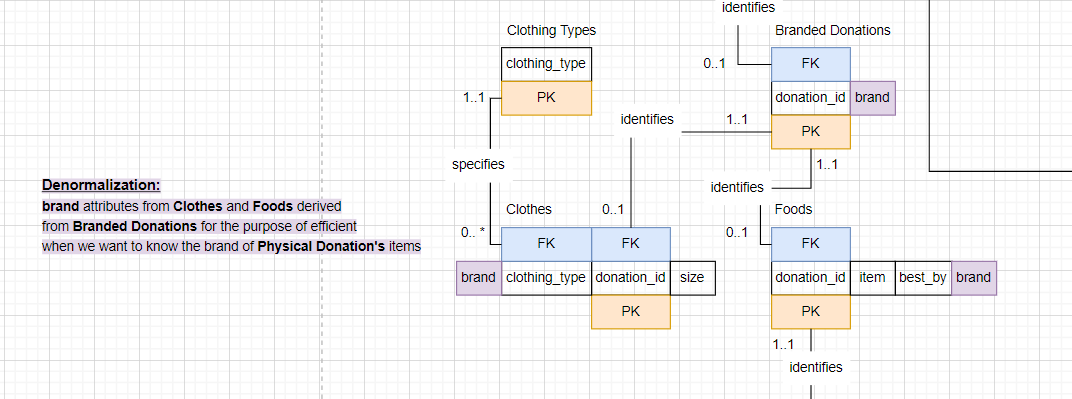
**Denormalization:**

* The denormalization shows the brand of Clothes and Foods from Physical Donations. First, cloning the brand in BrandedDonation (Parent) to other columns of Clothes and Foods (Children). Then create triggers to make sure that the data will stay in sync across all tables.
* RS:



**Code:**

-- Cloning

ALTER TABLE Clothes

ADD COLUMN Brand varchar(50) NOT NULL;

UPDATE Clothes C

INNER JOIN BrandedDonations D using(donation\_id)

SET C.brand = D.brand;

ALTER TABLE Foods

ADD COLUMN Brand varchar(50) NOT NULL;

UPDATE Foods F

INNER JOIN BrandedDonations D using(donation\_id)

SET F.brand = D.brand;

-- Triggers

-- on insert trigger in Clothes and Foods that will copy the value of the Branded Donation's brand from Branded Donation into Clothes and Foods

DELIMITER !!

CREATE TRIGGER `clothes\_BEFORE\_INSERT` before insert ON `clothes` FOR EACH ROW

BEGIN

set new.brand = (

select brand

from BrandedDonations

where donation\_ID = new.donation\_ID);

END !!

DELIMITER ;

DELIMITER !!

CREATE TRIGGER `foods\_BEFORE\_INSERT` before insert ON `foods` FOR EACH ROW

BEGIN

set new.brand = (

select brand

from BrandedDonations

where donation\_ID = new.donation\_ID);

END !!

DELIMITER ;

-- on-update trigger in Branded Donation that will make sure that you cannot change the value of the brand from within Clothes or Foods but on Branded Donation instead. After updating the brand on Branded Donations, the data will sync with Clothes and Foods

CREATE TRIGGER `clothes\_AFTER\_UPDATE` AFTER UPDATE ON `brandeddonations` FOR EACH ROW

BEGIN

UPDATE clothes C SET C.brand = new.brand

where c.donation\_id = new.donation\_id;

end;

CREATE TRIGGER `foods\_AFTER\_UPDATE` AFTER UPDATE ON `brandeddonations` FOR EACH ROW

BEGIN

UPDATE foods F SET F.brand = new.brand

where F.donation\_id = new.donation\_id;

end;

DELIMITER !!

CREATE TRIGGER `clothes\_BEFORE\_UPDATE` BEFORE UPDATE ON `clothes` FOR EACH ROW

BEGIN

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

set new.brand = (

select brand

from BrandedDonations

where BrandedDonations.donation\_ID = new.donation\_ID);

-- SIGNAL SQLSTATE '45000'

-- SET MESSAGE\_TEXT = 'Please update the brand from the Branded Donation query';

end if;

END !!

DELIMITER ;

drop trigger `foods\_BEFORE\_UPDATE`;

DELIMITER !!

CREATE TRIGGER `foods\_BEFORE\_UPDATE` BEFORE UPDATE ON `foods` FOR EACH ROW

BEGIN

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

set new.brand = (

select brand

from BrandedDonations

where BrandedDonations.donation\_ID = new.donation\_ID);

end if;

END !!

DELIMITER ;