

Trigger

SID: 12310401

Name: 王子恒

1.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

Run | Copy

CREATE FUNCTION emp_stamp() RETURNS trigger

AS

\$emp_stamp\$

BEGIN

-- Check that salary are given

IF NEW.name IS NULL THEN

RAISE EXCEPTION 'emp_name cannot be null';

END IF;

IF NEW.salary IS NULL THEN

RAISE EXCEPTION '% cannot have null salary', NEW.name;

END IF;

-- Who works for us when they must pay for it?

IF NEW.salary < 0 THEN

RAISE EXCEPTION '% cannot have a negative salary', NEW.name;

END IF;

END;

\$emp_stamp\$ LANGUAGE plpgsql;

Result

搜索结果集

耗时: 14ms

Execution completed in 14ms

问题 1 输出 调试控制台 端口 终端 SPELL CHECKER 2 注释

psql: 错误: 连接到套接字"/tmp/.s.PGSQL.5432"上的服务器失败:FATAL: database "eare
ndelh" does not exist

~/Documents/大二上/CS213DB/ psql -d lab1

psql (15.8 (Homebrew))

输入 "help" 来获取帮助信息.

lab1=# insert into dept values(1, '开发部', 'Bao'an');

insert into dept values(2, '测试部', 'Bao'an');

insert into dept values(3, '财务部');

insert into dept values(4, '市场部', 'Bao'an');

insert into dept values(5, '总经办');

insert into dept values(6, '生产部', 'Bao'an');

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

lab1=#

2.

```
76  --
77  IF (TG_OP = 'DELETE') THEN
78  INSERT INTO emp_log SELECT 'D', now(), user, TG_LEVEL, TG_WHEN,
79  OLD.id, OLD.name, OLD.salary;
80  ELSIF (TG_OP = 'UPDATE') THEN
81  INSERT INTO emp_log SELECT 'U', now(), user, TG_LEVEL, TG_WHEN,
82  NEW.id, NEW.name, NEW.salary;
83  ELSIF (TG_OP = 'INSERT') THEN
84  INSERT INTO emp_log SELECT 'I', now(), user, TG_LEVEL, TG_WHEN,
85  NEW.id, NEW.name, NEW.salary;
86  END IF;
87  RETURN NULL; -- result is ignored since this is an AFTER trigger
88  END;
89  $emp_check$ LANGUAGE plpgsql;
90
91
```

Result

搜索结果集



耗时: 17m

Execution completed in 17ms

3.

Run | New Tab

91 CREATE TRIGGER check_after

92 AFTER INSERT OR UPDATE OR DELETE ON emp

93 FOR EACH ROW

94 EXECUTE FUNCTION process_emp_check();

95 --test, and see result in emp_log

Run | New Tab

96 insert into emp values (017,'贺小山', 7000, 1);

Run | New Tab

97 UPDATE emp set salary=8000 where id=017;

Run | New Tab

98 delete from emp where id=017;

99

Run | New Tab | JSON

100 SELECT * FROM emp_log; 6ms

101

emp_log

搜索结果集

导出

耗时: 33ms

< 1 > Total 3

		* operation character(1)	* stamp timestamp without time	* userid text	* level text	* when_trigger text
		Filter	Filter	Filter	Filter	Filter
	>	I	2024-11-22 20:09:25.86736	earendelh	ROW	AFTER
	>	U	2024-11-22 20:09:51.56276	earendelh	ROW	AFTER
	>	D	2024-11-22 20:09:54.79985	earendelh	ROW	AFTER

4.

Run | New Tab

102 CREATE TRIGGER check_before

103 BEFORE INSERT OR UPDATE OR DELETE ON emp

104 FOR EACH ROW

105 EXECUTE FUNCTION process_emp_check();

106 --test, and see result in emp_log

Run | New Tab

107 insert into emp values (017,'贺小山', 7000, 1);

Run | New Tab

108 UPDATE emp set salary=8000 where id=017;

Run | New Tab

109 delete from emp where id=017;

110

Run | New Tab | JSON

111 SELECT * FROM emp_log; 3ms

112

113

emp_log

搜索结果集

导出

耗时: 27ms

Total 4

		* operation character(1)	* stamp timestamp without time	* userid text	* level text	* when_trigger text
		Filter	Filter	Filter	Filter	Filter
	>	I	2024-11-22 20:09:25.86736	earendelh	ROW	AFTER
	>	U	2024-11-22 20:09:51.56276	earendelh	ROW	AFTER
	>	D	2024-11-22 20:09:54.79985	earendelh	ROW	AFTER
	>	I	2024-11-22 20:10:58.76319	earendelh	ROW	BEFORE

Run | New Tab

127 UPDATE emp set salary=8000 where id=017;

Run | New Tab

128 delete from emp where id=017;

Run | New Tab

129 UPDATE emp set salary=salary*1.2;

Run | New Tab

130 delete from emp where id=013;

131

Run | New Tab | JSON

132 select * from emp_log;

