Task 1 Teknologi Basis Data

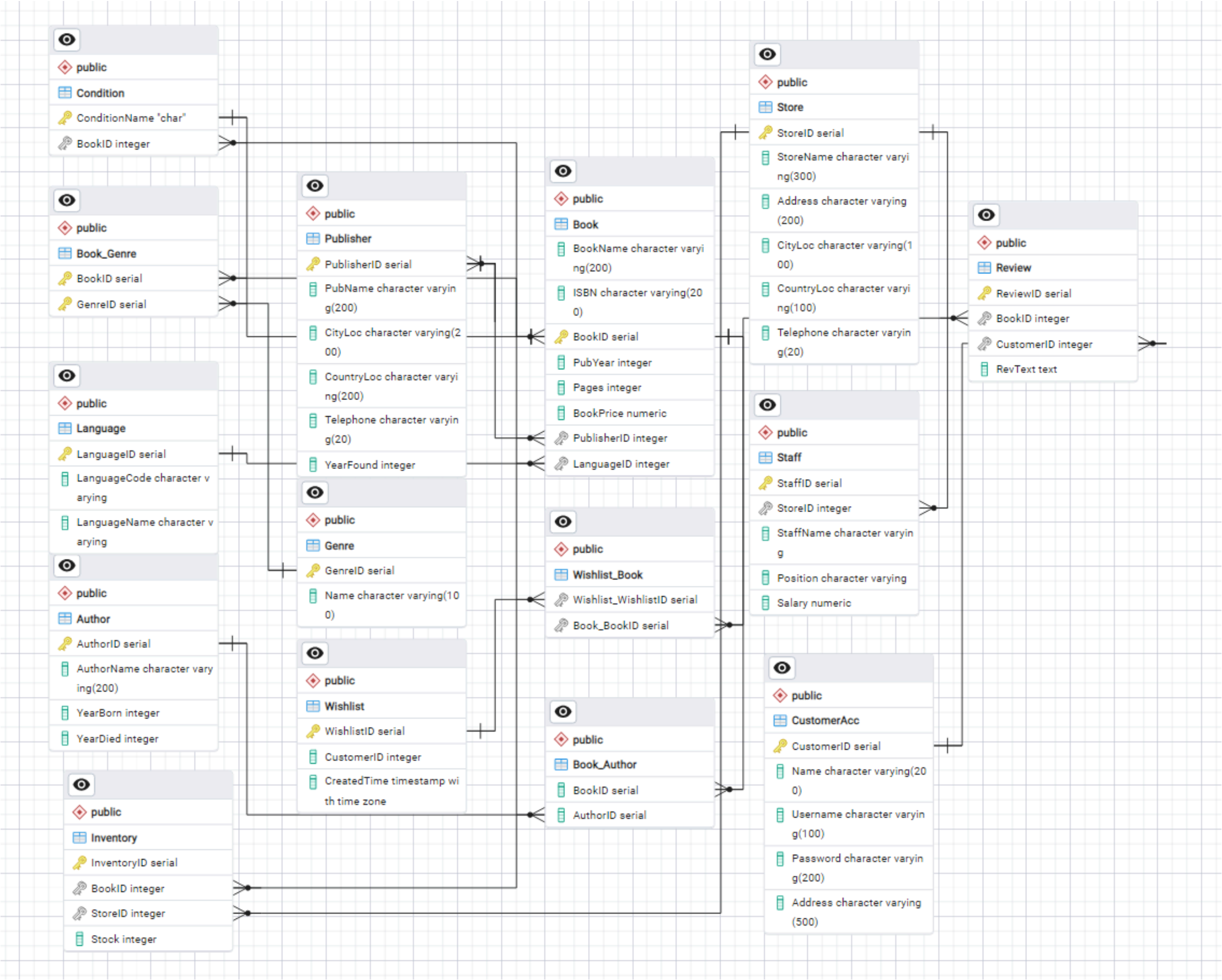
DB Expansion and Implementation

Oleh :

Jhon Samuel Kudadiri

22/503772/TK/55066

A. Entity Relationship Diagram (ERD)



Relationship Explanation:

Dengan menggunakan pgAdmin 4 postgresql, diagram di atas dibuat.

Berikut adalah penjelasan singkat mengenai setiap relationship yang ada:

1. Author - Book_Author
   - Setiap Author bisa memiliki beberapa baris di Book_Author
   - Setiap Book_Author hanya punya satu Author
2. Book_Author - Book
   - Setiap Book bisa memiliki beberapa baris di Book_Author

   - Setiap Book_Author hanya punya satu Author

3. Genre - Book_Genre
   - Satu Genre dapat termuat di dalam beberapa Book_Genre
   - Satu Book_Genre hanya punya satu Genre
4. Book_Genre - Book
   - Satu Book_Genre hanya punya satu Book
   - Satu Book dapat memiliki beberapa Book_Genre
5. Publisher - Book
   - Satu Publisher dapat mempublikasi beberapa Book
   - Satu Book hanya dipublikasi satu Publisher
6. CustomerAcc - Wishlist
   - Satu CustomerAcc hanya memiliki hanya punya satu Wishlist
   - Satu Wishlist hanya dimiliki satu CustomerAcc
7. Wishlist - Wishlist_Book
   - Satu Wishlist dapat memiliki beberapa Wishlist_Book
   - Satu Wishlist_Book hanya punya satu Wishlist
8. Book - Inventory
   - Satu Book dapat memiliki beberapa Inventory
   - Satu Inventory hanya memiliki satu Book
9. Store - Inventory
   - Satu Store dapat memiliki beberapa Inventory
   - Satu Inventory hanya memiliki satu Store
10. Store- Staff
    - Setiap Store dapat memiliki beberapa Staff
    - Satu Staff hanya dapat bekerja dalam satu Store
11. Book - Review
    - Satu Book dapat memiliki beberapa Review

- Satu Review hanya berkaitan dengan satu Book

12. Review - CustomerAcc
    - Satu Review hanya dibuat satu CustomerAcc
    - Satu CustomerAcc dapat membuat beberapa review
13. Language - Book
    - Satu Language dapat memiliki beberapa Book
    - Satu Book hanya ditulis dalam satu Language
14. Condition - Book
    - Satu Book hanya memiliki memiliki satu Condition
    - Satu Condition dapat memiliki beberapa Book

