- where (sAge>20 and sAge<30) or(sAge=50)
- Where中可以使用的其他逻辑运算符：(||)or、(&&)and、(!)not、<、>、>=、<=、 <>（不等）等

数据删除

- 删除表中全部数据：DELETE FROM Student。
- Delete只是删除数据，表还在，和 Drop Table不同。
- Delete 也可以带where子句来删除一部分数据：DELETE FROM Student WHERE sAge > 20
- ==========
- truncate table student 的作用与delete from student一样，都是删除student表中的全部数据，区别在于：
  - 1.truncate语句非常高效。由于truncate操作采用按最小方式来记录日志，所以效率非常高。对于数百万条数据使用truncate删除只要几秒钟，而使用delete则可能耗费几小时。
  - 2.truncate语句会把表中的自动编号重置为默认值。
  - 3.truncate语句不触发delete触发器。

练习：

- 针对提供的数据库表：
- 练习1：给studentId是1的英语成绩加10分
- 练习2：考试题偏难，所有人的成绩加5分
- 练习3：所有女学生的年龄减1岁

- 删除年龄大于30岁的学员信息

- ==========将老师表清空=========
- 删除所有老师
- 删除数据时候 把自增长列的值还原成种子

约束-保证数据完整性（数据检查）

- 先用设计器创建约束、再用代码创建约束。
- 数据库约束是为了保证数据的完整性(正确性)而实现的一套机制
- 见文件Employee.sql
- 非空约束
- 主键约束(PK) primary key constraint 唯一 且 不为空
- 唯一约束 (UQ)unique constraint 唯一，允许为空，但只能出现一次
- 默认约束 (DF)default constraint 默认值
- 检查约束 (CK)check constraint 范围以及格式限制
- 外键约束 (FK)foreign key constraint 表关系：保证外键值来源于主键
- 增加外键约束时，设置级联更新、级联删除：
  - [**ON DELETE** { NO ACTION | **CASCADE** | SET NULL | SET DEFAULT }]
  - [**ON UPDATE** { NO ACTION | **CASCADE** | SET NULL | SET DEFAULT }]

练习

- Teacher表中
  - Gender 控制只能是男 女，默认女
  - Age 在30-40之间 默认30

- Score表中
  - studentId 是外键 先要把Student表中的sId设置为主键
  - 测试外键约束：
  - 1：在学生表（主表）中删除在成绩表中被引用的学生记录。
  - 2：成绩表中添加一条新成绩，studentId在 学生表中没有。
- 保存SQL脚本。再次打开即可执行。

数据检索

- 执行备注中的代码创建测试数据表。
- 简单的数据检索： SELECT * FROM Student
- 只检索需要的列： SELECT sName FROM Student 、SELECT sName,sAge FROM Student
- 列别名：SELECT sName AS 姓名,sAge AS 年龄,sBirthday AS 出生日期 FROM Student
- 使用where检索符合条件的数据：SELECT sName FROM Student WHERE sSex= ‘女’ 。
- 还可以检索不与任何表关联的数据：select 1+1;select select getdate();

Top、Distinct

- Top 获取前几条数据，top一般都与order by连用
  - 获得年纪最小的5个学生
  - 获得年纪最大的10%的学生
- Distinct 去除重复数据
  - select distinct sName from student
  - select distinct sName,sAge from student

聚合函数

- SQL聚合函数:
- MAX（最大值）、MIN（最小值）、AVG （平均值）、SUM （和）、COUNT（数量:记录的条数。）
- 聚合函数对null值不计算。
- 如果一行的数据都是null，count(*)包含对空值行、重复行的统计。
- 平均成绩select avg(english) from score
- 男学生出生日期的最大值和最小值：select max(sBirthday),min(sBirthday) from student where sSex='男'

带条件的查询

- Select …from…where …
  - 查询没有及格的学生的学号 select studentno from result where studentResult<60
  - 查询年龄在20-30岁之间的男学生
- Between…and …在之间
  - 查询年龄在20-30岁之间的男学生
  - 查询math成绩在80-90分之间的所有学生
  - 建议：优先使用between … and …，而不是“列>=值1 and 列<=值2”，between … and …已做过优化处理，效率高。
- 查询班级id为1，2，3的所有学生
  - select sName,sAge from student where sClassId=1 or sClassId=2 or sClassId=3
  - select sName,sAge from student where **sClassId in (1,2,3)**

带条件的查询-模糊查询(都是针对字符串操作的)

