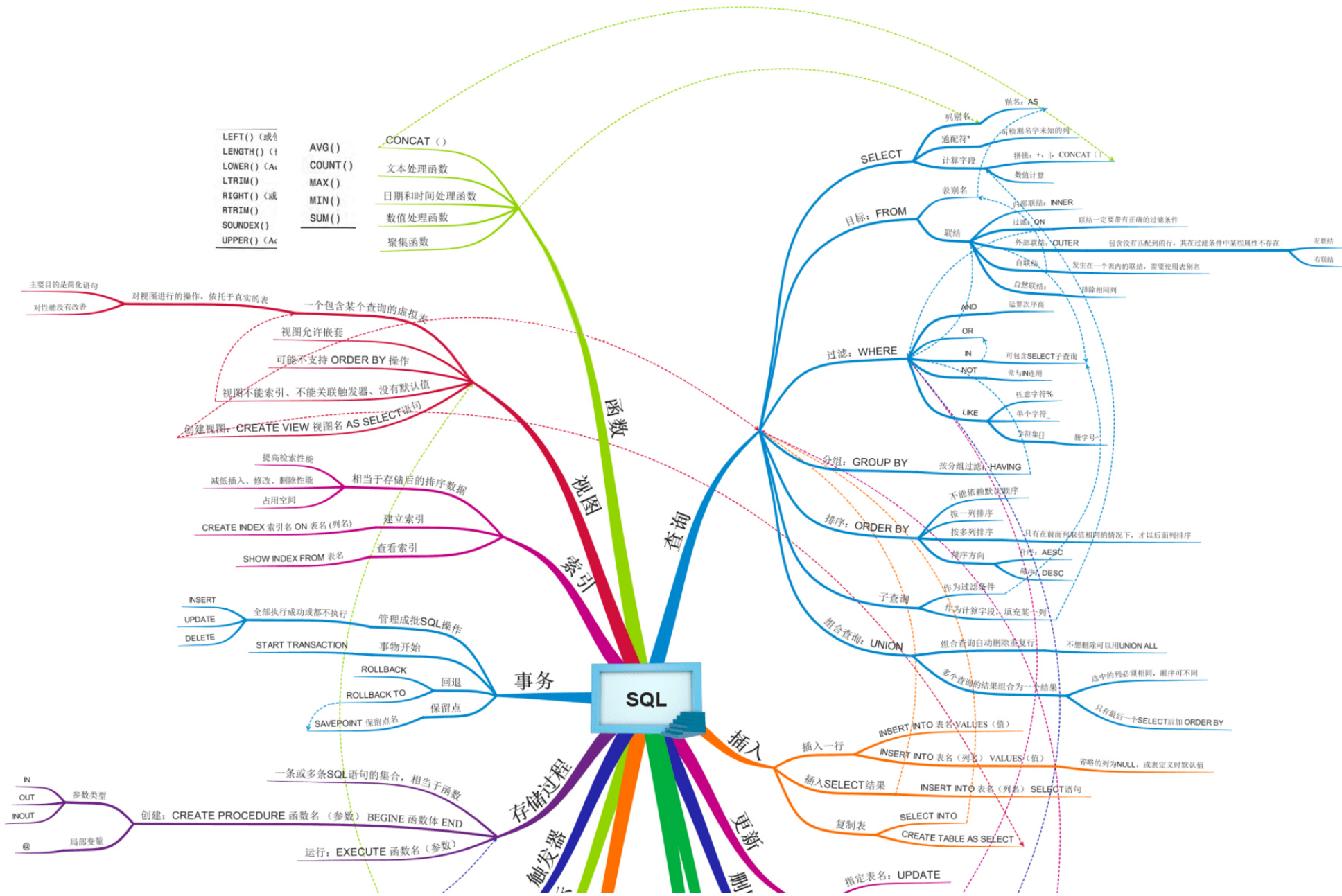
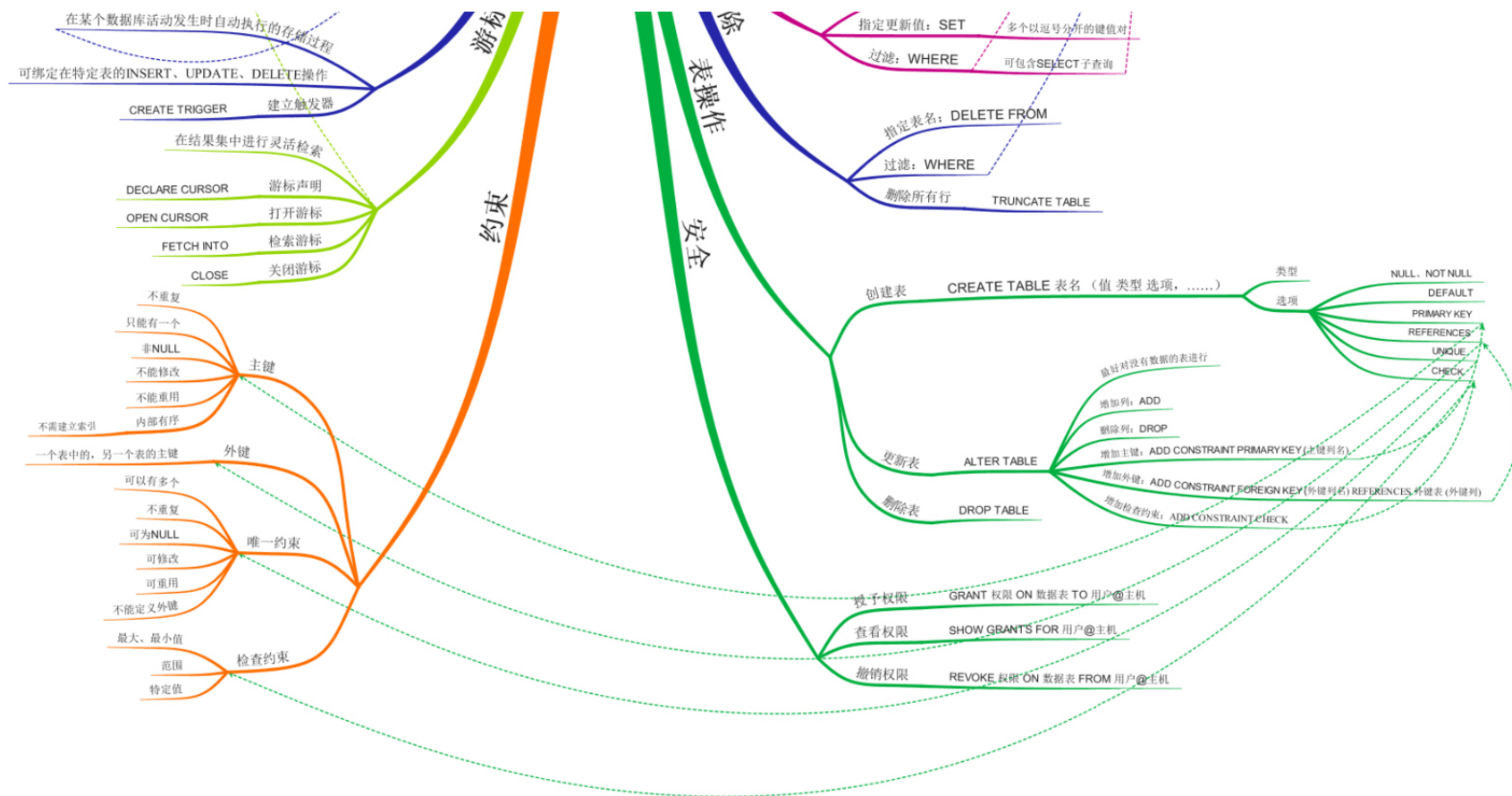


