1. **PostGreSQL基础知识：**

**整数类型：**

smallint //小范围整数，取值范围：-32768~32767；

int(integer) //普通大小整数-2147483648~2147483647；

**任意精度浮点数类型：**

real //6位十进制数字精度；

numeric(m,n) //任意精度类型，m为数值最大长度，n为保留的小数，小数点不占位数；

**日期与时间类型：**

time //只用于一日内时间，如 10:05:05；

date //只用于日期，如 1987-04-04；

timestamp //日期和时间，如 1987-04-04 10:05:05；

**字符串类型：**

char(n)/character(n) //固定长度n字符串，不足补空白；

varchar(n)/character varying(n) //变长字符串，有长度限制；

text //变长字符串，无长度限制；

**常用的数值函数：**

avg() //返回某列的平均值；

count() //返回某列的行数；

max() //返回某列的最大值；

例：select max(e\_salary) from employee;//查询雇员工资最大值；

min() //返回某列的最小值

例：select min(e\_salary) from employee;//查询雇员工资最小值；

sum() //返回某列的值之和；

**常用的字符串函数：**

length(s) //计算字符串长度 ；

例：select e\_name,length(e\_name) from employee;//查询雇员名称及其名称长度；

concat(s1,s2,...) //字符串合并函数 ；

例：select e\_no, e\_name, e\_hireDate, concat(c\_no, e\_name, e\_hireDate) from employee;

ltrim(s)/rtrim(s)/trim(s) //删除字符串空格函数；

replace(s,s1,s2) //字符串替换函数；

substring(s,n,len) //获取子串函数；

**创建函数的语法：**

creat function //声明创建函数；

add(integer,integer) //定义函数名称，参数类型；

return integer //定义函数返回值；

as 'select $1 + $2;' //定义函数体；

language SQL //通过SQL语言来创建函数；

returns NULL ON NULL INPUT; //定义参数为NULL时处理情况；

**操作数据库：**

1、创建数据库对象：create database 数据库名;

create database db\_jikexueyuan #数据库名；

with ENCODING='utf-8' #编码为utf-8；

OWNER=user #所有者；

CONNECTION LIMIT=10; #限制连接数为10；

2、修改数据库对象（其中的一些参数）：alter database ...

(1)alter database db\_jikexueyuan

rename to db\_jikexueyuan1; #修改数据库名称；

(2)alter database db\_jikexueyuan1

with CONNECTION LIMIT = 100; #修改数据库连接数限制；

或 alter database db\_jikexueyuan1 CONNECTION LIMIT 100;也相同；

表级操作：

(3)alter table student rename to student1;

#修改表名。修改表名student为student1；

(4)alter table student1 rename id to bh;

#修改字段名。修改表student1的字段id为bh；

(5)alter table student1 alter column name type varchar(40);

#修改字段数据类型。修改name字段的数据类型为VARCHAR(40)；

(6)alter table student1 drop column birthday;

#删除字段。删除字段birthday，同时会删除对应的信息；

(7)alter table student1 add column address varchar(200);

#添加字段。添加address字段，类型为VARCHAR(200)；

3、删除数据库对象：drop database 数据库名;

表级操作：

(1)drop table student1; #删除数据表

(2)drop table if exists student1; #只有数据表存在时才执行删除操作

4、操作数据表对象：create table 表名 ...

create table student (

id int,

name varchar(30),

birthday date,

score numeric(5,2)

);

5、更新数据库表：

(1)指定条件更新数据：update 表名 set 要更新的字段名 = 要更新的字段内容 where 要更新的哪一条数据;

update student set name = 'lisi' where id = 2; //更新id=2的学生姓名为“lisi”

(2) 批量更新数据

update student set score = 0; //将student表中的学分全更新为0

insert into student (id, name, birthday) values (2, 'zhangsan', '1990-01-02');