B. PostgresSQL Database Schema : Database Definition Language SQL

```
BEGIN;


CREATE TABLE IF NOT EXISTS public."Author"
(
    "AuthorID" serial NOT NULL,
    "AuthorName" character varying(200) NOT NULL,
    "YearBorn" integer NOT NULL,
    "YearDied" integer NOT NULL,
    PRIMARY KEY ("AuthorID")
);

CREATE TABLE IF NOT EXISTS public."Publisher"
(
    "PublisherID" serial NOT NULL,
    "PubName" character varying(200) NOT NULL,
    "CityLoc" character varying(200) NOT NULL,
    "CountryLoc" character varying(200) NOT NULL,
    "Telephone" character varying(20) NOT NULL,
    "YearFound" integer NOT NULL,
    PRIMARY KEY ("PublisherID")
);

CREATE TABLE IF NOT EXISTS public."Genre"
(
    "GenreID" serial NOT NULL,
    "Name" character varying(100) NOT NULL,
    PRIMARY KEY ("GenreID")
);

CREATE TABLE IF NOT EXISTS public."Book"
(
    "BookName" character varying(200) NOT NULL,
    "ISBN" character varying(200),
    "BookID" serial NOT NULL,
    "PubYear" integer,
    "Pages" integer,
    "BookPrice" numeric,
    "PublisherID" integer,
    "LanguageID" integer,
    "ConditionName" character varying,
    PRIMARY KEY ("BookID")
);

CREATE TABLE IF NOT EXISTS public."Review"
(
    "ReviewID" serial NOT NULL,
    "BookID" integer NOT NULL,
    "CustomerID" integer,
    "RevText" text,
    PRIMARY KEY ("ReviewID")
);
```

```
CREATE TABLE IF NOT EXISTS public."Language"
(
    "LanguageID" serial NOT NULL,
    "LanguageCode" character varying NOT NULL,
    "LanguageName" character varying NOT NULL,
    PRIMARY KEY ("LanguageID")
);

CREATE TABLE IF NOT EXISTS public."Wishlist"
(
    "WishlistID" serial NOT NULL,
    "CustomerID" integer,
    "CreatedTime" timestamp with time zone,
    PRIMARY KEY ("WishlistID")
);

CREATE TABLE IF NOT EXISTS public."CustomerAcc"
(
    "CustomerID" serial,
    "Name" character varying(200),
    "Username" character varying(100),
    "Password" character varying(200),
    "Address" character varying(500),
    PRIMARY KEY ("CustomerID")
);

CREATE TABLE IF NOT EXISTS public."Inventory"
(
    "InventoryID" serial NOT NULL,
    "BookID" integer,
    "StoreID" integer,
    "Stock" integer,
    PRIMARY KEY ("InventoryID")
);
```

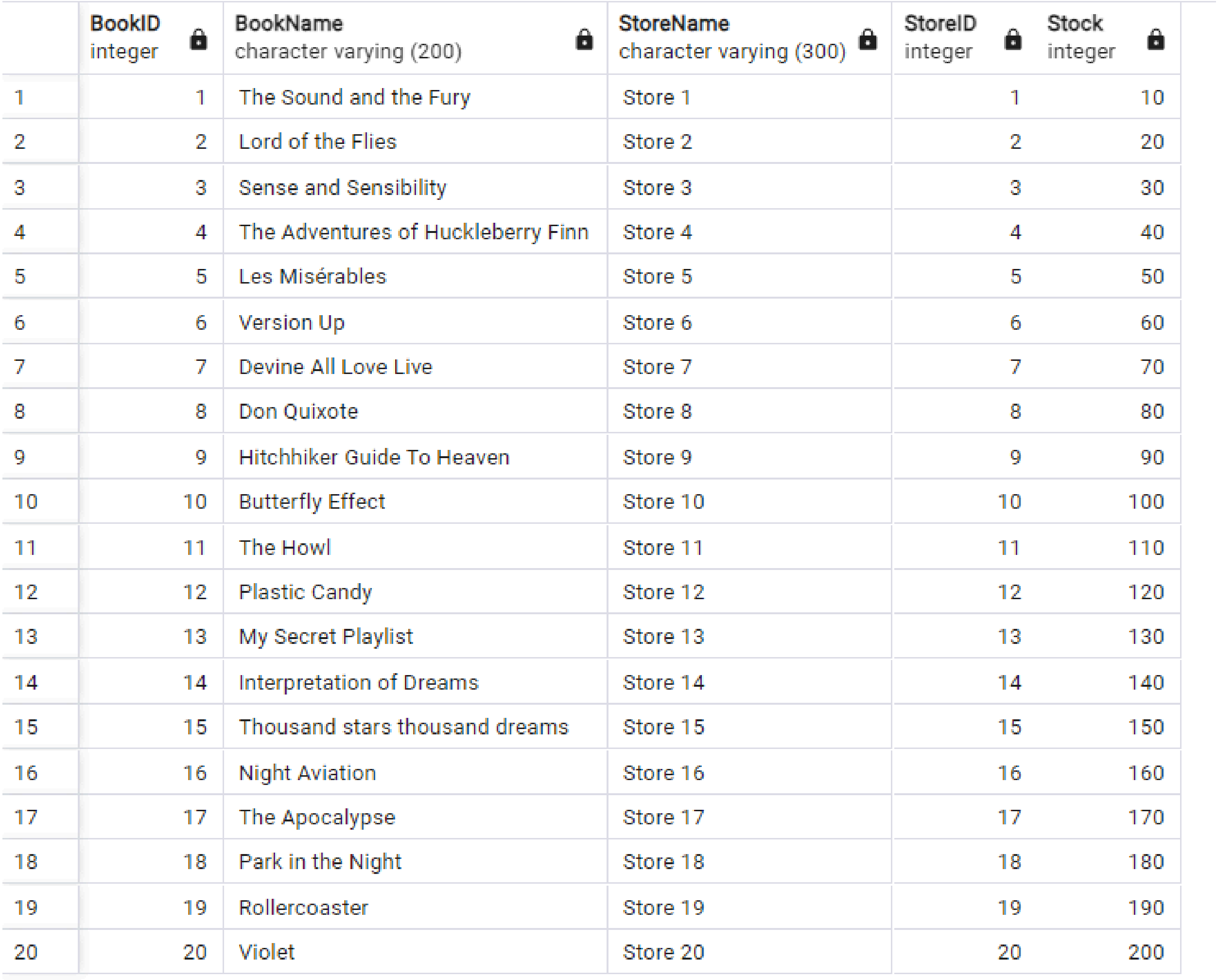
```
CREATE TABLE IF NOT EXISTS public."Store"
(
    "StoreID" serial,
    "StoreName" character varying(300),
    "Address" character varying(200),
    "CityLoc" character varying(100),
    "CountryLoc" character varying(100),
    "Telephone" character varying(20),
    PRIMARY KEY ("StoreID")
);

CREATE TABLE IF NOT EXISTS public."Staff"
(
    "StaffID" serial NOT NULL,
    "StoreID" integer,
    "StaffName" character varying,
    "Position" character varying,
    "Salary" numeric,
    PRIMARY KEY ("StaffID")
);

CREATE TABLE IF NOT EXISTS public."Book_Author"
(
    "BookID" serial NOT NULL,
    "AuthorID" serial NOT NULL,
    PRIMARY KEY ("BookID", "AuthorID")
);

CREATE TABLE IF NOT EXISTS public."Book_Genre"
(
    "BookID" serial NOT NULL,
    "GenreID" serial NOT NULL,
    PRIMARY KEY ("GenreID", "BookID")
);

CREATE TABLE IF NOT EXISTS public."Wishlist_Book"
(
    "WishlistID" serial NOT NULL,
    "BookID" serial NOT NULL,
    PRIMARY KEY ("WishlistID", "BookID")
);

CREATE TABLE IF NOT EXISTS public."Condition"
(
    "ConditionName" character varying(200) NOT NULL,
    PRIMARY KEY ("ConditionName")
);

ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "LanguageID" FOREIGN KEY ("LanguageID")
    REFERENCES public."Language" ("LanguageID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "PublisherID" FOREIGN KEY ("PublisherID")
    REFERENCES public."Publisher" ("PublisherID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "ConditionName" FOREIGN KEY ("ConditionName")
    REFERENCES public."Condition" ("ConditionName") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Review"
    ADD CONSTRAINT "BookID" FOREIGN KEY ("BookID")
    REFERENCES public."Book" ("BookID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Review"
    ADD CONSTRAINT "CustomerID" FOREIGN KEY ("CustomerID")
    REFERENCES public."CustomerAcc" ("CustomerID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Inventory"
    ADD CONSTRAINT "BookID" FOREIGN KEY ("BookID")
    REFERENCES public."Book" ("BookID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Inventory"
    ADD CONSTRAINT "StoreID" FOREIGN KEY ("StoreID")
    REFERENCES public."Store" ("StoreID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Staff"
    ADD CONSTRAINT "StoreID" FOREIGN KEY ("StoreID")
    REFERENCES public."Store" ("StoreID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Book_Author"
    ADD FOREIGN KEY ("BookID")
    REFERENCES public."Book" ("BookID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Book_Author"
    ADD FOREIGN KEY ("AuthorID")
    REFERENCES public."Author" ("AuthorID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;


ALTER TABLE IF EXISTS public."Book_Genre"
    ADD FOREIGN KEY ("BookID")
    REFERENCES public."Book" ("BookID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book_Genre"
   ADD FOREIGN KEY ("GenreID")
   REFERENCES public."Genre" ("GenreID") MATCH SIMPLE
   ON UPDATE NO ACTION
   ON DELETE NO ACTION
   NOT VALID;


ALTER TABLE IF EXISTS public."Wishlist_Book"
   ADD FOREIGN KEY ("WishlistID")
   REFERENCES public."Wishlist" ("WishlistID") MATCH SIMPLE
   ON UPDATE NO ACTION
   ON DELETE NO ACTION
   NOT VALID;


ALTER TABLE IF EXISTS public."Wishlist_Book"
   ADD FOREIGN KEY ("BookID")
   REFERENCES public."Book" ("BookID") MATCH SIMPLE
   ON UPDATE NO ACTION
   ON DELETE NO ACTION
   NOT VALID;

END;
```