SQL 的学习路径脑图和常用 SQL 语句





功能	代码
查询列	select 顾客姓名,.... from tb_顾客表
查询全部信息	select * from tb_顾客表
换标题	select 顾客编号 '编号',

	顾客姓名 '姓名', 所在城市 '城市' from tb_顾客表
添加列	select 顾客姓名,邮编,电话,传真, (电话+传真) AS 逗, (邮编+电话) AS 逼 from tb_顾客表
区间查询,选择数据范围	select 顾客姓名,邮编,电话,传真 from tb_顾客表 where 邮编 > 131000 select * from mrbooks where 图书价格 between 68 and 88
模糊查询	select * from mrbooks WHERE 图书价格 LIKE '_9'
前五名	select TOP 5 * from tb_BookSell

	ORDER BY 金额 DESC
后五名	select TOP 5 书号,书名,SUM(金额) AS 合计销售金额 from tb_BookSell GROUP BY 书号,书名,作者 ORDER BY 3
查询结果升(降)序	select * from tb_employee05 order by 工资,奖金
姓名首字母	select * from tb_abstu05 order by substring(国籍,1,1)
姓氏笔画	select * from tb_stu05 order by 姓名 collate chinese_prc_stroke_cs_as_ks_ws
查询数学不大于90, 或音乐大于等于95	select ID,Name,Math_Score,Music from tb_stuscore where not (Math_Score >= 90) OR (Music_Score>=95)
80-89之间	SELECT * FROM tb_StuScore