- 查询所有姓张的同学
  - Select * from student where left(sName,1)= ‘张,如果改成查询名字中带亮的学生怎么做?
- 换一种做法 like
  - Select * from student where sName like ‘张%’ 会吧所有姓张的都查询到，现在我想查询姓张并且名字是一个字的学生?
  - Select * from student where sName like ‘%亮%’
- _(单个任_意字符) 、 %（0 个或者多个任意字符） 、 [](代表一个具体的或者连续的范围) 、 ^(取反值) [^0-9]
- ^只有MSSQL Server支持，其他DBMS用not like。
- 通配符 %多字符匹配的通配符，它匹配任意次数（零或多个）出现的任意字符
- 通配符_ 单字符匹配，它匹配单个出现的字符
- [] 只匹配一个字符 并且这个字符必须是[]范围内的 [0-9] [a-z]
- not与like一起使用：**not like** ....
- 要通配_、%、[、^这些字符怎么办？[_]、[%]、[[ ]、^(要放到中括号里，因为^只有放到中括号中才认为是通配符)

空值处理

- 数据库中，一个列如果没有指定值，那么值就为null，数据库中的null表示“**不知道**”，而不是表示没有。因此select null+1结果是null，因为“不知道”加1的结果还是“不知道”。
- select * from score where english = null ;
- select * from score where english != null ; 都没有任何返回结果，因为数据库也“不知道”。
- SQL中使用is null、is not null来进行空值判断：
  select * from score where english **is null** ;
  select * from score where english **is not null** ;
- ISNULL (check_expression , replacement_value)

数据排序

- ORDER BY子句位于SELECT语句的末尾，它允许指定按照一个列或者多个列进行排序，还可以指定排序方式是升序（从小到大排列，ASC）还是降序（从大到小排列，DESC）。
- 按照年龄升序排序所有学生信息的列表：SELECT * FROM Student ORDER BY sAge ASC
- 按照英语成绩从大到小排序，如果英语成绩相同则按照数学成绩从大到小排序： SELECT * FROM Score ORDER BY english DESC,math DESC
- ORDER BY子句要放到整个语句的最后： SELECT * FROM Score where english>=60 and math>=60 ORDER BY english DESC,math DESC
- Order by 语句一般要放到where语句的后面，就是先让其他语句进行筛选，全部筛选完成后，最后排序一下。
- 表中数据是集合，集合是没有顺序的。Order by 返回的数据是有顺序的，故此我们把order by 以后返回的数据集合叫“游标”。

--1.查询六期班所有姓 陈 的学员
--2.查询所有科目中包含c 字符的科目信息
--3.查询office最近一次考试时间

数据分组-统计信息！！！

- 在使用select查询的时候，有时需要对数据进行分组汇总（即：将现有的数据按照某列来汇总统计），这时就需要用到group by语句。select 语句中可以使用group by 子句将行划分成较小的组，然后，使用聚组函数返回每一个组的汇总信息。//分组一般都和聚合函数连用。
- 1.请从学生表中查询出每个班的班级Id和班级人数：（见备注1）
- 2 .请从学生表中查询出每个班的班级Id和班级中男同学的人数：（见备注2）
- GROUP BY子句必须放到WHERE语句的之后 ,Group By与Order By都是对筛选后的数据进行处理，而Where是用来筛选数据的。
- 没有出现在GROUP BY子句中的列是不能放到SELECT语句后的列名列表中的 （聚合函数中除外）
  - 错误： select sClassId,count(sName),sAge from student group by sClassId
  - 正确： select sClassId,count(sName),avg(sAge) from student group by sClassId

Having语句-对分好的组做条件筛选

- 对表中的数据分组后，会得到一个分组后的结果集，如何对该结果集在进行筛选？→ having
- 查询班级人数超过三个人的班级。（见备注1）
- 注意Having中不能使用未参与分组的列，Having不能替代where。作用不一样，Having是对组进行过滤。
- Having 是Group By的条件对分组后的数据进行筛选（与Where类似，都是筛选，只不过having是用来筛选分组后的组的。）
- 在Where中不能使用聚合函数，必须使用Having，Having要位于Group By之后。
- Having的使用几乎是与where一样的，也可以用in。
  - Having count(*) in (5,8,10)

SQL语句的执行顺序

- 5>...Select 5-1>选择列,5-2>distinct,5-3>top（确定列）
- 1>...From 表 （确定表）
- 2>...Where 条件（确定行）
- 3>...Group by 列（分组）
- 4>...Having 对组来做筛选条件（对组做筛选）
- 6>...Order by 列（排序,在已经存在 的结果集上进行记录的重排）