Membuat VIEW

- Books_Availability

```
CREATE VIEW public."Books_Availability" AS
SELECT
        b."BookID",
        b."BookName",
        s."StoreName",
        s."StoreID",
        be."Stock"
FROM
        public."Book" b
JOIN
        public."Inventory" be ON b."BookID" = be."BookID"
JOIN
        public."Store" s ON be."StoreID" = s."StoreID"
```



	BookID integer	BookName character varying (200)	StoreName character varying (300)	StoreID integer	Stock integer
1	1	The Sound and the Fury	Store 1	1	10
2	2	Lord of the Flies	Store 2	2	20
3	3	Sense and Sensibility	Store 3	3	30
4	4	The Adventures of Huckleberry Finn	Store 4	4	40
5	5	Les Misérables	Store 5	5	50
6	6	Version Up	Store 6	6	60
7	7	Devine All Love Live	Store 7	7	70
8	8	Don Quixote	Store 8	8	80
9	9	Hitchhiker Guide To Heaven	Store 9	9	90
10	10	Butterfly Effect	Store 10	10	100
11	11	The Howl	Store 11	11	110
12	12	Plastic Candy	Store 12	12	120
13	13	My Secret Playlist	Store 13	13	130
14	14	Interpretation of Dreams	Store 14	14	140
15	15	Thousand stars thousand dreams	Store 15	15	150
16	16	Night Aviation	Store 16	16	160
17	17	The Apocalypse	Store 17	17	170
18	18	Park in the Night	Store 18	18	180
19	19	Rollercoaster	Store 19	19	190
20	20	Violet	Store 20	20	200

- Books Genre

```
CREATE VIEW public."View_Genre_Books" AS
SELECT
   g."GenreName",
   b."BookName"
FROM
   public."Genre" g
JOIN
   public."Book_Genre" bg ON g."GenreID" = bg."GenreID"
JOIN
   public."Book" b ON bg."BookID" = b."BookID";
```