	WHERE Math_score LIKE '8_'
	WHERE Math_score LIKE '9[5-9]%'
不在90-99之间的人	SELECT * FROM tb_StuScore WHERE Math_score LIKE '[^9][5-9]%'
姓李的人	select * from tb_StuScore where 姓名 LIKE '李%'
去重	SELECT DISTINCT 书号,书名,作者,出版社 FROM tb_BookSell ORDER BY 书号
列出重复超过1次的	SELECT 书名,书号,作者,COUNT(书名) AS 重复数量 FROM tb_Booksell GROUP BY 书名,书号,作者 HAVING COUNT(书名)>=2
查询备注不为空的信息	select 学生姓名,所在学院,备注 from tb_stu

	where 备注 is Not NULL
格式化信息	<pre>select *'姓名:' + Name +';编 号:' + CONVERT(varchar(3),id) AS 学 生信息 from tb_stu</pre>
去除空格	<pre>select 姓名,LTRIM(姓名) AS去除左面空 格 RTRIM(姓名) AS去除右面空格 LTRIM(RTRIM(姓名)) AS去除左右面空 格 from 客户信息表</pre>
四舍五入	<pre>select 人员姓名,ROUND(代扣税,-1) AS 代扣税 ROUND(本月扣零) AS 本月扣零 CAST(ROUND(应发合计,1) AS real) AS 应发合计 from 工资表</pre>

功能	代码
编号和姓名字符串尾部去掉空格	select 编号,rtrim (编号)as 去空格后, 姓名,rtrim(姓名)as 去空格后, 专业,rtrim(专业)as 去空格后 from tb_stu0690
学生信息表中截取姓和名字	select 姓名,substring(姓名,1,1) as 姓, substring(姓名,2,2)as 名字 from tb_stu05
书名+别类 生成信息	select 图书名称,图书分类, (图书名称+ '的类别为' + 图书分类) as 说明 from tb_aspnetbook
将某个数字插入编号中(第二个删除)	select 学号,姓名, stuff(学号,2,2,200900) as 新学号 from tb_stu05
计算字符串p出现的次数	select len('apples is in the apple tree')- len(replace ('apples is in the apple tree','p',''))/len('p') as a from tb_name06
大写改为小写	select upper(firstname)as up_firstname ,

	<pre>lower(lastname)as low_lastname from tb_name06</pre>
元音字符删除	<pre>select firstname, replace(replace(replace(replace(replace(firstname,'A',''),'E',''),'I',''),'O',''),'U','') as 修改后 from tb_name06</pre>
分离元音字符和数字	<pre>declare @i nvarchar(50);set @i=""; declare @j nvarchar(50);set @j=""; select @i=@i+b.c from(select substring(a.n,iter.pos,1) as c from (select 'asdf1*h456' as n)a , (select id as pos from tb_num06)iter where iter.pos<=len(a.n))b where ascii(b.c) between 48 and 57; select @j=@j+b.c from(select substring(a.n,iter.pos,1) as c from (select 'asdf1*h456' as n)a ,</pre>

	<pre>(select id as pos from tb_num06)iter where iter.pos<=len(a.n))b where ascii(b.c) not between 48 and 57; select @i as 数字,@j as 字符</pre>
去除非字母和数字	<pre>declare @i nvarchar(50);set @i=""; select @i=@i+b.c from(select substring(a.n,iter.pos,1) as c from (select 'as#df*4,56' as n)a , (select id as pos from tb_num06)iter where iter.pos<=len(a.n))b where ascii(b.c) between 48 and 122; select @i</pre>
比较姓名	<pre>select (case when 'Helen'='Helan' then '相同' else '不同' end) as 比较结果 select (case when 'Patricia'='Patricia' then '相同' else '不同' end) as 比较结果</pre>
比较相似(发音相似)	<pre>select difference('green','greene')as 结果, difference('visual','basic')as 结果, difference('wear','where')as 结果</pre>
外籍学生姓名大小写区分	<pre>select *</pre>

从表中提取记录中的一部分	<pre> from tb_num06 where 姓名='MARY' collate Chinese_PRC_CS_AI </pre>
去除分机号'-'	<pre> select 电话号码 as 删除前, stuff (电话号码,charindex('-',电话号码),1,'') as 删除后 from tb_tel06 </pre>
提取区号和分机号	<pre> select 电话号码 as 提取前, left (电话号码,charindex('-',电话号码)-1)as 区号, substring(电话号码,charindex('-',电话号码)+1, len(电话号码)-charindex('-',电话号码))as 号码 from tb_tel06 </pre>
格式化日期 YY-MM-DD	<pre> select 图书名称,图书分类,出版日期, convert(varchar(10) , cast(出版日期 as smalldatetime),120)as 格式化日期 from tb_aspnetbook </pre>
求平方根	<pre> SELECT A ,SQRT(ABS(A)) AS SQRT FROM tb_Numbers </pre>
e的a次方	<pre> SELECT A ,EXP(A) AS EXP FROM tb_Numbers </pre>
三角函数	<pre> SELECT A,COS(A) AS MY_COS,SIN(A) AS MY_SIN,TAN(A) AS MY TAN </pre>

