#########################################

## Jakob Roberts - v00484900

## CSC370

## Assignment 6

#########################################

##################################################################################

db14=> \d parts

Table "public.parts"

Column | Type | Modifiers

--------+-----------------------+-----------

pid | integer |

pname | character varying(40) |

color | character varying(20) |

Triggers:

do\_check AFTER INSERT OR DELETE OR UPDATE ON parts FOR EACH ROW EXECUTE PROCEDURE trigfunc()

db14=> \d partshistory

Table "public.partshistory"

Column | Type | Modifiers

-----------+-----------------------------+-----------

pid | integer |

pname | character varying(40) |

color | character varying(20) |

operation | character(1) |

opwhen | timestamp without time zone |

opuser | character(20) |

db14=>

#################################################################################

db14=> \df

List of functions

Schema | Name | Result data type | Argument data types | Type

--------+----------+------------------+---------------------+---------

public | trigfunc | trigger | | trigger

(1 row)

db14=>

#################################################################################

**TABLE CREATION**

CREATE TABLE parts (

pid INTEGER,

pname VARCHAR(40),

color VARCHAR(20),

);

CREATE TABLE partshistory (

pid INTEGER,

pname VARCHAR(40),

color VARCHAR(20),

operation CHAR,

opwhen TIMESTAMP,

opuser CHAR(20)

);

#################################################################################

**UDF & TRIGGER CREATION**

CREATE OR REPLACE FUNCTION trigfunc ( )

RETURNS TRIGGER AS $partshistory$

BEGIN

IF (TG\_OP = 'INSERT') THEN

INSERT INTO partshistory SELECT NEW.\*, 'I', now(), user;

RETURN NEW;

ELSIF (TG\_OP = 'UPDATE') THEN

INSERT INTO partshistory SELECT NEW.\*, 'U', now(), user;

RETURN NEW;

ELSIF (TG\_OP = 'DELETE') THEN

INSERT INTO partshistory SELECT OLD.\*, 'D', now(), user;

RETURN OLD;

END IF;

RETURN NULL;

END;

$partshistory$ LANGUAGE plpgsql;

CREATE TRIGGER do\_check

AFTER INSERT OR UPDATE OR DELETE ON parts

FOR EACH ROW

EXECUTE PROCEDURE trigfunc();