C. Postgresql Data Population

```
BEGIN;

-- Insert data into "Author" table
INSERT INTO public."Author" ("AuthorName", "YearBorn", "YearDied") VALUES
('Markus Zusak', 1900, 1980),
('Ray Bradbury', 1910, 1990),
('Toni Morrison', 1920, 2000),
('Luoisa May Alcott', 1930, 2010),
('Ernest Hemmingway', 1940, 2020),
('Little Women', 1950, 2000),
('Jeon Heejin', 1960, 1999),
('Kim Hyunjin', 1970, 2005),
('Jo Haseul', 1980, 2020),
('Wong Kahei', 1990, 2020),
('Kim Jungeun', 1901, 1981),
('Jeong Jinsol', 1911, 1991),
('Choi Yerim', 1921, 2001),
('Ha Sooyoung', 1931, 2011),
('Kim Jiwoo', 1941, 2021),
('Park Chaewon', 1951, 2000),
('Son Hyeju', 1961, 2005),
('Olivia Hye', 1971, 2022),
('Jaden Flaubert', 1981, 0),
('Jang Wonyoung', 1991, 0);

-- Insert data into "Publisher" table
INSERT INTO public."Publisher" ("PubName", "CityLoc", "CountryLoc", "Telephone", "YearFound")
VALUES
('Egerton', 'City 1', 'Country 1', '111-111-1111', 1950),
('Lippincot', 'City 2', 'Country 2', '222-222-2222', 1960),
('Allen George', 'City 3', 'Country 3', '333-333-3333', 1970),
('Chatto', 'City 4', 'Country 4', '444-444-4444', 1980),
('Russian Messenger', 'City 5', 'Country 5', '555-555-5555', 1990),
('Harper Row', 'City 6', 'Country 6', '666-666-6666', 2000),
('Lacroix Verb', 'City 7', 'Country 7', '777-777-7777', 2010),
('Sylvia Brond', 'City 8', 'Country 8', '888-888-8888', 2020),
('Francisco de Robles', 'City 9', 'Country 9', '999-999-9999', 1995),
('Gallimand', 'City 10', 'Country 10', '101-101-1010', 1985),
('Faber n Faber', 'City 11', 'Country 11', '111-111-1111', 1951),
('Viking Press', 'City 12', 'Country 12', '222-222-2222', 1961),
('Blockberry', 'City 13', 'Country 13', '333-333-3333', 1971),
('Top Creative', 'City 14', 'Country 14', '444-444-4444', 1981),
('Wakeone Press', 'City 15', 'Country 15', '555-555-5555', 1991),
('Mystical Hall', 'City 16', 'Country 16', '666-666-6666', 2001),
('Modhaus Press', 'City 17', 'Country 17', '777-777-7777', 2011),
('ATRP Publisher', 'City 18', 'Country 18', '888-888-8888', 2021),
('Paix Per Mill', 'City 19', 'Country 19', '999-999-9999', 1996),
('CTD Press', 'City 20', 'Country 20', '101-101-1010', 1986);

-- Insert data into "Genre" table
INSERT INTO public."Genre" ("Name") VALUES
```

