1. CREATE TABLE Cust\_User (

user\_id INT PRIMARY KEY,

username VARCHAR(50),

email VARCHAR(100),

password VARCHAR(100),

profile\_id INT

);

CREATE TABLE Profile (

profile\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

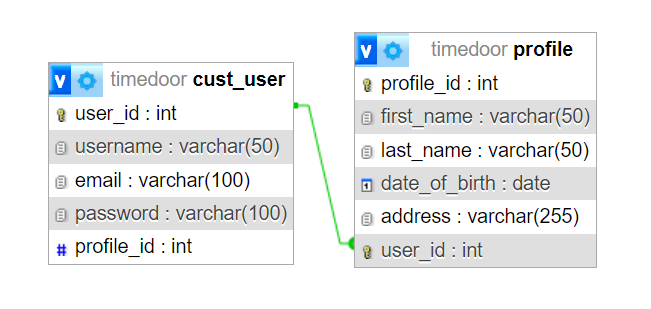
date\_of\_birth DATE,

address VARCHAR(255),

user\_id INT UNIQUE,

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

);



1. INSERT INTO Cust\_User (user\_id, username, email, password, profile\_id)

VALUES

(1, 'joni\_123', 'joni123@contoh.com', 'sandi123', 1),

(2, 'lina\_456', 'lina456@contoh.com', 'sandi456', 2);

INSERT INTO Profile (profile\_id, first\_name, last\_name, date\_of\_birth, address, user\_id)

VALUES

(1, 'Joni', 'Santoso', '1995-10-20', 'Jl. Jendral Sudirman No. 123', 1),

(2, 'Lina', 'Wijaya', '1990-03-15', 'Jl. Gatot Subroto No. 456', 2);

1. Untuk menetapkan hubungan one-to-one antara tabel User dan Profile menggunakan foreign key, saya harus memilih satu dari tabel tersebut sebagai tabel "induk" atau "utama" yang akan memiliki PK, dan tabel lainnya akan memiliki FK yang merujuk ke kunci primer dari tabel utama. Sebagai contoh di dalam tabel cust\_user terdapat field user\_id sebagai primary key. User\_id ini akan berelasi dengan field user\_id yang ada di tabel profil.
2. CREATE TABLE Category (

category\_id INT PRIMARY KEY,

category\_name VARCHAR(100)

);

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

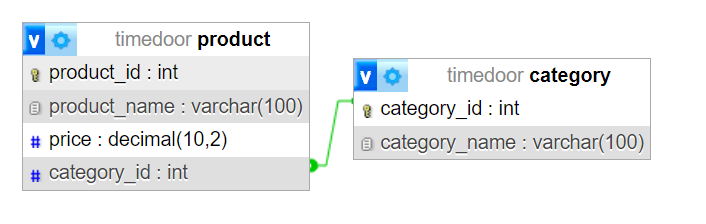
product\_name VARCHAR(100),

price DECIMAL(10, 2),

category\_id INT,

FOREIGN KEY (category\_id) REFERENCES Category(category\_id)

);



INSERT INTO Category (category\_id, category\_name)

VALUES

(1, 'Elektronik'),

(2, 'Pakaian'),

(3, 'Makanan');

INSERT INTO Product (product\_id, product\_name, price, category\_id)

VALUES

(1, 'Laptop', 1200.00, 1),

(2, 'Smartphone', 800.00, 1),

(3, 'T-Shirt', 20.00, 2),

(4, 'Jeans', 50.00, 2),

(5, 'Snack', 5.00, 3),

(6, 'Minuman', 2.50, 3);

1. Untuk menghapus kategori dari tabel Category tanpa kehilangan produk terkait dalam tabel Product adalah memeriksa apakah ada produk yang langsung terkait dengan kategori yang akan dihapus. Jika ada, produk tersebut perlu dipindahkan ke kategori lain atau dihapus terlebih dahulu. Penggunaan kunci asing memastikan bahwa aturan referensial diterapkan, sehingga tidak memungkinkan penghapusan kategori jika ada produk yang masih terkait dengannya.
2. CREATE TABLE Student (

student\_id INT PRIMARY KEY,

student\_name VARCHAR(100)

);

CREATE TABLE Course (

course\_id INT PRIMARY KEY,

course\_name VARCHAR(100)

);

