

Interactive - 31 - only one row of data (uk with no sequence)

With system configuration tables it is often useful to only have 1 row of data. Each configuration item is a distinct column in the table.

This can be done with a primary key and a constraint. The constraint limits the values on the primary key and results in an error if a attempt to insert a 2nd row occurs.

In this example we just have `applicaiton_name` as the single configuration item.

```
DROP TABLE if exists ct_config ;
```

```
CREATE TABLE ct_config (  
    config_id serial primary key check ( config_id = 1 ),  
    application_name text  
);
```

The first insert will work. Then 2nd will not.

```
INSERT INTO ct_config ( application_name ) values ( '4820 class' );
```

and 2nd insert

```
insert into ct_config ( application_name ) values ( '4010 blockchain class' );
```

on the 2nd insert you should get an error similar to:

```
psql:hw31_3.sql:3: ERROR:  new row for relation "ct_config" violates check constraint "ct_c  
DETAIL:  Failing row contains (2, 4010 blockchain class).
```

We can prevent delete of the row with a trigger.

```
CREATE OR REPLACE FUNCTION ct_config_prevent_delete()  
RETURNS trigger AS $$  
BEGIN  
    IF OLD.config_id = 1 THEN  
        RAISE EXCEPTION 'cannot delete configuration row';  
    END IF;  
END;  
$$  
LANGUAGE plpgsql;  
  
CREATE TRIGGER ct_config_prevent_delete  
BEFORE DELETE ON ct_config  
FOR EACH ROW EXECUTE PROCEDURE ct_config_prevent_delete();
```

When you try to delete:

```
delete from ct_config;
```

You should get an error similar to:

```
ERROR:  cannot delete configuration row
```

```
CONTEXT:  PL/pgSQL function ct_config_prevent_delete() line 4 at RAISE
```

Tags: “single row”, “raise exception”, “trigger”, “delete trigger”

Validate: SQL-Select, “select ‘PASS’ as x”