

---

Subject: PostgreSQL issues

Posted by [zsolt](#) on Wed, 23 May 2007 20:34:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I have checked current version and found some strange things. In PostgreSQLSchema.h there are some buggy lines:

```
#define SERIAL(x)          COLUMN("integer autoincrement", int64, x, 0, 0) //int is not enough,
as it is unsigned
```

```
#define SERIAL_ARRAY(x, items)  COLUMN_ARRAY("integer autoincrement", int64, x, 0, 0,
items)
```

```
#define SERIAL_(x)          COLUMN_("integer autoincrement", int64, x, 0, 0)
```

```
#define SERIAL_ARRAY_(x, items) COLUMN_ARRAY_("integer autoincrement", int64, x, 0, 0,
items)
```

```
#define BIGSERIAL(x)          COLUMN("integer autoincrement", int64, x, 0, 0)
```

```
#define BIGSERIAL_ARRAY(x, items) COLUMN_ARRAY("integer autoincrement", int64, x, 0, 0,
items)
```

```
#define BIGSERIAL_(x)          COLUMN_("integer autoincrement", int64, x, 0, 0)
```

```
#define BIGSERIAL_ARRAY_(x, items) COLUMN_ARRAY_("integer autoincrement", int64, x, 0,
0, items)
```

The above lines would be useful in SQLite (I wanted to propose it), but not in PG. The correct lines would be:

```
#define SERIAL(x)          COLUMN("serial", int64, x, 0, 0) //int is not enough, as it is unsigned
```

```
#define SERIAL_ARRAY(x, items)  COLUMN_ARRAY("serial", int64, x, 0, 0, items)
```

```
#define SERIAL_(x)          COLUMN_("serial", int64, x, 0, 0)
```

```
#define SERIAL_ARRAY_(x, items) COLUMN_ARRAY_("serial", int64, x, 0, 0, items)
```

```
#define BIGSERIAL(x)          COLUMN("bigserial", int64, x, 0, 0)
```

```
#define BIGSERIAL_ARRAY(x, items) COLUMN_ARRAY("bigserial", int64, x, 0, 0, items)
```

```
#define BIGSERIAL_(x)          COLUMN_("bigserial", int64, x, 0, 0)
```

```
#define BIGSERIAL_ARRAY_(x, items) COLUMN_ARRAY_("bigserial", int64, x, 0, 0, items)
```

I would propose some other lines to SQLite for compatibility with PostgreSQL:

```
#define CLOB(x)          COLUMN("text", String, x, 0, 0)
```

```
#define CLOB_(x)          COLUMN_("text", String, x, 0, 0)
```

---

Subject: Re: PostgreSQL issues

Posted by [zsolt](#) on Wed, 23 May 2007 20:47:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

BTW in my dev environment, I use SERIAL differently, because I wanted a common .sch file for my SQLite and PostgreSQL databases with the same behaviour.

So I use these lines for SQLite:

```
#define SERIAL(x)          COLUMN("integer primary key autoincrement", int64, x, 0, 0) //int is
```

not enough, as it is unsigned

```
#define SERIAL_ARRAY(x, items)    COLUMN_ARRAY("integer primary key autoincrement",
int64, x, 0, 0, items)
#define SERIAL_(x)                COLUMN_("integer primary key autoincrement", int64, x, 0, 0)
#define SERIAL_ARRAY_(x, items)   COLUMN_ARRAY_("integer primary key autoincrement",
int64, x, 0, 0, items)

#define BIGSERIAL(x)              COLUMN("integer primary key autoincrement", int64, x, 0, 0)
#define BIGSERIAL_ARRAY(x, items) COLUMN_ARRAY("integer primary key autoincrement",
int64, x, 0, 0, items)
#define BIGSERIAL_(x)            COLUMN_("integer primary key autoincrement", int64, x, 0, 0)
#define BIGSERIAL_ARRAY_(x, items) COLUMN_ARRAY_("integer primary key autoincrement",
int64, x, 0, 0, items)
```

and these lines for PostgreSQL:

```
#define SERIAL(x)                COLUMN("serial primary key", int64, x, 0, 0) //int is not enough, as it
is unsigned
#define SERIAL_ARRAY(x, items)   COLUMN_ARRAY("serial primary key", int64, x, 0, 0, items)
#define SERIAL_(x)              COLUMN_("serial primary key", int64, x, 0, 0)
#define SERIAL_ARRAY_(x, items)  COLUMN_ARRAY_("serial primary key", int64, x, 0, 0,
items)

#define BIGSERIAL(x)            COLUMN("bigserial primary key", int64, x, 0, 0)
#define BIGSERIAL_ARRAY(x, items) COLUMN_ARRAY("bigserial primary key", int64, x, 0, 0,
items)
#define BIGSERIAL_(x)          COLUMN_("bigserial primary key", int64, x, 0, 0)
#define BIGSERIAL_ARRAY_(x, items) COLUMN_ARRAY_("bigserial primary key", int64, x, 0, 0,
items)
```

So SERIAL in my terminology means a field type as it is incrementing automatically, serves as a primary key, and records can never have the same (old) id after deleting rows.

---

Subject: Re: PostgreSQL issues  
Posted by [mirek](#) on Thu, 24 May 2007 21:30:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

At the moment I am not using either...

Can anybody comment about this? Which patches should we apply?

Mirek

---

Subject: Re: PostgreSQL issues

Posted by [unodgs](#) on Fri, 25 May 2007 07:08:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

What is the problem with explicitly defining column as PRIMARY\_KEY?

SERIAL (ID) PRIMARY\_KEY

I think primary\_key attribute should be removed from sqlite schema to preserve consistency instead of adding it to postgresql.

Besides PostgreSQL documentation says:

Note: Prior to PostgreSQL 7.3, serial implied UNIQUE. This is no longer automatic. If you wish a serial column to be in a unique constraint or a primary key, it must now be specified, same as with any other data type.

---

---

Subject: Re: PostgreSQL issues

Posted by [zsolt](#) on Fri, 25 May 2007 17:43:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

The problem is, that I need the same .sch file for PostgreSQL and SQLite. But "ID\_FIELD integer autoincrement primary key" is not allowed in SQLite. The correct column definition is "ID\_FIELD integer primary key autoincrement".

---

---

Subject: Re: PostgreSQL issues

Posted by [mirek](#) on Fri, 13 Jul 2007 11:49:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In any case, trying SQL\_Postgresql, SERIAL was the first thing I had to fix...

Mirek

---

---

Subject: Re: PostgreSQL issues

Posted by [zsolt](#) on Fri, 13 Jul 2007 20:30:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

What was the problem? (Currently I'm unable to connect to the UVS server.)

---

---

Subject: Re: PostgreSQL issues

Posted by [mirek](#) on Sun, 15 Jul 2007 09:45:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Well, this was the first time I tried postgresql... With the current version, postgresql reported error on executing the schema - the exact one described in this thread. So I have changed SERIAL

---

definitions as suggested here and everything is OK now...

Mirek

---