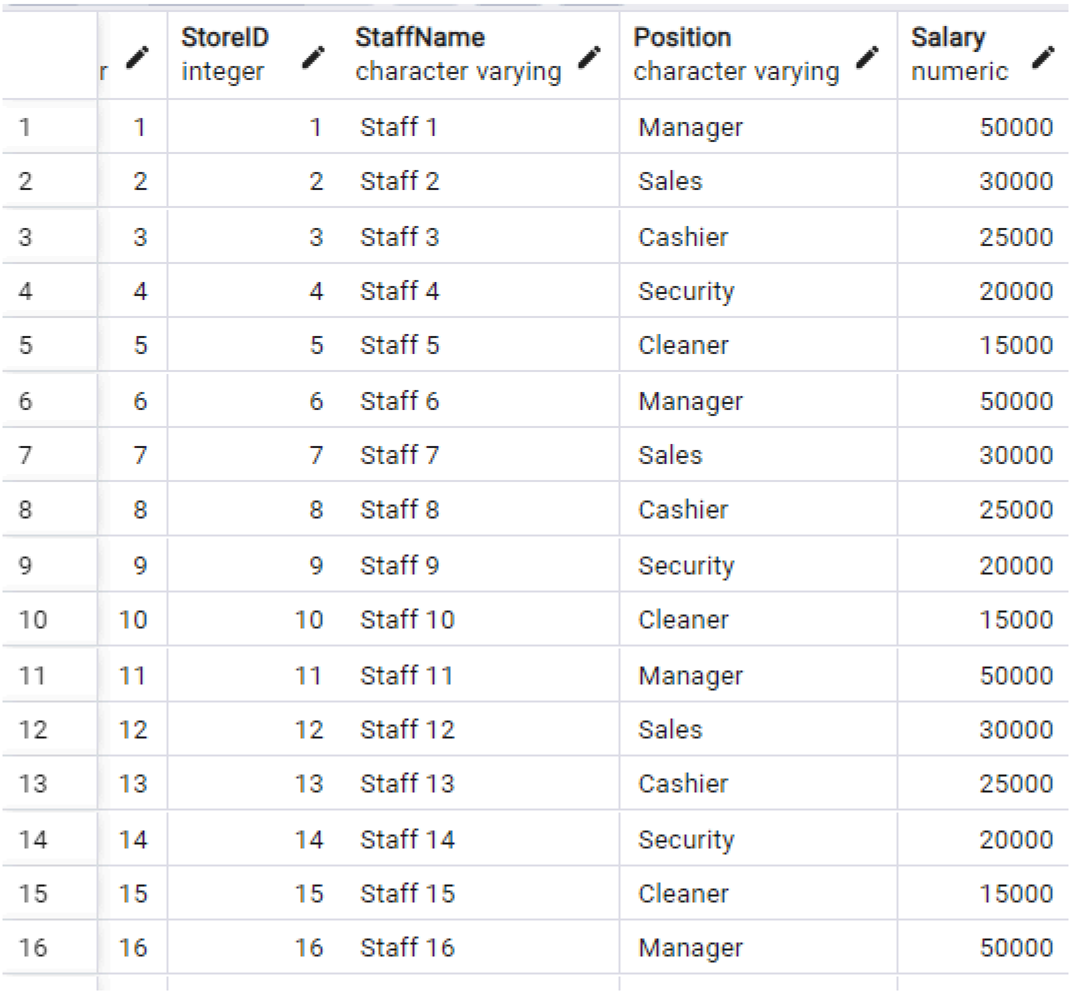
```
('Allegory'),
('Satire'),
('Fiction'),
('Sci-fi'),
('Adventure'),
('Quest'),
('Gothic Fiction'),
('Philosophical Fiction'),
('Novella'),
('Modernist'),
('Epic'),
('Psychological'),
('Dystopian'),
('Magical Realism'),
('Poetry'),
('True Crime'),
('Memoir'),
('Travel'),
('Horror'),
('Graphic Novels');

-- Insert data into "Language" table
INSERT INTO public."Language" ("LanguageCode", "LanguageName") VALUES
('EN', 'English'),
('FR', 'French'),
('ES', 'Spanish'),
('DE', 'German'),
('IT', 'Italian'),
('RU', 'Russian'),
('CN', 'Chinese'),
('JP', 'Japanese'),
('KR', 'Korean'),
('AR', 'Arabic'),
('PT', 'Portuguese'),
('HI', 'Hindi'),
('BN', 'Bengali'),
('PA', 'Punjabi'),
('JA', 'Javanese'),
('TE', 'Telugu'),
('VI', 'Vietnamese'),
('KO', 'Korean'),
('FA', 'Persian'),
('PL', 'Polish');

-- Insert data into "Condition" table
INSERT INTO public."Condition" ("ConditionName") VALUES
('New'),
('Like New'),
('Very Good'),
('Good'),
('Acceptable'),
('Poor'),
('Very Poor'),
('Damaged'),
('Repaired'),
('Refurbished'),
('New Condition'),
('Slightly Used'),
('Moderate Wear'),
('Heavy Wear'),
('Functional'),
('Non-functional'),
('Collectible'),
('Rare'),
('Common'),
('Unique');

-- Insert data into "Book" table
```

```
INSERT INTO public."Book" ("BookName", "ISBN", "PubYear", "Pages", "BookPrice", "PublisherID",
"LanguageID", "ConditionName") VALUES
('The Sound and the Fury', 'ISBN001', 2000, 300, 19.99, 1, 1, 'New'),
('Lord of the Flies', 'ISBN002', 2001, 320, 9.99, 2, 2, 'Like New'),
('Sense and Sensibility', 'ISBN003', 2002, 340, 14.99, 3, 3, 'Very Good'),
('The Adventures of Huckleberry Finn', 'ISBN004', 2003, 360, 12.99, 4, 4, 'Good'),
('Les Misérables', 'ISBN005', 2004, 380, 22.99, 5, 5, 'Acceptable'),
('Version Up', 'ISBN006', 2005, 400, 18.99, 6, 6, 'Poor'),
('Devine All Love Live', 'ISBN007', 2006, 420, 16.99, 7, 7, 'Very Poor'),
('Don Quixote', 'ISBN008', 2007, 440, 20.99, 8, 8, 'Damaged'),
('Hitchhiker Guide To Heaven', 'ISBN009', 2008, 460, 24.99, 9, 9, 'Repaired'),
('Butterfly Effect', 'ISBN010', 2009, 480, 17.99, 10, 10, 'Refurbished'),
('The Howl', 'ISBN011', 2010, 500, 19.99, 11, 11, 'New Condition'),
('Plastic Candy', 'ISBN012', 2011, 520, 9.99, 12, 12, 'Slightly Used'),
('My Secret Playlist', 'ISBN013', 2012, 540, 14.99, 13, 13, 'Moderate Wear'),
('Interpretation of Dreams', 'ISBN014', 2013, 560, 12.99, 14, 14, 'Heavy Wear'),
('Thousand stars thousand dreams', 'ISBN015', 2014, 580, 22.99, 15, 15, 'Functional'),
('Night Aviation', 'ISBN016', 2015, 600, 18.99, 16, 16, 'Non-functional'),
('The Apocalypse', 'ISBN017', 2016, 620, 16.99, 17, 17, 'Collectible'),
('Park in the Night', 'ISBN018', 2017, 640, 20.99, 18, 18, 'Rare'),
('Rollercoaster', 'ISBN019', 2018, 660, 24.99, 19, 19, 'Common'),
('Violet', 'ISBN020', 2019, 680, 17.99, 20, 20, 'Unique');

-- Insert data into "CustomerAcc" table
INSERT INTO public."CustomerAcc" ("Name", "Username", "Password", "Address") VALUES
('Customer 1', 'username1', 'password1', 'Address 1'),
('Customer 2', 'username2', 'password2', 'Address 2'),
('Customer 3', 'username3', 'password3', 'Address 3'),
('Customer 4', 'username4', 'password4', 'Address 4'),
('Customer 5', 'username5', 'password5', 'Address 5'),
('Customer 6', 'username6', 'password6', 'Address 6'),
('Customer 7', 'username7', 'password7', 'Address 7'),
('Customer 8', 'username8', 'password8', 'Address 8'),
('Customer 9', 'username9', 'password9', 'Address 9'),
('Customer 10', 'username10', 'password10', 'Address 10'),
('Customer 11', 'username11', 'password11', 'Address 11'),
('Customer 12', 'username12', 'password12', 'Address 12'),
('Customer 13', 'username13', 'password13', 'Address 13'),
('Customer 14', 'username14', 'password14', 'Address 14'),
('Customer 15', 'username15', 'password15', 'Address 15'),
('Customer 16', 'username16', 'password16', 'Address 16'),
('Customer 17', 'username17', 'password17', 'Address 17'),
('Customer 18', 'username18', 'password18', 'Address 18'),
('Customer 19', 'username19', 'password19', 'Address 19'),
('Customer 20', 'username20', 'password20', 'Address 20');

-- Insert data into "Store" table
INSERT INTO public."Store" ("StoreName", "Address", "CityLoc", "CountryLoc", "Telephone")
VALUES
('Store 1', 'Store Address 1', 'City 1', 'Country 1', '111-111-1111'),