	FROM tb_Numbers
10月份的信息	select 图书名称,图书分类,出版日期 from tb_aspnetbook where month(出版日期)=10 order by 出版日期
今天是周几	select datename(dw,getdate()) as 今天是周几
时间差	select datediff(day,(select 入校时间 from tb_student06 where 姓名='贯红'), (select 入校时间 from tb_student06 where 姓名='陈丹')) as 时间差距
计算全年工作日	select datename(dw,'2008-01-01') as '2008-01-01星期几', datename(dw,'2008-12-31') as '2008-12-31星期几', datediff(wk,'2008-01-01','2008-12-31') as 全年星期数, (datediff(day,'2008-01-01','2008-12-31') - (datediff(ww,'2008-01-01','2008-12-31'))*2) - (select 全年请假天数 from tb_AllYearPay where 员工编号 ='mr001') as 员工编号001全年工作日
查询当前图书出版日期与下一	select x.*,

次出版日期间隔	<pre> datediff(day,x.出版日期,x.下一次出版日期) 两次出版相 差的天数 from (select e.书号, e.书名, e.出版日期, (select min(d.出版日期) from tb_booksell06 d where d.出版日期 > e.出版日期) 下一次出版日期 from tb_booksell06 e) x </pre>
日期以年月日分开	<pre> select 书名, year((select 出版日期 from tb_booksell06 where 书号 = '1')) as 年份, month((select 出版日期 from tb_booksell06 where 书号 = '1'))as 月份, day((select 出版日期 from tb_booksell06 where 书号 = '1'))as 日期 from tb_booksell06 where 书号 = '1' </pre>
加三个月后的日期	<pre> select dateadd(mm,3,getdate()) as 'DAY NUMBER' </pre>
获取某月,本周和当前季度的	<pre> /*获取某月份第一天 */ </pre>

第一天	<pre> select dateADD(mm,datediff(mm,0,'2009-3-18'),0) as 'DAY_NUMBER' /*获取本周的第一天*/ select dateadd(wk,datediff(wk,0,getdate()), 0) as 'DAY_NUMBER' /*获取当前季度的第一天*/ select dateadd(qq,datediff(qq,0,getdate()),0) as 'DAY_NUMBER' </pre>
获取最后一天	<pre> /*获取去个月最后一天*/ select dateadd(ms,-3,dateadd(mm,datediff(mm,0,getdate()),0)) as 'DATE NUMBER上个月最后一天' /*获取去年最后一天*/ SELECT dateadd(ms,-3,DATEADD(yy, DATEDIFF(yy,0,getdate()), 0)) as 'DATE NUMBER去年最后的一天' /*获取本年的最后一天*/ SELECT dateadd(ms,-3,DATEADD(yy, DATEDIFF(yy,0,getdate())+1, 0)) as 'DATE NUMBER本年最后的一天' </pre>
长日期转化为短日期	<pre> select 书名, convert(char(10),出版日期,120) as 日期 from the book table </pre>

	from tb_booksellub
显示长日期中的小时,分,秒	SELECT CONVERT(nvarchar(10), getdate(), 8) as 当前时间
. 转化为-	update tb_student06 set 入校时间='20'+replace(入校时间,',','-') where 编号=3

功能	代码
用sum对工资进行汇总	SELECT SUM(salary) AS 工资,SUM(salary + bonus) AS 工资及奖金 FROM tb_treatment
avg 求平均成绩	SELECT AVG(语文) AS 语文 ,AVG(代数) AS 代数 ,AVG(几何) AS 几何 ,AVG(英语) AS 英语 FROM tb_stuAchievement
min 求销售和利润最低值	SELECT DISTINCT(编号),商品名称,销价 AS 销售额最少, 利润 AS 利润最少,门店名称 FROM tb_sell WHERE 销价 IN (SELECT MIN(销价)

```
FROM tb_sell  
)  
OR 利润 IN  
( SELECT MIN(利润)  
FROM tb_sell  
)
```

