

pgAdmin 4

127.0.0.1:49188/browser/

History

Search history

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

Browsers

Columns (8)

c_id

e_type

e_name

length_in

start_date

end_date

Constraints

Indexes

RLS Policies

Rules

Triggers

students

teams

Columns (3)

team_id

l_name

t_strength

Constraints

Indexes

RLS Policies

Rules

Triggers

winners

Columns (4)

co_id

team_id

c_id

position

Constraints

Indexes

RLS Policies

Rules

Triggers

Trigger Functions (2)

tr_comp()

tr_student()

Types

Views

public

postgres

SQL

Statistics

Properties

Dependencies

Dependents

Dashboard

201801030_db/postgres@PostgreSQL 10 *

Query Editor

Query History

Scratch Pad

```
1 CREATE OR REPLACE FUNCTION "cf_db".Tr_i_student()
2 RETURNS trigger
3 LANGUAGE 'plpgsql'
4 AS $$
5
6 DECLARE
7
8 BEGIN
9
10 FOR i in (select * from "cf_db"."students")
11 LOOP
12 IF (NEW.s_id=i.s_id) then
13 raise notice 'Student with same ID exists';
14 RETURN NULL;
15 END IF;
16 END LOOP;
17
18 RETURN NEW;
19 END
20
21
22
23
24
25 CREATE TRIGGER "Trigger_ins"
26 BEFORE INSERT ON "cf_db".students
27 FOR EACH ROW EXECUTE PROCEDURE "cf_db".Tr_i_student();
28
29
30 INSERT INTO "cf_db".students(s_id) VALUES (9);
31
32
```

Data Output

Explain

Messages

Notifications

NOTICE: Student with same ID exists
INSERT 0 0

Query returned successfully in 79 msec.

pgAdmin 4

127.0.0.1:49188/browser/

History

Search history

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

Browsers

Servers (1)

PostgreSQL 10

Databases (2)

201801030_db

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Schemas (2)

cf_db

Collations

Domain

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions (1)

winners.comp()

Materialized Views

Sequences

Tables (6)

catalog

Columns (4)

c_id

c_name

c_address

c_ph

Constraints

Indexes

RLS Policies

Rules

Triggers

competition

Columns (6)

co_id

e_id

win_price

date

time

Constraints

Indexes

SQL

Statistics

Properties

Dependencies

Dependents

Dashboard

201801030_db/postgres@PostgreSQL 10 *

Query Editor

Query History

Scratch Pad

```
1 CREATE OR REPLACE FUNCTION "cf_db".Tr_i_comp()
2 RETURNS trigger
3 LANGUAGE 'plpgsql'
4 AS $$
5
6 DECLARE
7 curr_e record;
8
9 BEGIN
10
11 FOR curr_e in (select * from "cf_db"."events")
12 LOOP
13 IF (NEW.e_id=curr_e.e_id) then
14 RETURN NEW;
15 END IF;
16 END LOOP;
17
18 raise notice 'No event exists for the given ID, creating link not possible';
19 RETURN NULL;
20 END
21
22
23
24
25
26 CREATE TRIGGER "Trigger_insert"
27 BEFORE INSERT
28 ON "cf_db".competition
29 FOR EACH ROW EXECUTE PROCEDURE "cf_db".Tr_i_comp();
30
31
32 INSERT INTO "cf_db".competition(co_id,e_id) VALUES (32,8);
33
```

Data Output

Explain

Messages

Notifications

NOTICE: No event exists for the given ID, creating link not possible
INSERT 0 0

Query returned successfully in 63 msec.

pgAdmin 4

127.0.0.1:49188/browser/

History

Search history

Views

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

pgAdmin

File

Object

Tools

Help

SQL

Statistics

Properties

Dependencies

Dependents

Dashboard

201801030_db/postgres@PostgreSQL 10 *

Browser

competition

Columns (6)

co_id

e_id

co_name

win_price

date

time

Constraints

Indexes

RLS Policies

Rules

Triggers

events

Columns (6)

e_id

e_type

e_name

length_in_u

start_date

end_date

Constraints

Indexes

RLS Policies

Rules

Triggers

students

teams

Columns (3)

team_id

t_name

t_strength

Constraints

Indexes

RLS Policies

Rules

Triggers

winner

Columns (4)

co_id

team_id

e_id

position

Constraints

Query Editor

Query History

Scratch Pad

```
1 create or replace function "cf_db"."winners_competition"(s character varying(256))
2 returns table(b character varying(256))
3 language 'plpgsql'
4 as $$
5
6 declare
7     r RECORD;
8
9 begin
10     for i in(select competition.co_name from CF_DB.competition join(select teams.t_name,winners.co_id from CF_DB.winners join
11         CF_DB_teams on winners.team_id=teams.team_id where winners.position=3 and teams.t_names) as R on
12         competition.co_id=R.co_id)
13     loop b:=(i.co_name);
14     end loop;
15     return next;
16 end loop;
17
18 end;
19 $$
20
21 select "cf_db"."winners_competition"('ITB A');
```