('Store 2', 'Store Address 2', 'City 2', 'Country 2', '222-222-2222'),
('Store 3', 'Store Address 3', 'City 3', 'Country 3', '333-333-3333'),
('Store 4', 'Store Address 4', 'City 4', 'Country 4', '444-444-4444'),
('Store 5', 'Store Address 5', 'City 5', 'Country 5', '555-555-5555'),
('Store 6', 'Store Address 6', 'City 6', 'Country 6', '666-666-6666'),
('Store 7', 'Store Address 7', 'City 7', 'Country 7', '777-777-7777'),
('Store 8', 'Store Address 8', 'City 8', 'Country 8', '888-888-8888'),
('Store 9', 'Store Address 9', 'City 9', 'Country 9', '999-999-9999'),
('Store 10', 'Store Address 10', 'City 10', 'Country 10', '101-101-1010'),
('Store 11', 'Store Address 11', 'City 11', 'Country 11', '111-111-1111'),
('Store 12', 'Store Address 12', 'City 12', 'Country 12', '222-222-2222'),
('Store 13', 'Store Address 13', 'City 13', 'Country 13', '333-333-3333'),
('Store 14', 'Store Address 14', 'City 14', 'Country 14', '444-444-4444'),
('Store 15', 'Store Address 15', 'City 15', 'Country 15', '555-555-5555'),
('Store 16', 'Store Address 16', 'City 16', 'Country 16', '666-666-6666'),
('Store 17', 'Store Address 17', 'City 17', 'Country 17', '777-777-7777'),
('Store 18', 'Store Address 18', 'City 18', 'Country 18', '888-888-8888'),
('Store 19', 'Store Address 19', 'City 19', 'Country 19', '999-999-9999'),
('Store 20', 'Store Address 20', 'City 20', 'Country 20', '101-101-1010');

-- Insert data into "Inventory" table
INSERT INTO public."Inventory" ("BookID", "StoreID", "Stock") VALUES
(1, 1, 10),
(2, 2, 20),
(3, 3, 30),
(4, 4, 40),
(5, 5, 50),
(6, 6, 60),
(7, 7, 70),
(8, 8, 80),
(9, 9, 90),
(10, 10, 100),
(11, 11, 110),
(12, 12, 120),
(13, 13, 130),
(14, 14, 140),
(15, 15, 150),
(16, 16, 160),
(17, 17, 170),
(18, 18, 180),
(19, 19, 190),
(20, 20, 200);

-- Insert data into "Staff" table
INSERT INTO public."Staff" ("StoreID", "StaffName", "Position", "Salary") VALUES
(1, 'Staff 1', 'Manager', 50000),
(2, 'Staff 2', 'Sales', 30000),
(3, 'Staff 3', 'Cashier', 25000),
(4, 'Staff 4', 'Security', 20000),
(5, 'Staff 5', 'Cleaner', 15000),
(6, 'Staff 6', 'Manager', 50000),
(7, 'Staff 7', 'Sales', 30000),
(8, 'Staff 8', 'Cashier', 25000),
(9, 'Staff 9', 'Security', 20000),
(10, 'Staff 10', 'Cleaner', 15000),
(11, 'Staff 11', 'Manager', 50000),
(12, 'Staff 12', 'Sales', 30000),
(13, 'Staff 13', 'Cashier', 25000),
(14, 'Staff 14', 'Security', 20000),
(15, 'Staff 15', 'Cleaner', 15000),
(16, 'Staff 16', 'Manager', 50000),
(17, 'Staff 17', 'Sales', 30000),
(18, 'Staff 18', 'Cashier', 25000),
(19, 'Staff 19', 'Security', 20000),
(20, 'Staff 20', 'Cleaner', 15000);

-- Insert data into "Review" table
INSERT INTO public."Review" ("BookID", "CustomerID", "RevText") VALUES
(1, 1, 'Review for Book 1'),
(2, 2, 'Review for Book 2'),
(3, 3, 'Review for Book 3'),
(4, 4, 'Review for Book 4'),
(5, 5, 'Review for Book 5'),
(6, 6, 'Review for Book 6'),
(7, 7, 'Review for Book 7'),
(8, 8, 'Review for Book 8'),
(9, 9, 'Review for Book 9'),
(10, 10, 'Review for Book 10'),
(11, 11, 'Review for Book 11'),
(12, 12, 'Review for Book 12'),
(13, 13, 'Review for Book 13'),
(14, 14, 'Review for Book 14'),
(15, 15, 'Review for Book 15'),
(16, 16, 'Review for Book 16'),
(17, 17, 'Review for Book 17'),
(18, 18, 'Review for Book 18'),
(19, 19, 'Review for Book 19'),
(20, 20, 'Review for Book 20');

-- Insert data into "Wishlist" table
INSERT INTO public."Wishlist" ("CustomerID", "CreatedTime") VALUES
(1, '2024-05-31 00:00:00+00'),
(2, '2024-05-31 01:00:00+00'),
(3, '2024-05-31 02:00:00+00'),
(4, '2024-05-31 03:00:00+00'),
(5, '2024-05-31 04:00:00+00'),
(6, '2024-05-31 05:00:00+00'),
(7, '2024-05-31 06:00:00+00'),
(8, '2024-05-31 07:00:00+00'),
(9, '2024-05-31 08:00:00+00'),
(10, '2024-05-31 09:00:00+00'),
(11, '2024-05-31 10:00:00+00'),
(12, '2024-05-31 11:00:00+00'),
(13, '2024-05-31 12:00:00+00'),
(14, '2024-05-31 13:00:00+00'),
(15, '2024-05-31 14:00:00+00'),
(16, '2024-05-31 15:00:00+00'),
(17, '2024-05-31 16:00:00+00'),
(18, '2024-05-31 17:00:00+00'),
(19, '2024-05-31 18:00:00+00'),
(20, '2024-05-31 19:00:00+00');

-- Insert data into "Wishlist_Book" table
INSERT INTO public."Wishlist_Book" ("WishlistID", "BookID") VALUES
(1, 1),
(1, 2),
(1, 3),
(1, 4),
(1, 5),
(2, 6),
(2, 7),
(2, 8),
(2, 9),
(2, 10),
(2, 11),
(3, 12),
(3, 13),
(3, 14),
(3, 15),
(4, 16),
(4, 17),
(4, 18),
(4, 19),
(4, 20);

COMMIT;
```

