Практическая работа. Табличные триггеры.

2 Вариант.

--1.Запретить удаление авиакомпании, если она участвует в международных рейсах

-- Создаем функцию, которая проверяет, есть ли международные рейсы для данной авиакомпании

CREATE OR REPLACE FUNCTION check\_international\_flights()

RETURNS TRIGGER AS $$

BEGIN

IF EXISTS (

SELECT 1

FROM routetypesrt

JOIN routes r ON rt.routetype\_id= r.routetype\_id

WHERE rt.name = I AND r.airline\_id= OLD.airline\_id

) THEN

RAISE EXCEPTION Нельзя удалить авиакомпанию, участвующую в международных рейсах;

END IF;

RETURN OLD;

END;

$$ LANGUAGE plpgsql;

-- Создаем триггер, который использует функцию для проверки перед удалением авиакомпании

CREATE TRIGGER prevent\_deletion\_of\_airline

BEFORE DELETE ON Airlines

FOR EACH ROW

EXECUTE FUNCTION check\_international\_flights();

New

-- Создаем функцию, которая проверяет, есть ли международные рейсы для данной авиакомпании.

CREATE OR REPLACE FUNCTION check\_international\_flights()

RETURNS TRIGGER AS $$

BEGIN

IF EXISTS (

SELECT 1

FROM routetypes rt

JOIN routes r ON rt.routetype\_id = r.routetype\_id

WHERE rt.name = 'международный' AND r.airline\_id = OLD.airline\_id

) THEN

RAISE EXCEPTION 'Нельзя удалить авиакомпанию, участвующую в международных рейсах';

END IF;

RETURN OLD;

END;

$$ LANGUAGE plpgsql;

-- Создаем триггер, который использует функцию для проверки перед удалением авиакомпании

CREATE TRIGGER prevent\_deletion\_of\_airline

BEFORE DELETE ON airlines

FOR EACH ROW

EXECUTE FUNCTION check\_international\_flights();

2.--Отслеживать все изменения данных в таблице Aircrafts,

--записывая в таблицу ARCHIVE пользователя,

--который произвел изменения, дату ивремя изменения,

--действие (Insert/Update/Delete) иколичество измененных строк.

CREATE TABLE ARCHIVE (

operation char(1) NOT NULL,

stamp timestamp NOT NULL,

userid text NOT NULL,

cunt integer not null);

CREATE OR REPLACE FUNCTION process\_aircrafts\_audit() RETURNS trigger AS $aircrafts\_audit$

BEGIN

-- В зависимости от типа операции записываем соответствующую информацию в таблицу ARCHIVE

IF (TG\_OP = DELETE) THEN

INSERT INTO ARCHIVE(operation, stamp, userid, cunt) VALUES (D,CURRENT\_TIMESTAMP, current\_user, TG\_NARGS);

ELSIF (TG\_OP = UPDATE) THEN

INSERT INTO ARCHIVE (operation, stamp, userid, cunt) VALUES (U,CURRENT\_TIMESTAMP, current\_user, TG\_NARGS);

ELSIF (TG\_OP = INSERT) THEN

INSERT INTO ARCHIVE (operation, stamp, userid, cunt) VALUES (I,CURRENT\_TIMESTAMP, current\_user, TG\_NARGS);

END IF;

IF TG\_WHEN = BEFORE THEN

RETURN NEW;

ELSE

RETURN OLD;

END IF;

END;

$aircrafts\_audit$ LANGUAGE plpgsql;

CREATE TRIGGER aircrafts\_audit\_trig

BEFORE INSERT OR UPDATE OR DELETE ON Aircrafts FOR EACH ROW

EXECUTE PROCEDURE process\_aircrafts\_audit();

New

CREATE TABLE ARCHIVE (

operation char(1) NOT NULL,

stamp timestamp NOT NULL,

userid text NOT NULL,

cnt integer NOT NULL

);

CREATE OR REPLACE FUNCTION process\_aircrafts\_audit() RETURNS TRIGGER AS $aircrafts\_audit$

BEGIN

-- В зависимости от типа операции записываем соответствующую информацию в таблицу ARCHIVE

IF (TG\_OP = 'DELETE') THEN

INSERT INTO ARCHIVE (operation, stamp, userid, cnt) VALUES ('D', CURRENT\_TIMESTAMP, current\_user, 1);

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO ARCHIVE (operation, stamp, userid, cnt) VALUES ('U', CURRENT\_TIMESTAMP, current\_user, 1);

ELSIF (TG\_OP = 'INSERT') THEN

INSERT INTO ARCHIVE (operation, stamp, userid, cnt) VALUES ('I', CURRENT\_TIMESTAMP, current\_user, 1);

END IF;

IF TG\_WHEN = 'BEFORE' THEN

RETURN NEW;

ELSE

RETURN OLD;

END IF;

END;

$aircrafts\_audit$ LANGUAGE plpgsql;

CREATE TRIGGER aircrafts\_audit\_trig

BEFORE INSERT OR UPDATE OR DELETE ON Aircrafts FOR EACH ROW

EXECUTE PROCEDURE process\_aircrafts\_audit();