//没插入的部分为null

insert into student (id, name, birthday) values

(3, 'zhangsan1', '1990-02-01'),

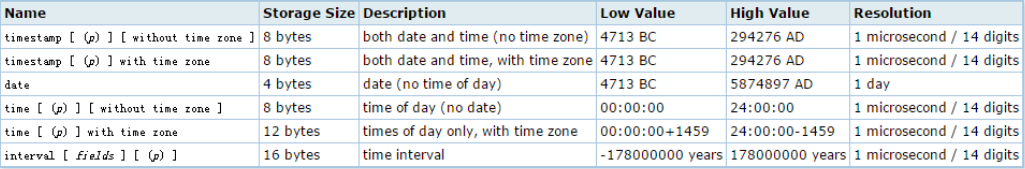
(4, 'zhangsan2', '1990-02-02'), //批量插入数据

1. **SQL server与PostgreSQL对比：**

1、SQLserver: date:就是日期类型，与PostgreSQL里面的date类型相同；

2、SQLserver: datetime:就是日期类型(加上时间)，与PostgreSQL里面的datestamp类型相同都是8个字节，但是datestamp做了在时区做了详细区分，而且PostgreSQL里面没有datetime类型；

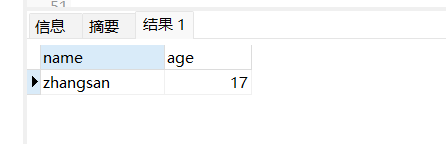
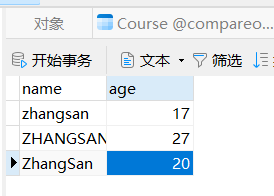
3、PostgreSQL时间类型具体如下，PostgreSQL具体参考：[PostgreSQL 日期/时间类型 (yiibai.com)](https://www.yiibai.com/manual/postgresql/datatype-datetime.html)；



4、postgresql不支持nvarchar类型 ……

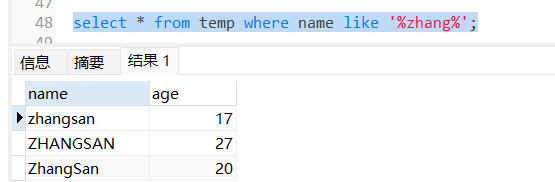
5、表的数据：

相同函数：select \* from temp where name like '%zhang%';



数据库表 1

图表1 PostgreSQL查询结果



图表2 SQLserver查询结果

结论：

PostgreSQL在模糊查询的时候时会区分大小写；而在下面的SQLserver中则不区分大小写：

当PostgreSQL和SQL Server的模糊匹配like是不一样的，PostgreSQL的like是区分大小写的，SQL Server不区分。如果想让PostgreSQL的like也不区分大小写的话，可以使用ilike。或者使用lower或者upper都转换成小写或者大写再模糊匹配，这种方式的SQL两种数据库都兼容。

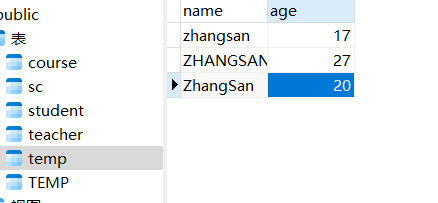
select \* from t\_user where name ilike '%zhang%';

select \* from t\_user where upper(name) like upper('%zhang%');

select \* from t\_user where lower(name) like lower('%zhang%');

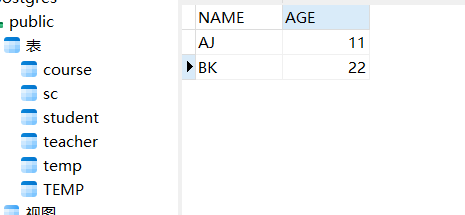
6、 PostgreSQL表名与字段名要用双引号 否则将会变成小写字符.