D. Transaction Control Language

TCL dapat digunakan untuk menyimpan perubahan yang dilakukan dalam transaksi menggunakan COMMIT

Sebelum Transaksi:

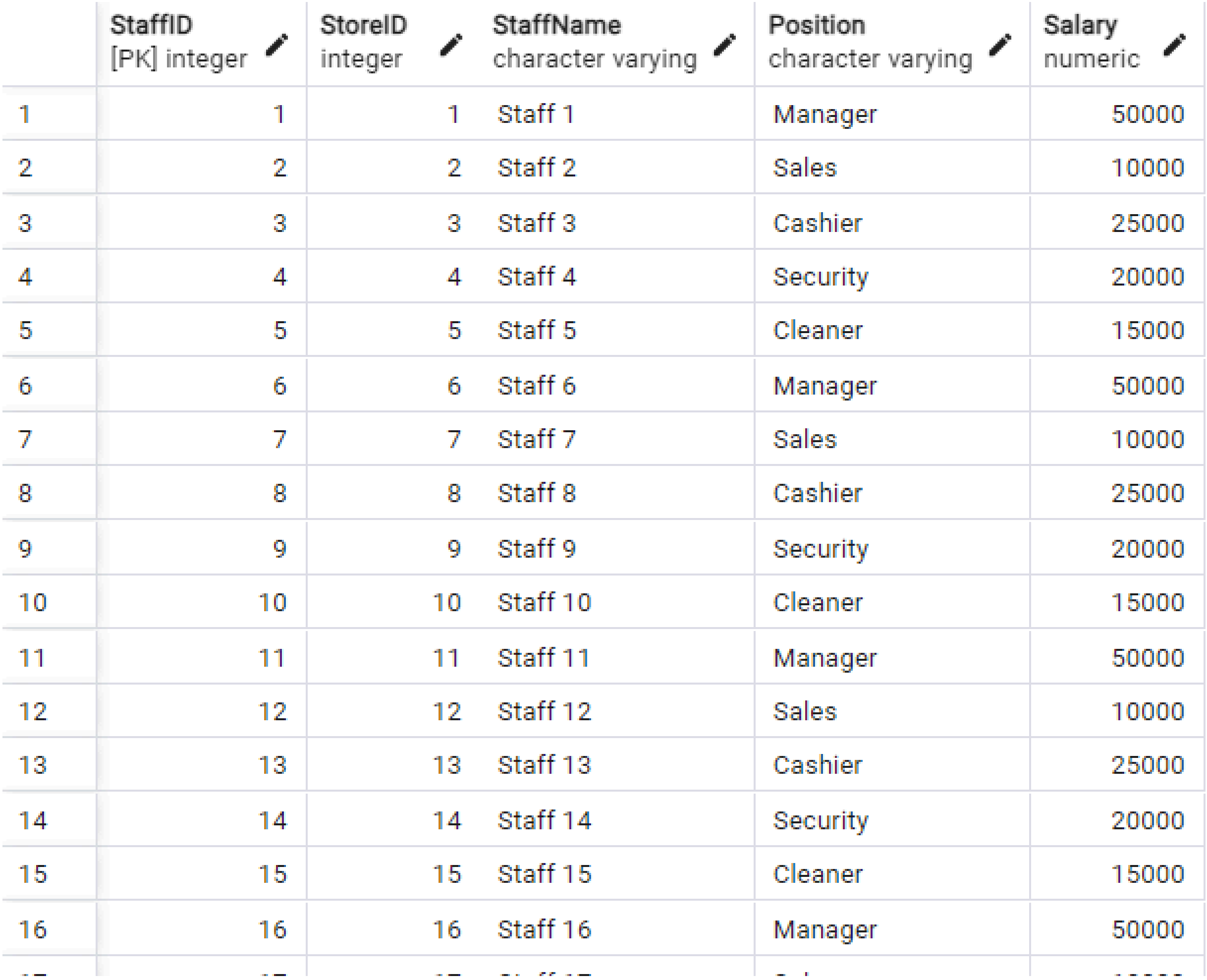


	r	StoreID integer	StaffName character varying	Position character varying	Salary numeric
1	1	1	Staff 1	Manager	50000
2	2	2	Staff 2	Sales	30000
3	3	3	Staff 3	Cashier	25000
4	4	4	Staff 4	Security	20000
5	5	5	Staff 5	Cleaner	15000
6	6	6	Staff 6	Manager	50000
7	7	7	Staff 7	Sales	30000
8	8	8	Staff 8	Cashier	25000
9	9	9	Staff 9	Security	20000
10	10	10	Staff 10	Cleaner	15000
11	11	11	Staff 11	Manager	50000
12	12	12	Staff 12	Sales	30000
13	13	13	Staff 13	Cashier	25000
14	14	14	Staff 14	Security	20000
15	15	15	Staff 15	Cleaner	15000
16	16	16	Staff 16	Manager	50000

SQL:

```
BEGIN TRANSACTION;
UPDATE "Staff" SET "Salary" = 10000 WHERE "Position"="Cashier";
COMMIT
```

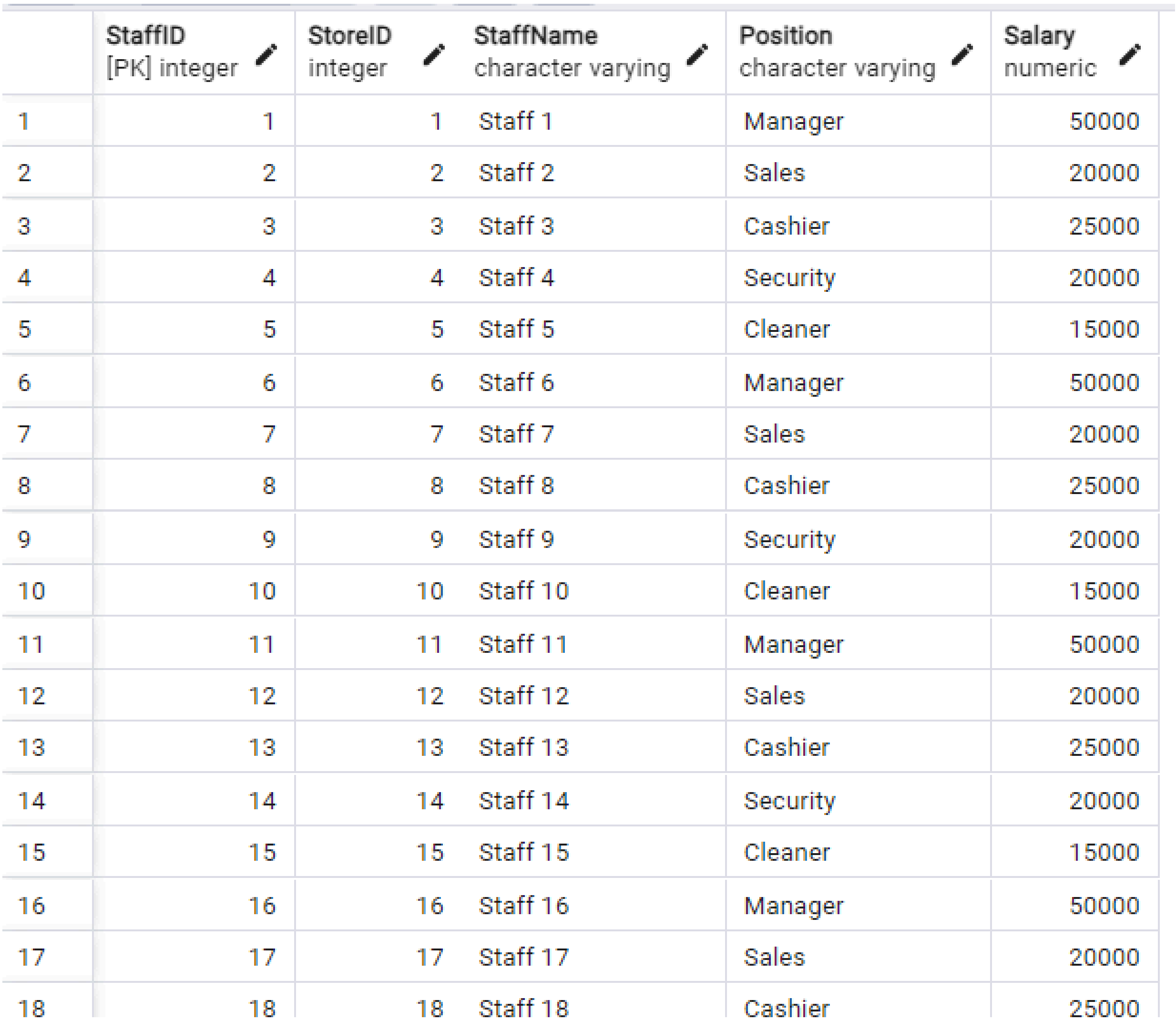
Setelah Transaksi:



	StaffID [PK] integer	StoreID integer	StaffName character varying	Position character varying	Salary numeric
1	1	1	Staff 1	Manager	50000
2	2	2	Staff 2	Sales	10000
3	3	3	Staff 3	Cashier	25000
4	4	4	Staff 4	Security	20000
5	5	5	Staff 5	Cleaner	15000
6	6	6	Staff 6	Manager	50000
7	7	7	Staff 7	Sales	10000
8	8	8	Staff 8	Cashier	25000
9	9	9	Staff 9	Security	20000
10	10	10	Staff 10	Cleaner	15000
11	11	11	Staff 11	Manager	50000
12	12	12	Staff 12	Sales	10000
13	13	13	Staff 13	Cashier	25000
14	14	14	Staff 14	Security	20000
15	15	15	Staff 15	Cleaner	15000
16	16	16	Staff 16	Manager	50000

ROLLBACK akan membatalkan perubahan tersebut

Sebelum:

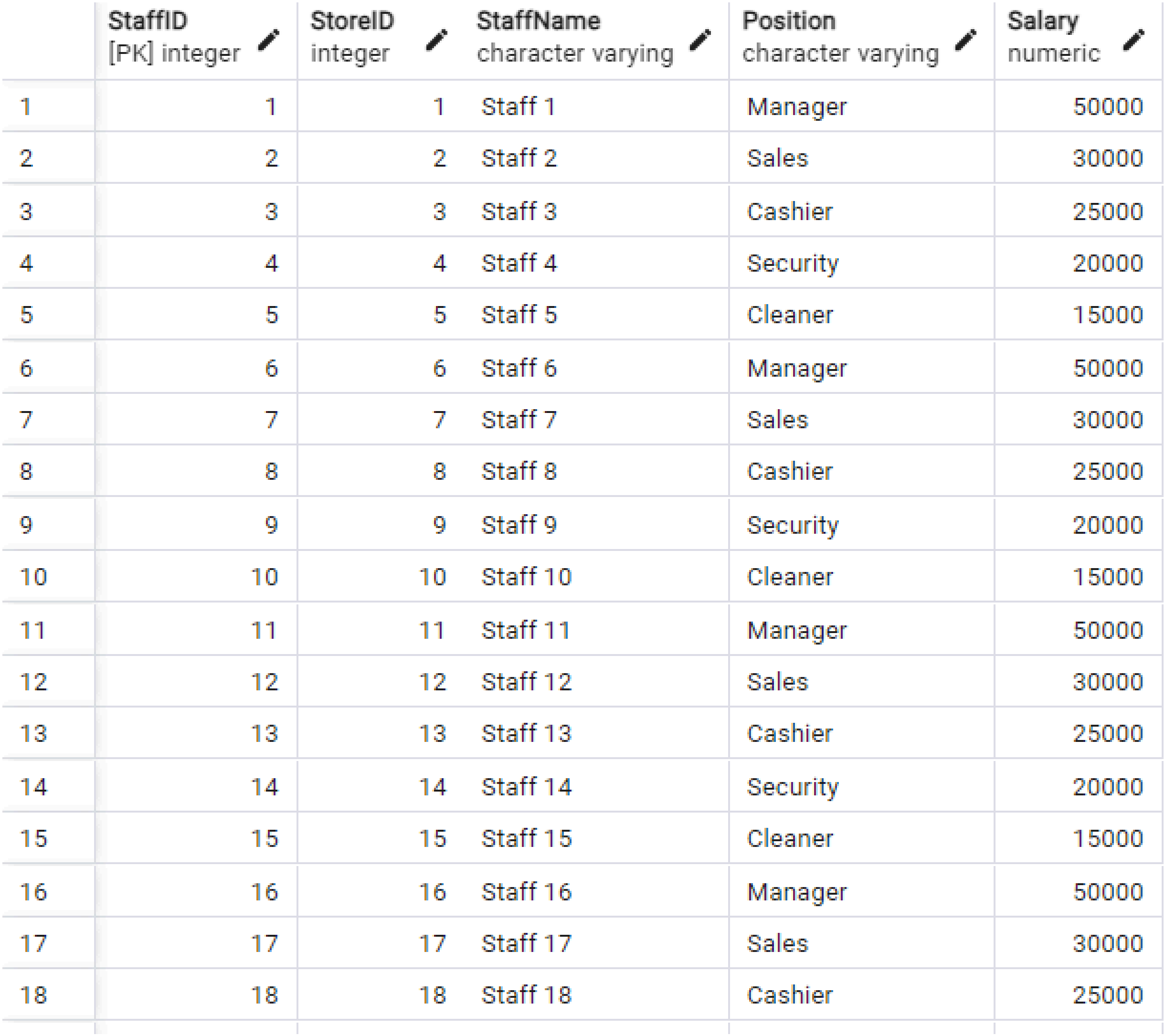


	StaffID [PK] integer	StoreID integer	StaffName character varying	Position character varying	Salary numeric
1	1	1	Staff 1	Manager	50000
2	2	2	Staff 2	Sales	20000
3	3	3	Staff 3	Cashier	25000
4	4	4	Staff 4	Security	20000
5	5	5	Staff 5	Cleaner	15000
6	6	6	Staff 6	Manager	50000
7	7	7	Staff 7	Sales	20000
8	8	8	Staff 8	Cashier	25000
9	9	9	Staff 9	Security	20000
10	10	10	Staff 10	Cleaner	15000
11	11	11	Staff 11	Manager	50000
12	12	12	Staff 12	Sales	20000
13	13	13	Staff 13	Cashier	25000
14	14	14	Staff 14	Security	20000
15	15	15	Staff 15	Cleaner	15000
16	16	16	Staff 16	Manager	50000
17	17	17	Staff 17	Sales	20000
18	18	18	Staff 18	Cashier	25000

SQL

```
BEGIN TRANSACTION;
UPDATE "Staff" SET "Salary" = 20000 WHERE "Position"='Sales';
ROLLBACK;
```

sesudah



	StaffID [PK] integer	StoreID integer	StaffName character varying	Position character varying	Salary numeric
1	1	1	Staff 1	Manager	50000
2	2	2	Staff 2	Sales	30000
3	3	3	Staff 3	Cashier	25000
4	4	4	Staff 4	Security	20000
5	5	5	Staff 5	Cleaner	15000
6	6	6	Staff 6	Manager	50000
7	7	7	Staff 7	Sales	30000
8	8	8	Staff 8	Cashier	25000
9	9	9	Staff 9	Security	20000
10	10	10	Staff 10	Cleaner	15000
11	11	11	Staff 11	Manager	50000
12	12	12	Staff 12	Sales	30000
13	13	13	Staff 13	Cashier	25000
14	14	14	Staff 14	Security	20000
15	15	15	Staff 15	Cleaner	15000
16	16	16	Staff 16	Manager	50000
17	17	17	Staff 17	Sales	30000
18	18	18	Staff 18	Cashier	25000