Data Output

Explain

Messages

Notifications

winners_competition

character varying

1 Beetle Of Band

2 INC

3 iRelay

pgAdmin 4

127.0.0.1:49188/browser/

History

Search history

Views

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

pgAdmin

File

Object

Tools

Help

SQL

Statistics

Properties

Dependencies

Dependents

Dashboard

201801030_db/postgres@PostgreSQL 10 *

Browser

cf_db

Collations

Domains

RLS Configurations

RLS Dictionaries

RLS Filters

RLS Templates

Foreign Tables

Functions

Materialized Views

Sequences

Tables (6)

collage

Columns (4)

c_id

c_name

c_address

c_pin

Constraints

Indexes

RLS Policies

Rules

Triggers

competition

Columns (6)

co_id

e_id

co_name

win_price

date

time

Constraints

Indexes

RLS Policies

Rules

Triggers

events

Columns (6)

e_id

e_type

e_name

length_in_u

start_date

end_date

Constraints

Query Editor

Query History

Scratch Pad

```
1 create or replace function "cf_db"."events_competition_list"(s character varying(256))
2 returns table(b character varying(256),price integer,date_1 character varying(180),time_1 character varying(180))
3 language 'plpgsql'
4 as $$
5
6 declare
7     r RECORD;
8
9 begin
10     for i in(select competition.co_name,competition.win_price,competition.date,competition.time from CF_DB.competition join CF_DB.events
11         where events.e_name=s)
12     loop b:=(i.co_name); price:=(i.win_price); date_1:=(i.date); time_1:=(i.time);
13     return next;
14     end loop;
15 end;
16
17 select "cf_db"."events_competition_list"('Synapse');
```

Data Output

Explain

Messages

Notifications

events_competition_list

record

1 ("Beetle Of Band",2500,2020-02-21 16:00 PM)

2 (BartleDrome,2000,2020-02-21 9:00 PM)

3 (Conceive,1400,2020-02-21 12:00 AM)

4 ("Crazy Math",2000,2020-02-21 12:00 PM)

5 (DA Rush,2500,2020-02-21 3:00 PM)

6 ("DA West",1000,2020-02-22 00:00 PM)

7 (Foolhouse,2000,2020-02-22 10:00 AM)

8 ("Goshawk Holmes",2000,2020-02-22 12:00 A.)

9 (Headrush,2500,2020-02-22 3:00 PM)

10 (INC,3000,2020-02-22 4:30 PM)

11 (Misconstrued,3500,2020-02-23 10:00 AM)

12 (Natch,5000,2020-02-23 7:00 PM)

13 (Raaga,7000,2020-02-23 11:00 PM)

14 (Rampage,4000,2020-02-23 9:00 PM)

15 (Encore,3000,2020-02-23 10:00 PM)

16 (Rhapsody,2000,2020-02-24 10:00 AM)

17 (Shades,5000,2020-02-24 2:00 PM)

18 (Showdown,6000,2020-02-24 4:00 PM)

19 (Symphony,4000,2020-02-24 6:00 PM)

pgAdmin 4

127.0.0.1:49180/browsers/

History

Search history

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

pgAdmin

File

Object

Tools

Help

SQL

Statistics

Properties

Dependencies

Dashboard

201801030_db/postgres@PostgreSQL 10

Query Editor

Query History

Scratch Pad

```
1 create or replace function check_teams()
2 returns trigger
3 language 'plpgsql'
4 as $$
5
6 begin
7
8     if(new.team_id IS NULL)
9         then raise notice 'Team ID should not be null';
10
11     return old;
12
13     elsif (new.t_name IS NULL)
14         then raise notice 'Team name should not be null';
15
16     return old;
17
18     elsif (new.t_strength < 1)
19         then raise notice 'Team strength should not be less than one';
20
21     return old;
22
23     end if;
24
25     return new;
26 end
27 $$;
28
29 create trigger "Trigger_teams"
30 before insert on cf_db.teams
31 for each row execute procedure check_teams();
32
33 INSERT INTO "cf_db".teams(team_id,t_name,t_strength) VALUES (67,'team-corona',8);
34
```

Data Output

Explain

Messages

Notifications

NOTICE: Team strength should not be less than one

INSERT 8/8

Query returned successfully in 132 msec.

Type here to search

ENG 5:41 PM 9/20/2020

pgAdmin 4

127.0.0.1:49180/browsers/

History

Search history

Today

Yesterday

Last 7 days

This month

August

July

June

May

April

Older than 6 months

pgAdmin

File

Object

Tools

Help

SQL

Statistics

Properties

Dependencies

Dashboard

201801030_db/postgres@PostgreSQL 10

Query Editor

Query History

Scratch Pad

```
1 create or replace function check_price()
2 returns trigger
3 language 'plpgsql'
4 as $$
5
6 begin
7
8     if(new.win_price < 2000)
9         then new.win_price = 2000;
10         raise notice 'The Win price entered is less than 2000, updated to default price money';
11
12     end if;
13
14     return new;
15 end
16 $$;
17
18 create trigger "winning_price_check"
19 before insert on cf_db.competition
20 for each row execute procedure check_price();
21
22 INSERT INTO "cf_db".competition(co_id,e_id,co_name,win_price,date,time) VALUES (72,3,'corona_1',1000,'2020-02-21','6:00 PM');
23
24 select * from cf_db.competition
25
```

Data Output

Explain

Messages

Notifications

co_id	e_id	co_name	win_price	date	time
36	36	3 Babuchah	12000	2019-10..	4:00 PM
37	37	3 LOHunt	7000	2019-10..	9:00 AM
38	38	3 Deocher	6000	2019-10..	12:00..
39	39	3 I.Biz	4000	2019-10..	2:00 PM
40	40	3 I.Gavth	5000	2019-10..	4:00 PM
41	41	3 Treasurehunt	4000	2019-10..	12:00..
42	67	3 corona	1000	2020-02..	6:00 PM
43	68	3 corona_1	1000	2020-02..	6:00 PM
44	71	3 corona_1	1000	2020-02..	6:00 PM
45	72	3 corona_1	2000	2020-02..	6:00 PM

Type here to search

ENG 8:10 PM 9/20/2020