max 求销售业绩最高

```
SELECT tb1.销售员,tb1.销售额 AS 最高销售额  
from ( SELECT 销售员,SUM(销售额) AS 销售额  
FROM tb_Seller  
WHERE MONTH(日期)='12'  
GROUP BY 销售员  
)AS tb1  
  
WHERE tb1.销售额 = ( SELECT MAX(销售额) AS 最高销售额  
FROM  
( SELECT 销售员,SUM(销售额) AS 销售额  
FROM tb_Seller  
WHERE MONTH(日期)='12'  
GROUP BY 销售员  
)AS tb1  
)
```


count 求日销售的大于某值	SELECT COUNT(DISTINCT(书名)) AS 图书种类 FROM (SELECT 书名,日期,SUM(金额) AS 总金额 FROM tb_BookSell GROUP BY 书名,日期 HAVING SUM(金额) > 150) AS tb1
统计学生总成绩,排名	SELECT 学生编号,学生姓名, sum(高数+外语+计算机文化基础+马经+数据库管理+数据结构 +软件工程) AS 总成绩 FROM tb_StuMark GROUP BY 学生编号 ,学生姓名 ORDER BY 总成绩 DESC
coalesce 把NULL转化为0	SELECT 所属部门, AVG(COALESCE(工龄工资,0)) AS 工龄平均工资 FROM tb_emPay GROUP BY 所属部门
单价48和20销售总额	SELECT 书名,出版社,SUM(金额) AS 总计金额 FROM tb_BookSell

having 返回图书分组后, 汇总销售的大于某值	SELECT 书名,出版社,SUM(金额) AS 总计金额 FROM tb_BookSell GROUP BY 书名,出版社 HAVING SUM(金额)>500
grouping sets 组合多个分 组结果,按照书名,出版社, 销售的总额分组	use db_sql2008 go select 书名,出版社,SUM(金额) as 总计金额 from tb_BookSell group by GROUPING sets (书名,出版社)
over sum 根据订单编号 分区聚合数据	USE db_sql2005 GO SELECT 订单编号,商品编号,订货数量 ,SUM(订货数量) OVER(PARTITION BY 订单编号) AS '总计' ,COUNT(订货数量) OVER(PARTITION BY 订单编号) AS '计数' ,AVG(订货数量) OVER(PARTITION BY 订单编号) AS '平均' ,MIN(订货数量) OVER(PARTITION BY 订单编号) AS '最小' ,MAX(订货数量) OVER(PARTITION BY 订单编号) AS '最大' FROM tb_Sale WHERE 订单编号 IN('DD1305','DD1306') GO

compute 对销价汇总	SELECT 编号,商品名称,数量,销价,门店名称 FROM tb_Sell ORDER BY 门店名称 COMPUTE SUM(销价)
compute by 根据部分分组,对应汇总信息	SELECT * FROM tb_工资表 ORDER BY 所属部门 COMPUTE SUM(工资) BY 所属部门

功能	代码
select 平均工资,平均工资之差	select 姓名,工资,所属部门,(select avg(工资) from tb_工资表) as 平均工资, (工资-(select avg(工资) from tb_工资表)) as 与平均工资的差额 from tb_工资表
大于平均成绩	SELECT * FROM tb_stuAchievement WHERE 总分 > (SELECT AVG(总分) AS 平均总分 FROM tb_stuAchievement)
from 后加别名	SELECT top 1 * FROM (SELECT top 3 挂账单位.SUM(欠款金额) AS 金额

	<pre> FROM tb_gzmx GROUP BY 挂账单位 ORDER BY 金额 desc) tb1 ORDER BY 金额 </pre>
查询外语成绩大于某值的学生信息, 两个表	<pre> SELECT 学生姓名,所在学院,家庭住址 FROM tb_StuInfo WHERE 学生编号 IN (SELECT 学生编号 FROM tb_StuMark WHERE 外语 >80) </pre>
两个表,不在同类书1前100 的图书信息	<pre> select * from tb_BookInfo where 书号 not in (select 书号 from tb_BookOrder where 同类图书排名 < 100) </pre>
一班学生成绩大于二班所有学生成绩的学生信息	<pre> SELECT 学生编号,学生姓名,总分 FROM tb_stuAchievement WHERE 总分 > ALL(SELECT 总分 FROM tb_stuAchievement WHERE 班级 = '二班') AND 班级 = '一班' </pre>
三月销售大于当月平均水平的商品信息	<pre> SELECT 商品编号,所属类别,商品等级,商品名称,三月 FROM tb_商品销售 WHERE 三月 > ANY (</pre>