执行PostgreSQL*语句:*select \* from TEMP……



图表3 PostgreSQL执行结果

再次执行PostgreSQL*语句：*select \* from " TEMP" ……



图表 4 PostgreSQL加引号的执行结果

结论：

PostgreSQL在执行操作数据库时，表名和字段名的时候要加上引号，不加则不能区分大小写。

所以解决办法：

表 名： SQLSERVER-> DCS\_RISK == POSTGRESQL->”DCS\_RISK”;

字段名： SQLSERVER-> STATUS\_NAME == POSTGRESQL->”STATUS\_NAME”

备注：（sqlserver 表中 表名与字段名上添加“双引号”不会报错 并且没有“字符大小写”控制）。

7、Sqlserver中ISNULL函数在postgresql 中是不能使用的!

解决办法：ISNULL(IS\_CLOSED,0)=0 == coalesce("IS\_CLOSED" IS NULL OR "IS\_CLOSED"=0)

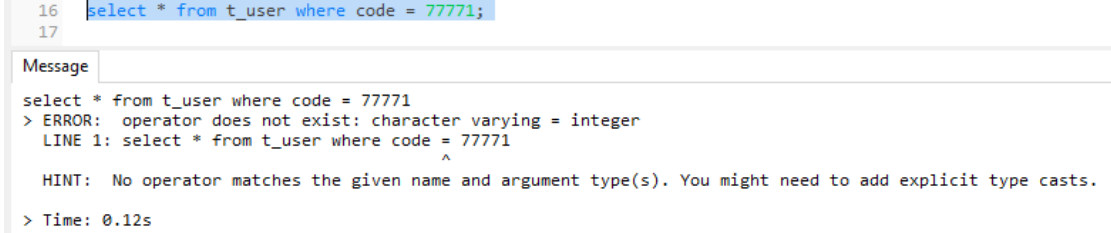
8、Sqlserver 中 字符类型是数值型，按空字符串查询 可以使用 在 postgresql中 字符类型是数值型是不可使用的!

例如：pgsql中

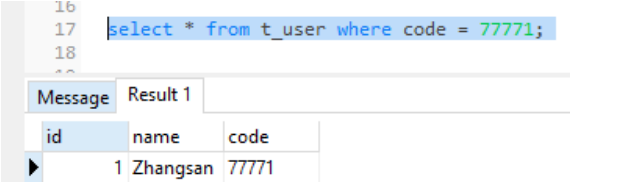
TEMP表中的 AGE 它是int 类型 使用 AGE=”” 作为条件会报错!

9、PostgreSQL在做条件查询的时候是强类型校验的，但是SQL Server是弱类型。也就是说 用数值类型值 查询字符类型值 是报错的 需要用 单引号包起来呀 反之 用字符串数值 查询 数组类型值 是不报错的

执行语句：select \* from t\_user where code = 77771;