Group by 练习

- 查询：
  - 1.查询每个班级的总学时数，并按照升序排列
  - 2.查询每个参加考试的学员的平均分
  - 3.查询每门课程的平均分，并按照降序排列
  - 4.查询每个班级男女生的人数

类型转换函数

- CAST (expression AS data_type)
- CONVERT (data_type, expression,[style])
- Select ‘您的班级编号’ + 1 错误这里+是数学运算符
- **SELECT** FIdNumber,
- **CAST**(**RIGHT**(sNo,3) **AS INTEGER**) **as** 后三位的整数形式,
- **CAST**(**RIGHT**(sNo,3) **AS INTEGER**)+1 **as** 后三位加1,
- **CONVERT**(**INTEGER**,**RIGHT**(sNo,3))/2 **as** 后三位除以2
- **FROM** student
- 对编号排序，但编号是字符串类型。1、2、11、3、21、36…
- 对日期的转换。转换成各种国家格式的日期。
  - select convert(varchar(**20**),getdate(),**104**)
  - Style的格式，查sql帮助。（输入convert函数查询）
  - 将日期转换为指定格式的字符串。**日期→字符串**

联合结果集union(集合运算符)

- 集合运算符是对两个集合操作的，两个集合必须具有相同的列数，列具有相同的数据类型（至少能隐式转换的），最终输出的集合的列名由第一个集合的列名来确定。（可以用来连接多个结果）
- 联合(union)与连接(join)不一样
- 简单的结果集联合(老师、学生)：
  - select tName,tSex from teacher union
  - select sName,sSex from student
- 基本的原则：每个结果集必须有相同的列数；每个结果集的列必须类型相容。
  - select tName,tSex,-1 from teacher union
  - select sName,sSex,sClassId from student
- **联合：将多个结果集合并成一个结果集。union(去除重复，相当于默认应用了distinct)、union all**
- **常见应用：底部汇总。**

Union all

- select tName,tSex from teacher union
- select sName,sSex from student

UNION合并两个查询结果集，并且将其中完全重复的数据行合并为一条

- select tName,tSex from teacher union all
- select sName,sSex from student

Union因为要进行重复值扫描，所以效率低，因此如果不是确定要合并重复行，那么就用**UNION ALL**

案例1

- 要求在一个表格中查询出学生的英语最高成绩、最低成绩、平均成绩
- 查询结果为：
  - select 'english最高成绩',max(english) from score
  - union all
  - select 'english最低成绩',min(english) from score
  - union all
  - select 'english平均',avg(english) from score
- 查询结果：
- Select max(english),min(english),avg(english) from score;

案例2

- 查询每位老师的信息，包括姓名、工资，并且在最后一行加上平均工资和最高工资
- 底部汇总：
  - select tName,tSalary from teacher
  - union all
  - select '平均工资',avg(tSalary) from teacher
  - union all
  - select '最高工资',max(tSalary) from teacher

一次插入多条数据

- insert into Score(studentId,english,math)
- select 1,80,100 union
- select 1,80,100 union
- select 3,50,59 union all
- select 4,66,89 union
- select 5,59,100
- 此处如果用union all同样会去除重复数据。

*一次插入多条数据

- --把现有表的数据插入到新表（表不能存在）,为表建备份。
- --select * into newStudent from student（newStudent表在select查询的同时自动建立。）
  - --把现有表的数据复制到一个已存在的表
  - 通过这种方式复制，只能复制表中的数据，以及列的名字和数据类型。对于约束，不会复制过来。
- Select * into newTbl from oldTbl where 1<>1,这样做可以只复制表结构，但效率并不高。建议：select top 0 * into newTbl from oldTbl
- --insert into backupStudent select * from students(backupStudent表必须提前建好)

字符串函数（*）

- LEN()：计算字符串长度（字符的个数。）
- datalength();//计算字符串所占用的字节数，不属于字符串函数。
  - 测试varchar变量与nvarchar变量存储字符串a的区别。见备注1.
- LOWER()、UPPER ()：转小写、大写
- LTRIM()：字符串左侧的空格去掉
- RTRIM ()：字符串右侧的空格去掉
- LTRIM(RTRIM(' bb '))
- LEFT()、RIGHT() 截取取字符串
  - SELECT LEFT('abcdefg',2)
- SUBSTRING(string,start_position,length)，索引从1开始。

