CREATE TABLE myemployees (

employee\_id SERIAL PRIMARY KEY,

firstname VARCHAR(32),

lastname VARCHAR(32),

title VARCHAR(32) NOT NULL DEFAULT '',

age INTEGER CHECK (age >= 0),

salary INTEGER CHECK (salary <= 100000)

);

--Câu 2

INSERT INTO myemployees (employee\_id, firstname, lastname, title, age, salary) VALUES

(1, 'Jonie', 'Weber', 'Secretary', 28, 19500),

(2, 'Potsy', 'Weber', 'Programmer', 32, 45300),

(3, 'Dirk', 'Smith', 'Programmer II', 45, 75020),

(4, 'Mike', 'Nicols', 'Programmer', 25, 35000),

(5, 'Jim', 'Smith', 'Secretary', 24, 17000),

(6, 'Dean', 'Yeager', 'Programmer II', 39, 73000),

(7, 'Mark', 'Middleton' ,NULL , 21, 10000);

--Câu 3

SELECT \*

FROM myemployees;

--Câu 4

SELECT employee\_id, firstname, lastname, salary

FROM myemployees

WHERE salary <30\_000;

--Câu 5

SELECT firstname, lastname

FROM myemployees

WHERE age <30;

--Câu 6

SELECT firstname, lastname, salary

FROM myemployees

WHERE title = 'Programmer';

--Câu 7

SELECT \*

FROM myemployees

WHERE lastname LIKE '%ebe%';

--Câu 8

SELECT \*

FROM myemployees

WHERE firstname = 'Potsy';

--Câu 9

SELECT \*

FROM myemployees

WHERE lastname LIKE '%ith';

--Câu 10

UPDATE myemployees

SET firstname = 'Jonie' , lastname = 'Williams'

WHERE firstname = 'Jonie' AND

lastname = 'Weber';

--Câu 11

UPDATE myemployees

SET age = age +1

WHERE firstname = 'Dirk' AND

lastname = 'Smith';

--Câu 12

UPDATE myemployees

SET title = 'Administrative Assistant'

WHERE title = 'Secretary';

--Câu 13

UPDATE myemployees

SET salary = salary + 3\_500

WHERE salary < 30\_000;

--Câu 14

UPDATE myemployees

SET salary = salary + 4\_500

WHERE salary > 33\_500;

--Câu 15

UPDATE myemployees

SET title = 'Programmer III'

WHERE title = 'Programmer II';

UPDATE myemployees

SET title = 'Programmer II'

WHERE title = 'Programmer I' ;

--Câu 16

DELETE FROM myemployees

WHERE firstname = 'Jonie' AND

lastname = 'Williams';

--Câu 17

DELETE FROM myemployees

WHERE salary > 70\_000;

--Câu 18

CREATE ROLE "" WITH LOGIN PASSWORD '';

CREATE DATABASE music

WITH OWNER = ""

ENCODING = 'UTF8';

--Câu 19

CREATE TABLE album (

Id SERIAL PRIMARY KEY,

Title VARCHAR(100)

);

CREATE TABLE artist (

Id SERIAL PRIMARY KEY,

Name VARCHAR(100)

);

CREATE TABLE track (

id SERIAL PRIMARY KEY,

title VARCHAR(100),

len INTEGER,

rating INTEGER CHECK (rating>0),

count INTEGER,

album\_id INTEGER REFERENCES album(id) ON DELETE CASCADE,

artist\_id INTEGER REFERENCES artist(id) ON DELETE CASCADE

);

--Câu 20

COPY album FROM 'C:/album.csv' DELIMITER ',' CSV HEADER;

COPY artist FROM 'C:/artist.csv' DELIMITER ',' CSV HEADER;

COPY track FROM 'C:/track.csv' DELIMITER ',' CSV HEADER;

--Câu 21

CREATE TABLE track\_raw (

id SERIAL PRIMARY KEY,

title VARCHAR(100),

name VARCHAR(100),

title\_album VARCHAR(100),

count INTEGER,

rating INTEGER CHECK (rating > 0),

len INTEGER

);

COPY track\_raw(title, name, title\_album, count, rating, len)

FROM 'C:/track\_raw.csv' DELIMITER ',' CSV;

-- import data table album

INSERT INTO album (title)

SELECT DISTINCT title\_album FROM track\_raw;

-- import data table artist

INSERT INTO artist (name)

SELECT DISTINCT name FROM track\_raw;

--import data table track

INSERT INTO track (title, len, rating, count, album\_id, artist\_id)

SELECT

t.title, t.len, t.rating, t.count,

a.id AS album\_id,

ar.id AS artist\_id

FROM track\_raw t

JOIN album a ON t.title\_album = a.title

JOIN artist ar ON t.name = ar.name;