133

134

emp_log

搜索结果集

导出

耗时: 36ms

Total 6

		* operation character(1)	* stamp timestamp without time	* userid text	* level text	* when_trigger text	id integer	name varchar(10)	salary numeric
		Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
	>	I	2024-11-22 20:09:25.86736	earendelh	ROW	AFTER	17	贺小山	7000.00
	>	U	2024-11-22 20:09:51.56276	earendelh	ROW	AFTER	17	贺小山	8000.00
	>	D	2024-11-22 20:09:54.79985	earendelh	ROW	AFTER	17	贺小山	8000.00
	>	I	2024-11-22 20:10:58.76319	earendelh	ROW	BEFORE	17	贺小山	7000.00
	>	I	2024-11-22 20:11:47.10214	earendelh	ROW	BEFORE	17	贺小山	7000.00
	>	D	2024-11-22 20:11:52.79207	earendelh	ROW	BEFORE	13	刘小毛	4500.00

Run | New Tab | Copy

134 CREATE TRIGGER check_update_col

135 AFTER UPDATE OF salary ON emp

136 FOR EACH ROW

137 --WHEN (OLD.salary IS DISTINCT FROM NEW.salary)

138 EXECUTE FUNCTION process_emp_check();

139 --test

Run | New Tab

140 UPDATE emp set deptId = 1 where id=12;

Run | New Tab

141 UPDATE emp set salary=salary*1.2 where id=13;

142

Run | New Tab | JSON

143 select * from emp_log; 2ms

144

145

emp_log

搜索结果集

耗时: 25ms < 1 > Total 8

	* operation	* stamp	* userid	* level	* when_trigger	id	name	salary
	character(1)	timestamp without time	text	text	text	integer	varchar(10)	numeric
	I	2024-11-22 20:09:25.86736	earendelh	ROW	AFTER	17	贺小山	7000.00
	U	2024-11-22 20:09:51.56276	earendelh	ROW	AFTER	17	贺小山	8000.00
	D	2024-11-22 20:09:54.79985	earendelh	ROW	AFTER	17	贺小山	8000.00
	I	2024-11-22 20:10:58.76319	earendelh	ROW	BEFORE	17	贺小山	7000.00
	I	2024-11-22 20:11:47.10214	earendelh	ROW	BEFORE	17	贺小山	7000.00
	D	2024-11-22 20:11:52.79207	earendelh	ROW	BEFORE	13	刘小毛	4500.00
	U	2024-11-22 20:13:06.57842	earendelh	ROW	BEFORE	12	李雷	7000.00
	U	2024-11-22 20:13:07.99442	earendelh	ROW	BEFORE	13	刘小毛	5400.00

6.

Run | New Tab | Copy

145 CREATE TRIGGER check_update_col_when

146 AFTER UPDATE OF salary ON emp

147 FOR EACH ROW

148 WHEN (OLD.salary IS DISTINCT FROM NEW.salary)

149 EXECUTE FUNCTION process_emp_check();

150 --test

Run | New Tab

151 UPDATE emp set deptId = 6 where id=12;

Run | New Tab

152 UPDATE emp set salary=salary*0.9 where id=13;

153

Run | New Tab | JSON

154 select * from emp_log; 2ms

155

156

emp_log

搜索结果集

耗时: 5ms < 1 > Total 10

	* operation	* stamp	* userid	* level	* when_trigger	id	name	salary
	character(1)	timestamp without time	text	text	text	integer	varchar(10)	numeric
	D	2024-11-22 20:09:54.79985	earendelh	ROW	AFTER	17	贺小山	8000.00
	I	2024-11-22 20:10:58.76319	earendelh	ROW	BEFORE	17	贺小山	7000.00
	I	2024-11-22 20:11:47.10214	earendelh	ROW	BEFORE	17	贺小山	7000.00
	D	2024-11-22 20:11:52.79207	earendelh	ROW	BEFORE	13	刘小毛	4500.00
	U	2024-11-22 20:13:06.57842	earendelh	ROW	BEFORE	12	李雷	7000.00
	U	2024-11-22 20:13:07.99442	earendelh	ROW	BEFORE	13	刘小毛	5400.00
	U	2024-11-22 20:13:42.13189	earendelh	ROW	BEFORE	12	李雷	7000.00
	U	2024-11-22 20:13:43.45826	earendelh	ROW	BEFORE	13	刘小毛	4050.00

7.