参数string为主字符串，start_position为子字符串在主字符串中的起始位置，length为子字符串的最大长度。SELECT SUBSTRING('abcdef111',2,3)

尝试使用SQLServer的帮助。

字符串函数

函数名	描　　述	示　　例
CHARINDEX	寻找一个指定的字符串在另一个字符串中的起始位置	SELECT CHARINDEX('JBNS','My Jbns Course',1) 返回：4
LEN	返回传递给它的字符串长度	SELECT LEN('SQL Server课程') 返回：12
UPPER	把传递给它的字符串转换为大写	SELECT UPPER('sql server课程') 返回：SQL SERVER课程
LTRIM	清除字符左边的空格	SELECT LTRIM (‘ 涛哥 ') 返回：涛哥（后面的空格保留）
RTRIM	清除字符右边的空格	SELECT RTRIM (‘涛哥 ') 返回：涛哥（前面的空格保留）
RIGHT	从字符串右边返回指定数目的字符	SELECT RIGHT('买卖提.吐尔松',3) 返回：吐尔松
REPLACE	替换一个字符串中的字符	SELECT REPLACE('莫乐可切.杨可','可','兰') 返回：莫乐兰切.杨兰
STUFF	在一个字符串中，删除指定长度的字符，并在该位置插入一个新的字符串	SELECT STUFF('ABCDEFG', 2, 3, '我的音乐我的世界') 返回：A我的音乐我的世界EFG

数学函数

函数名	描　　述	示　　例
RAND	返回从 0 到 1 之间的随机 float 值	SELECT RAND( ) 返回：0.79288062146374
ABS	取数值表达式的绝对值	SELECT ABS(-43) 返回：43
CEILING	取大于或等于指定数值、表达式的最小整数	SELECT CEILING(43.5) 返回：44
FLOOR	取小于或等于指定表达式的最大整数	SELECT FLOOR(43.5) 返回：43
POWER	取数值表达式的幂值	SELECT POWER(5,2) 返回：25
ROUND	将数值表达式四舍五入为指定精度	SELECT ROUND(43.543,1) 返回：43.500
SIGN	对于正数返回+1，对于负数返回-1，对于0则返回0	SELECT SIGN(-43) 返回：-1
SQRT	取浮点表达式的平方根	SELECT SQRT(9) 返回：3

日期函数

- **GETDATE()**：取得当前日期时间
- **DATEADD** (datepart , number, date)，计算增加以后的日期。参数date为待计算的日期；参数number为增量；参数datepart为计量单位，可选值见备注。DATEADD(DAY, 3,date)为计算日期date的3天后的日期，而DATEADD(MONTH ,-8,date)为计算日期date的8个月之前的日期 。（入职一年以上的员工发1000$）
- **DATEDIFF** (datepart , startdate , enddate)：计算两个日期之间的差额。datepart 为计量单位，可取值参考DateAdd。
- 统计不同入学年数的学生个数：
  - select DateDiff(year,sInDate,getdate()),count(*) from student Group by DateDiff(year,sInDate,getdate())
- **DATEPART** (datepart,date)：返回一个日期的特定部分
- Month()、year()、day()来代替。
- 统计学生的生日年份个数：
  - select DatePart(year,sBirthday),count(*)
  - from student
  - group by DatePart(year, sBirthday)
- 1990年出生的人的个数？ Select count(*) from P where year(birthday)=1990

日期函数

函数名	描　　述	示　　例
GETDATE	取得当前的系统日期	SELECT GETDATE() 返回：今天的日期
DATEADD	将指定的数值添加到指定的日期部分后的日期	SELECT DATEADD(mm,4,'01/01/2009') 返回：以当前的日期格式返回05/01/2009
DATEDIFF	两个日期之间的指定日期部分的间隔	SELECT DATEDIFF(mm, '01/01/2009', '05/01/2009') 返回：4
DATENAME	日期中指定日期部分的字符串形式	SELECT DATENAME(dw, '01/01/2000') 返回：Saturday或星期六
DATEPART	日期中指定日期部分的整数形式	SELECT DATEPART(day, '01/15/2000') 返回：15

练习

- --查询年龄超过20周岁的6期班的学生信息。
- --查询1月份过生日的学生信息
- --查询今天过生日的学生姓名及所在班级
- --查询学号为“10”的学生Email的域名。
- --新生入学，为其分配一个Email地址，规则如下：GZ+当前日期+4位随机数+@itcast.com