CREATE TABLE Student\_Course (

enrollment\_id INT PRIMARY KEY,

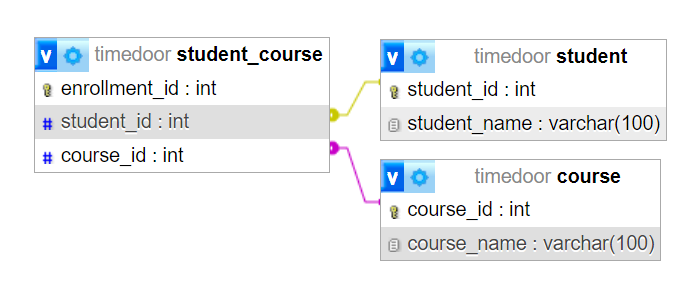
student\_id INT,

course\_id INT,

FOREIGN KEY (student\_id) REFERENCES Student(student\_id),

FOREIGN KEY (course\_id) REFERENCES Course(course\_id)

);



INSERT INTO Student (student\_id, student\_name)

VALUES

(1, 'Ani Wijaya'),

(2, 'Budi Santoso'),

(3, 'Citra Rahayu');

INSERT INTO Course (course\_id, course\_name)

VALUES

(1, 'Matematika'),

(2, 'Ilmu Pengetahuan Alam'),

(3, 'Sejarah');

1. Untuk menetapkan hubungan many-to-many antara tabel Student dan Course, digunakan tabel perantara "student\_course". Pertama, dibuat tabel Student dengan kunci primer student\_id, dan tabel Course dengan kunci primer course\_id. Kemudian, dibuat tabel perantara student\_course dengan kunci primer enrollment\_id serta dua kunci asing: student\_id yang merujuk ke tabel Student, dan course\_id yang merujuk ke tabel Course. Dengan struktur ini, setiap baris di tabel perantara mencerminkan hubungan antara satu siswa dan satu kursus.
2. CREATE TABLE Author (

author\_id INT PRIMARY KEY,

author\_name VARCHAR(100)

);

CREATE TABLE Publisher (

publisher\_id INT PRIMARY KEY,

publisher\_name VARCHAR(100)

);

CREATE TABLE Book (

book\_id INT PRIMARY KEY,

book\_title VARCHAR(100),

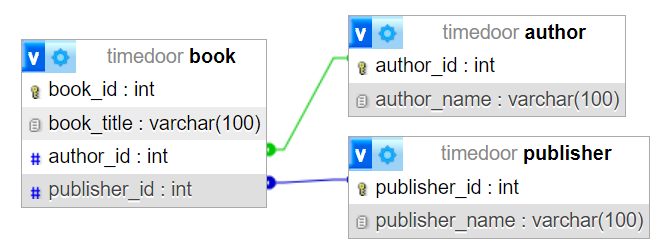
author\_id INT,

publisher\_id INT,

FOREIGN KEY (author\_id) REFERENCES Author(author\_id),

FOREIGN KEY (publisher\_id) REFERENCES Publisher(publisher\_id)

);



1. INSERT INTO Author (author\_id, author\_name)

VALUES

(1, 'Ahmad Setiawan'),

(2, 'Budi Utomo'),

(3, 'Citra Dewi');

INSERT INTO Publisher (publisher\_id, publisher\_name)

VALUES

(1, 'Penerbit Maju Jaya'),

(2, 'Penerbit Cahaya Ilmu'),

(3, 'Penerbit Cemerlang');

INSERT INTO Book (book\_title, author\_id, publisher\_id)

VALUES

('Biologi SMA', 1, 1),

('Kimia SMA', 1, 1),

('Geografi SMA', 1, 1),

('Bahasa Inggris', 2, 2),

('Bahasa Indonesia', 2, 2),

('Pendidikan Agama Islam', 3, 3);

1. Untuk menampilkan buku-buku yang ditulis oleh seorang penulis tertentu, kita menggunakan pernyataan SQL yang menggabungkan tabel Book dan Author dengan operasi JOIN. Setelah itu, kita memfilter hasil dengan klausa WHERE berdasarkan nama penulis yang diinginkan. Sebagai contoh, dengan menentukan nama penulis ‘Citra Dewi’,

SELECT Book.book\_title, Author.author\_name, Publisher.publisher\_name

FROM Book

JOIN Author ON Book.author\_id = Author.author\_id

JOIN Publisher ON Book.publisher\_id = Publisher.publisher\_id

WHERE Author.author\_name = 'Citra Dewi';

Pernyataan SQL ini menampilkan informasi buku yang ditulis oleh penulis dengan nama 'Citra Dewi'. Menggunakan JOIN untuk menghubungkan tabel Book, Author, dan Publisher, kita mendapatkan judul buku (book\_title), nama penulis (author\_name), dan nama penerbit (publisher\_name). Klausa WHERE memastikan hanya buku-buku yang ditulis oleh 'Citra Dewi' yang ditampilkan.

