-- TRIGGERS

-- -----------------------------------------------------

-- trigger chequear\_mail

-- -----------------------------------------------------

DELIMITER $$

CREATE trigger `chequear\_mail` after insert on usuario

for each row

BEGIN

declare invitador int default null;

declare espacio decimal(10,2) default null;

select usuario\_invitador into invitador from invitacion where mail\_invitado = NEW.mail;

insert into almacenamiento\_extra (usuario\_idusuario,extra) values(new.idusuario,0.00);

insert into carpeta (nombre,borrado,usuario\_idusuario) VALUES ("MI CARPETA",0,new.idusuario);

if ((select categoria\_usuario\_idcategoria\_usuario from usuario where idusuario = invitador) = 0) then

select extra into espacio from almacenamiento\_extra where usuario\_idusuario = invitador;

set espacio = espacio + 0.5;

if ( espacio <= 16) then

update almacenamiento\_extra set extra = (extra + 0.5) where usuario\_idusuario = invitador;

end if;

end if;

END$$

DELIMITER ;

-- -----------------------------------------------------

-- trigger aumentar\_almacenamiento\_utilizado

-- -----------------------------------------------------

DELIMITER $$

CREATE trigger `aumentar\_almacenamiento\_utilizado` after insert on archivo

for each row

BEGIN

declare usuarioDueno int;

select carpeta.usuario\_idusuario into usuarioDueno from carpeta where idcarpeta = new.carpeta\_idcarpeta;

update usuario set almacenamiento\_utilizado = (almacenamiento\_utilizado + new.peso) where idusuario = usuarioDueno;

END$$

DELIMITER ;

-- -----------------------------------------------------

-- trigger auditoria\_archivo

-- -----------------------------------------------------

DELIMITER $$

CREATE trigger `auditoria\_archivo` after update on archivo

for each row

BEGIN

declare dueno int;

declare diferencia decimal(10,2);

select usuario\_idusuario into dueno from carpeta where idcarpeta = old.carpeta\_idcarpeta;

insert into auditoria\_archivo

set idarchivo = old.idarchivo,

contenido = old.contenido,

nombre = old.nombre,

borrado = old.borrado,

peso = old.peso,

carpeta\_idcarpeta = old.carpeta\_idcarpeta,

fecha = now();

if(old.borrado <> new.borrado) then

if(new.borrado = 1) then

update usuario set almacenamiento\_utilizado = (almacenamiento\_utilizado - old.peso) where idusuario = dueno;

elseif(new.borrado = 0) then

update usuario set almacenamiento\_utilizado = (almacenamiento\_utilizado + old.peso) where idusuario = dueno;

end if;

end if;

if(old.peso <> new.peso) then

if(old.peso<new.peso) then

set diferencia = (new.peso - old.peso);

update usuario set almacenamiento\_utilizado = (almacenamiento\_utilizado + diferencia) where idusuario = dueno;

else

set diferencia = (old.peso - new.peso);

update usuario set almacenamiento\_utilizado = (almacenamiento\_utilizado - diferencia) where idusuario = dueno;

end if;

end if;

END$$

DELIMITER ;

-- -----------------------------------------------------

-- trigger estado\_de\_carpeta

-- -----------------------------------------------------

DELIMITER $$

CREATE TRIGGER estado\_de\_carpeta

AFTER UPDATE ON carpeta

FOR EACH ROW

BEGIN

-- declaro variable del handler

DECLARE fin INT DEFAULT 0;

DECLARE id int DEFAULT 0;

-- declaro cursor

DECLARE identificacion CURSOR FOR

SELECT DISTINCT idarchivo

FROM archivo

WHERE carpeta\_idcarpeta = NEW.idcarpeta;

-- declaro NOT FOUND handler

DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1;

IF(OLD.borrado <> NEW.borrado) THEN

OPEN identificacion;

obtenerID: LOOP

FETCH identificacion INTO id;

IF (fin = 1) THEN

LEAVE obtenerID;

END IF;

-- chequeo si el NEW.borrado es 1 o 0, y despues ingreso el id(archivo) en el procedure

IF (NEW.borrado = 1) THEN

call eliminar\_archivo(id,new.usuario\_idusuario,@msg);

ELSEIF (NEW.borrado = 0) THEN

call recuperar\_archivo(id,new.usuario\_idusuario,@msg);

END IF;

END LOOP obtenerID;

CLOSE identificacion;

END IF;

END $$

DELIMITER ;