**Vincolo emailFormat**

ALTER TABLE Email ADD CONSTRAINT emailFormat CHECK ( email LIKE '\_%@\_%.\_\_%')

**Vincoli passwordLen e usernameLen**

ALTER TABLE R\_User

ADD CONSTRAINT passwordLen CHECK (LENGTH(password)>7),

ADD CONSTRAINT usernameLen CHECK (LENGTH(nickname)>2)

**Vincolo uniqueMainAddress**

CREATE FUNCTION uniqueMainAddress()

RETURNS TRIGGER AS $uniqueMainAddress$

BEGIN

IF NEW.main = TRUE AND NEW.contactID IN (SELECT contactID FROM AssignedAddress AS AA WHERE main = 'true') THEN

RAISE EXCEPTION 'A main e-mail for this contact already exists!';

END IF;

RETURN NEW;

END;

$uniqueMainAddress$ LANGUAGE plpgsql;

CREATE TRIGGER uniqueMainAddress

BEFORE INSERT OR UPDATE ON AssignedAddress

FOR EACH ROW

EXECUTE PROCEDURE uniqueMainAddress();

**Vincolo checkContactNumbers**

CREATE FUNCTION checkContactNumbers()

RETURNS TRIGGER AS $checkContactNumbers$

BEGIN

IF NOT EXISTS (

SELECT \*

FROM AssignedPhone as AP, PhoneNumber as PN

WHERE AP.phoneNumber = PN.phoneNumber and OLD.contactID = AP.contactID and PN.phoneType = 'MOBILE')

OR NOT EXISTS (

SELECT \*

FROM AssignedPhone as AP, PhoneNumber as PN

WHERE AP.phoneNumber = PN.phoneNumber and OLD.contactID = AP.contactID and PN.phoneType = 'LANDLINE')

THEN

RAISE EXCEPTION 'A contact must have at least one landline number and one mobile number';

END IF;

RETURN NEW;

END;

$checkContactNumbers$ LANGUAGE plpgsql;

CREATE TRIGGER checkContactNumbers

BEFORE DELETE OR UPDATE ON AssignedPhone

FOR EACH ROW

EXECUTE PROCEDURE checkContactNumbers();

**Vincolo distinctEmail**

ALTER TABLE Email

ADD CONSTRAINT distinctEmail UNIQUE (email, contactID)

**Vincolo checkCallType**

ALTER TABLE PhoneCall

ADD CONSTRAINT checkCallType CHECK (callType in ('SENT', 'ENTERED', 'MISSED'))