	SELECT AVG(三月) FROM tb_商品销售 GROUP BY 所属类别,商品等级)
部门内职员工资最高的信息, IN,两个表	SELECT b.姓名,b.部门,a.salary,b.业绩 FROM tb_treatment AS a,tb_employee AS b WHERE a.salary in (select max(salary) from tb_treatment where a.name=b.姓名 group by dept)
销量大于某值的图书信息,两个表	SELECT * FROM tb_BookInfo AS a WHERE exists (SELECT * FROM tb_BookOrder AS b WHERE a.书号 = b.书号 AND 销售数量 > 400)
采购量不小于某数值的信息,not exists	SELECT * FROM tb_商品销售 AS a WHERE NOT EXISTS (SELECT * FROM tb_商品采购 AS b WHERE a.商品编号 = b.商品编号 AND 采购量 < 2000)
Having ,分组查询,大于平均年龄	SELECT 人员姓名,年龄 FROM tb_aMRem GROUP BY 人员姓名,年龄 HAVING 年龄 >

	(SELECT AVG(年龄) FROM tb_aMRem)
指点学历的部分经理当月的工资情况	SELECT * FROM tb_aMRPay WHERE 工资月份=10 AND 人员姓名 IN(SELECT 负责人 FROM tb_aMRDe WHERE 负责人 IN(SELECT 人员姓名 FROM tb_aMRem WHERE 学历='本科')) ORDER BY 人员编号
返回成绩小于制定的多个成绩任意一个的学生信息	SELECT * FROM tb_StuMark WHERE 软件工程 < SOME (SELECT 软件工程 FROM tb_StuMark WHERE 学生编号 IN(SELECT 学生编号 FROM tb_StuMark WHERE 学生编号 between 200941004 and 200941005))
查询更新数据 update	UPDATE tb_药品销售 SET 药品名称=(SELECT 药品名称 FROM tb_药品登记 WHERE tb_药品销售.药品编号=tb_药品登记.药品编号)
insert into 添加数据	INSERT INTO tb_药品销售 (药品编号, 药品名称)

insert into 添加数据	<pre> INSERT INTO tb_药品销售 (药品编号,药品名称) (SELECT 药品编号,药品名称 FROM tb_药品登记 WHERE 药品编号 NOT IN (SELECT 药品编号 FROM tb_药品销售)) </pre>
查询删除数据 delete where	<pre> DELETE tb_药品销售 WHERE 药品编号 IN (SELECT 药品编号 FROM tb_药品登记 WHERE 生产厂家='长春一通') </pre>

功能	代码
cross join 实现两个表 联接	<pre> SELECT * FROM tb_学系 CROSS JOIN tb_教研室 </pre>
where 设置联接条件	<pre> SELECT tb_student.*,tb_class.班主任编号,tb_teacher.教师姓名,tb_teacher.授课 名称 FROM tb_student,tb_class,tb_teacher WHERE tb_student.班级=tb_class.班级 </pre>

	AND tb_class.班主任编号=tb_teacher.教师编号
内联接,返回指点月份的工资和所在部分信息	SELECT a.部门编号,a.部门名称,a.负责人, b.人员编号,b.人员姓名,b.实发合计,b.工资年,b.工资月份 FROM tb_aMRDe AS a INNER JOIN tb_aMRPay AS b ON a.部门名称=b.部门名称 WHERE 工资月份='5'
自身:查询与 张二的 基本工资相同的信息	SELECT t1.员工编号,t1.姓名,t1.工龄工资,t1.基本工资,t1.所属部门 FROM tb_emPay t1,tb_emPay t2 WHERE t1.基本工资=t2.基本工资 AND t2.姓名='张二'
统计已出库的商品进货价格	SELECT fullname AS 商品名称, tsum1 AS 进货金额 FROM (SELECT a.tradecode, a.fullname, a.averageprice, b.qty1, b.tsum1 FROM tb_stock a INNER JOIN (SELECT SUM(qty) AS qty1, SUM(tsum) AS tsum1, fullname FROM tb_warehouse_detailed GROUP BY fullname) b ON a.fullname = b.fullname WHERE (a.price > 0)) tb1
left outer join 左表为主的所有匹配数据	SELECT a.部门编号,a.部门名称,a.负责人, b.人员编号 b.人员姓名 b.部门名称