156 CREATE TRIGGER check_update_col_when

157 AFTER UPDATE ON emp

158 FOR EACH ROW

159 WHEN (OLD.* IS DISTINCT FROM NEW.*)

160 EXECUTE FUNCTION process_emp_check();

161

Run | New Tab

162 UPDATE emp set deptId = 6 where id=12;

163

Run | New Tab | JSON

164 select * from emp_log; 3ms

165

166

emp_log

搜索结果集

导出

耗时: 29ms < 1 > Total 11

	* operation character(1)	* stamp timestamp without time	* userid text	* level text	* when_trigger text	id integer	name varchar(10)	salary numeric
	I	2024-11-22 20:10:58.76319	earendelh	ROW	BEFORE	17	贺小山	7000.00
	I	2024-11-22 20:11:47.10214	earendelh	ROW	BEFORE	17	贺小山	7000.00
	D	2024-11-22 20:11:52.79207	earendelh	ROW	BEFORE	13	刘小毛	4500.00
	U	2024-11-22 20:13:06.57842	earendelh	ROW	BEFORE	12	李雷	7000.00
	U	2024-11-22 20:13:07.99442	earendelh	ROW	BEFORE	13	刘小毛	5400.00
	U	2024-11-22 20:13:42.13189	earendelh	ROW	BEFORE	12	李雷	7000.00
	U	2024-11-22 20:13:43.45826	earendelh	ROW	BEFORE	13	刘小毛	4050.00
	U	2024-11-22 20:14:54.56167	earendelh	ROW	BEFORE	12	李雷	7000.00

8.

166 CREATE TRIGGER check_after_statement

167 AFTER INSERT OR UPDATE OR DELETE ON emp

168 FOR EACH STATEMENT

169 EXECUTE FUNCTION process_emp_check();

170 --test

Run | New Tab

171 insert into emp values (017,'贺小山', 7000, 1);

Run | New Tab

172 UPDATE emp set salary=8000 where id=017;

Run | New Tab

173 delete from emp where id=017;

Run | New Tab

174 UPDATE emp set salary=salary*1.15;

175

Run | New Tab | JSON

176 select * from emp_log; 6ms

177

178

emp_log

搜索结果集

导出

耗时: 29ms < 1 > Total 30

	* operation character(1)	* stamp timestamp without time	* userid text	* level text	* when_trigger text	id integer	name varchar(10)	salary numeric
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	8	殷十	17250.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	9	周一	17250.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	10	范二	28750.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	11	何玲	17250.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	12	李雷	8050.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	13	刘小毛	5175.00
	U	2024-11-22 20:16:09.56261	earendelh	ROW	BEFORE	14	张大壮	6900.00
	U	2024-11-22 20:16:09.56261	earendelh	STATEMENT	AFTER	(NULL)	(NULL)	(NULL)

9.

179 BEFORE INSERT OR UPDATE OR DELETE ON emp

180 FOR EACH STATEMENT

181 EXECUTE FUNCTION process_emp_check();

182 --test

▷ Run | New Tab

183 insert into emp values (017,'贺小山', 7000, 1);

▷ Run | New Tab

184 UPDATE emp set salary=8000 where id=017;

▷ Run | New Tab

185 delete from emp where id=017;

▷ Run | New Tab

186 UPDATE emp set salary=salary*0.95;

187

▷ Run | New Tab | JSON

188 select * from emp_log; 2ms

189

emp_log

搜索结果集

耗时: 26ms < 1 > Total 53

	operation character(1)	stamp timestamp without time	userid text	level text	when_trigger text	id integer	name varchar(10)	salary numeric
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	8	殷十	14250.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	9	周一	14250.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	10	范二	23750.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	11	何玲	14250.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	12	李雷	6650.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	13	刘小毛	4275.00
>	U	2024-11-22 20:17:02.39314	earendelh	ROW	BEFORE	14	张大壮	5700.00
>	U	2024-11-22 20:17:02.39314	earendelh	STATEMENT	AFTER	(NULL)	(NULL)	(NULL)

10.

```
lab1=# create table test(
test1 int,
test2 int
);
NOTICE: Operation here: ddl_command_start CREATE TABLE
CREATE TABLE
lab1=# drop table test;
NOTICE: Operation here: ddl_command_start DROP TABLE
DROP TABLE
lab1=#
```

show triggers


```
select * from pg_catalog.pg_trigger; 14ms
```

pg_catalog.pg_trigger

搜索结果集

导出 耗时: 19ms < 1 > Total 49

	* oid oid	* tgreloid oid	* tgparentid oid	* tgname name	* tgfoid oid	* tgtype smallint	* tgenabled "char"	* tgisint boolean
>	17658	17564	0	RI_ConstraintTrigger_a_17658	1655	17	O	true
>	17659	17649	0	RI_ConstraintTrigger_c_17659	1644	5	O	true
>	17660	17649	0	RI_ConstraintTrigger_c_17660	1645	17	O	true
>	17668	17564	0	RI_ConstraintTrigger_a_17668	1646	9	O	true
>	17669	17564	0	RI_ConstraintTrigger_a_17669	1655	17	O	true
>	17670	17661	0	RI_ConstraintTrigger_c_17670	1644	5	O	true
>	17671	17661	0	RI_ConstraintTrigger_c_17671	1644	17	O	true
>	17673	17649	0	RI_ConstraintTrigger_a_17673	1654	9	O	true
>	17674	17649	0	RI_ConstraintTrigger_a_17674	1655	17	O	true
>	17675	17661	0	RI_ConstraintTrigger_c_17675	1644	5	O	true
>	17676	17661	0	RI_ConstraintTrigger_c_17676	1645	17	O	true