图表 5 PostgreSQL结果

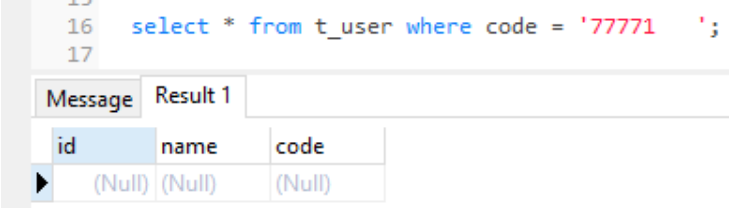


图表 6 SQLserver结果

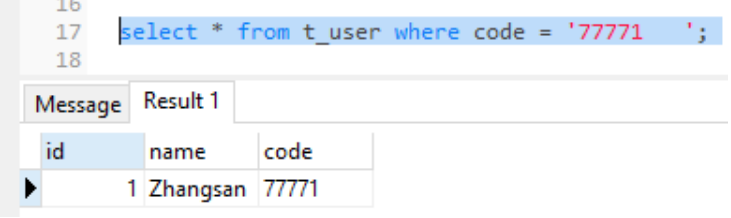
结论：

//code 是字符类型 77771 是数据值型 执行就会报错 所以要用 ‘77771’ 单引号包起来查询

10、SQL Server的查询如果末尾有空白的话，SQL Server会忽略但是PostgreSQL不会。

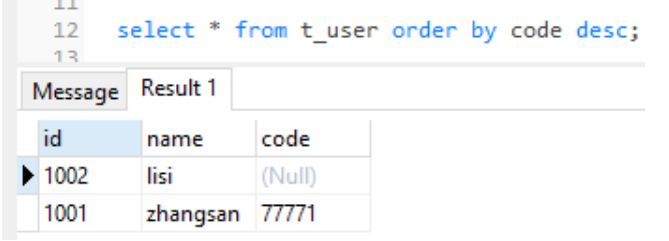


图表 7 PostgreSQl执行结果

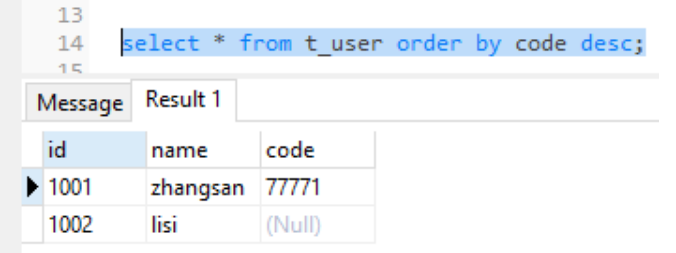


图表 8 SQlserver执行结果

11、PostgreSQL和SQL Server的默认order by行为是不一致的。order by的字段如果是null，PostgreSQL会将其放在前面，SQLServer则将其放在后面。



图表 9 PostgreSQL-order by



图表 10 SQLserver-order by

解决办法：

某些情况下如果要求数据顺序在两个数据库中要一致的话，可以在PostgreSQL的查询SQL中增加nulls last来让null数据滞后。

例：select \* from t\_user order by code desc nulls last;

也可以使用case when来统一SQL：

例：ORDER BY (case when xxx is null then '' else xxx end) DESC;

12、字符串拼接：

SQL Server使用" + “号来拼接字符串，并且在2012版本之前不支持concat函数。PostgreSQL使用” || "来拼接字符串，同时支持concat函数。

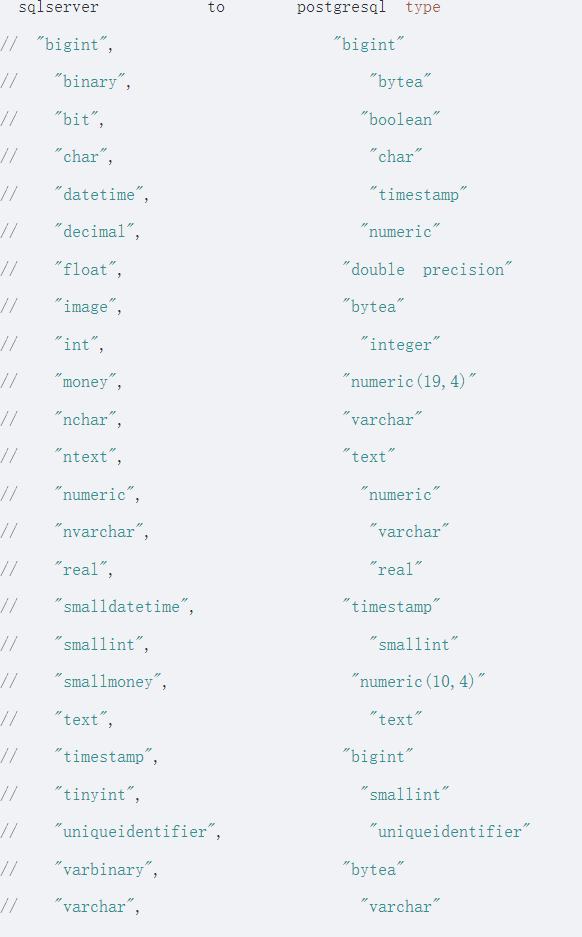
13、查询表是否存在：

//SQL Server

select count(name) from sys.tables where type='u' and name='t\_user';

//PostgreSQL

select count(table\_name) from information\_schema.tables where table\_name='t\_user';



图表 11 SqlServer与Postgresql数据库字段类型对照表