<p>SQL 语句</p>	<p>b.学历,b.技术职称</p> <p>FROM tb_aMRDe a LEFT OUTER JOIN tb_aMRem b</p> <p>ON a.部门名称=b.部门名称</p>
<p>以学系表为主体,列出每个学系的基本相关信息.</p> <p>返回右表为主的所有匹配数据</p>	<p>SELECT a.教研室编号,a.教研室名称,b.*</p> <p>FROM tb_教研室_TM a RIGHT OUTER JOIN tb_学系 b</p> <p>ON a.学系编号=b.学系编号</p>
<p>返回所有两表匹配的所有数据</p>	<p>SELECT a.*,b.教研室编号,b.教研室名称</p> <p>FROM tb_学系 AS a FULL OUTER JOIN tb_教研室_TM AS b</p> <p>ON a.学系编号=b.学系编号</p>
<p>union 汇集多个表的信息,期中和期末成绩</p>	<p>SELECT * FROM tb_期中考试成绩表</p> <p>WHERE 总分 > 275</p> <p>UNION</p> <p>SELECT * FROM tb_期末考试成绩表</p> <p>WHERE 总分 > 275</p>
<p>对多个表进行排序,心理学60分,演讲与口才70分及格</p>	<p>SELECT *</p> <p>FROM tb_Uman WHERE 学生编号 IN</p> <p>(SELECT 学生编号 FROM tb_Ucourse_score WHERE 选修课编号 IN</p>

	<pre> (SELECT 选修课编号 FROM tb_Ucourse WHERE 选修课='心理学') AND 选修成绩 >= 60) UNION SELECT * FROM tb_Uman WHERE 学生编号 IN (SELECT 学生编号 FROM tb_Ucourse_score WHERE 选修课编号 IN (SELECT 选修课编号 FROM tb_Ucourse WHERE 选修课='演讲与口才') AND 选修成绩 >= 70) ORDER BY 毕业时间,学生编号 </pre>
查:选修了心理学而没选演讲与口才的学生名单	<pre> use db_sql2005 go SELECT * FROM tb_学系 INTERSECT SELECT * FROM tb_学系_TM use db_sql2005 go SELECT * </pre>

	<pre> FROM tb_Uman WHERE 学生编号 IN (SELECT 学生编号 FROM tb_Ucourse_score WHERE 选修课编号 IN (SELECT 选修课编号 FROM tb_Ucourse WHERE 选修课='心理学')) EXCEPT SELECT * FROM tb_Uman WHERE 学生编号 IN (SELECT 学生编号 FROM tb_Ucourse_score WHERE 选修课编号 IN (SELECT 选修课编号 FROM tb_Ucourse WHERE 选修课='演讲与口才')) ORDER BY 学生编号 </pre>
intersect 量个结果的交集	<pre> use db_sql2005 go SELECT * FROM tb_学系 INTERSECT SELECT * FROM tb_学系_TM </pre>

功能	代码
返回备注信息,<10返回不足,>100返回超标	<pre> SELECT 书名,作者,现存数量, 备注= CASE WHEN 现存数量<10 THEN '库存不足' </pre>

	<pre> WHEN 现存数量>100 THEN '库存超标' ELSE '--' END FROM tb_BookStore </pre>
case 修改,男120,女100	<pre> UPDATE tb_emp SET 伙食补助= CASE WHEN 性别='男' THEN 120 WHEN 性别='女' THEN 100 END </pre>
各个部门的销售汇总	<pre> SELECT 员工姓名, SUM(CASE 所在部门 WHEN '食品部' THEN 销售业绩 ELSE NULL END) AS 食品部业绩, SUM(CASE 所在部门 WHEN '家电部' THEN 销售业绩 ELSE NULL END) AS 家电部业绩 FROM tb_销售 GROUP BY 员工姓名 </pre>
行列转换	<pre> use db_sql2000 GO if exists(select * from INFORMATION_SCHEMA.TABLES where table_name = 'tb_StuScore10') drop table tb_StuScore10 </pre>


```
GO
--创建学生成绩信息表tb_StuScore10
create table tb_StuScore10
(
    学生姓名 varchar(20),
    学生学科 varchar(20),
    成绩 float
)
GO
--向学生成绩信息表tb_StuScore10中插入数据
insert into tb_StuScore10 values('张红','语文','80')
insert into tb_StuScore10 values('张红','数学','90')
insert into tb_StuScore10 values('张红','外语','85')
insert into tb_StuScore10 values('李燕','语文','80')
insert into tb_StuScore10 values('李燕','数学','85')
insert into tb_StuScore10 values('李燕','外语','92')
GO

--在学生成绩信息表中tb_StuScore10实现行列互换
select 学生姓名,
sum(case 学生学科 when '语文' then 成绩 end)as 语文
sum(case 学生学科 when '外语' then 成绩 end)as 外语
```

```
,sum(case 学生学科 when '外语' then 成绩 end )AS 外语
,sum(case 学生学科 when '数学' then 成绩 end )AS 数学
from tb_StuScore10 group by 学生姓名
GO
```

```
-----

USE db_sql2008
GO
select 学生姓名,a.语文,
a.数学,a.外语
from tb_StuScore10
pivot(sum(成绩)for 学生学科
in([语文],[数学],[外语]))as a
GO
```

pivot 交叉统计,各月份销售
数量

```
USE db_sql2005
GO
select 商品名称,a.[9] as [九月],
a.[10] as [十月],a.[11] as [十一月]
,a.[12] as [十二月]
from tb_privot
pivot(sum(销售数量)for 月份
```

```

pivot(sum(销售数量 for 月份
in([9],[10],[11],[12]))as a
GO

```

反向交叉

```

use db_sql2005
GO
select * from
tb_Sell
unpivot(销售数量 for 月份 in([九月],[十月],[十一月],[十二月] )) as a

```

动态交叉表

```

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Corss]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Corss]
GO
CREATE procedure Corss
@strTabName as varchar(50) = 'tb_销售',
@strCol as varchar(50) = '所在部门',
@strGroup as varchar(50) = '员工姓名',--分组字段
@strNumber as varchar(50) = '销售业绩', --被统计的字段
@strSum as varchar(10) = 'Sum' --运算方式
AS
DECLARE @strSql as varchar(1000), @strTmpCol as varchar(100)
EXECUTE ('DECLARE corss_cursor CURSOR FOR SELECT DISTINCT ' + @strCol + ' from
' + @strTabName + ' for read only ') --生成游标
begin
SET nocount ON
SET @strsql ='select ' + @strGroup + ', ' + @strSum + '(' + @strNumber + ') AS [' +
@strNumber + ']' --查询的前半段

```



```

OPEN corss_cursor
while (0=0)
BEGIN
FETCH NEXT FROM corss_cursor --遍历游标，将列头信息放入变量@strTmpCol
INTO @strTmpCol
if (@@fetch_status<>0) break
SET @strsql = @strsql + ', ' + @strSum + '(CASE ' + @strCol + ' WHEN ''' +
@strTmpCol + ''' THEN ' + @strNumber + ' ELSE Null END) AS [' + @strTmpCol + ']' --
构造查询
END
SET @strsql = @strsql + ' from ' + @strTabname + ' group by ' + @strGroup --查询结
尾
EXECUTE(@strsql) --执行
IF @@error <>0 RETURN @@error --如果出错，返回错误代码
CLOSE corss_cursor
DEALLOCATE corss_cursor RETURN 0 --释放游标，返回0表示成功
end

```

CTE 递归,
创建标号,插入连续数

```

use db_sql2005
GO
with x(编号)
as(select 1 from tb_series
union all
select 编号+1 from x where 编号+1<=10)
select * from x

```

功能	代码
插入一条信息insert	insert into 顾客表 values('KH005','刘娜','吉林省长春市','东胜大街89号','130051','4912668','4912668')
union all 批量添加	insert tb_cmd select 1,1,'添加' union all select 1,2,'删除' union all select 2,1,'添加' union all select 2,3,'更改' union all select 2,5,'查找' union all select 3,4,'查看'
insert 插入多行	use db_sql2008 go insert into dbo.tb_顾客表 values ('KH005','刘丽','吉林省吉林市','吉林大街号','131000','562893','562789'), ('KH006','张雨','吉林省四平市','延安大街号','137000','756893','756852')
创建表并插入数据	create table tb_销售信息表 (编号 char(20),产品名称 varchar(50),销售单价 money) GO insert into tb_销售信息表 values('XS001','办公用品',60) GO
update 修改单条语句	update tb 顾客表

	<pre> set 顾客姓名='周小丽',所在城市='吉林省吉林市' where 顾客编号='KH003' </pre>
批量修改数据	<pre> update tb_laborage set 基本工资=2000 where 员工编号>1002 and 员工编号<1005 </pre>
修改指定日期的数据	<pre> update tb_雇员表 set 基本工资=2000 where 雇用日期='2008-12-16' </pre>
删除数据,'李一'	<pre> delete from tb_雇员表 where 雇员名称='李一' </pre>
删除空记录 null	<pre> delete from tb_xsd where 简称 is null </pre>
output 删除数据并返回结果	<pre> use db_sql2008 go DECLARE @MyTableVar table (顾客编号 nvarchar(50)NOT NULL, 顾客姓名 nvarchar(50)NOT NULL, 电话 nvarchar(50)NOT NULL); delete tb_顾客表 output deleted.顾客编号, deleted.顾客姓名, deleted.电话 </pre>


```
into @MyTableVar  
where 顾客编号 in ('KH005','KH006')  
select * from @MyTableVar
```

使用事务,删除数据

```
begin tran Update_data  
    update tb_顾客表  
    set 邮编='130051'  
    where 顾客编号='KH003'  
commit tran Update_data
```