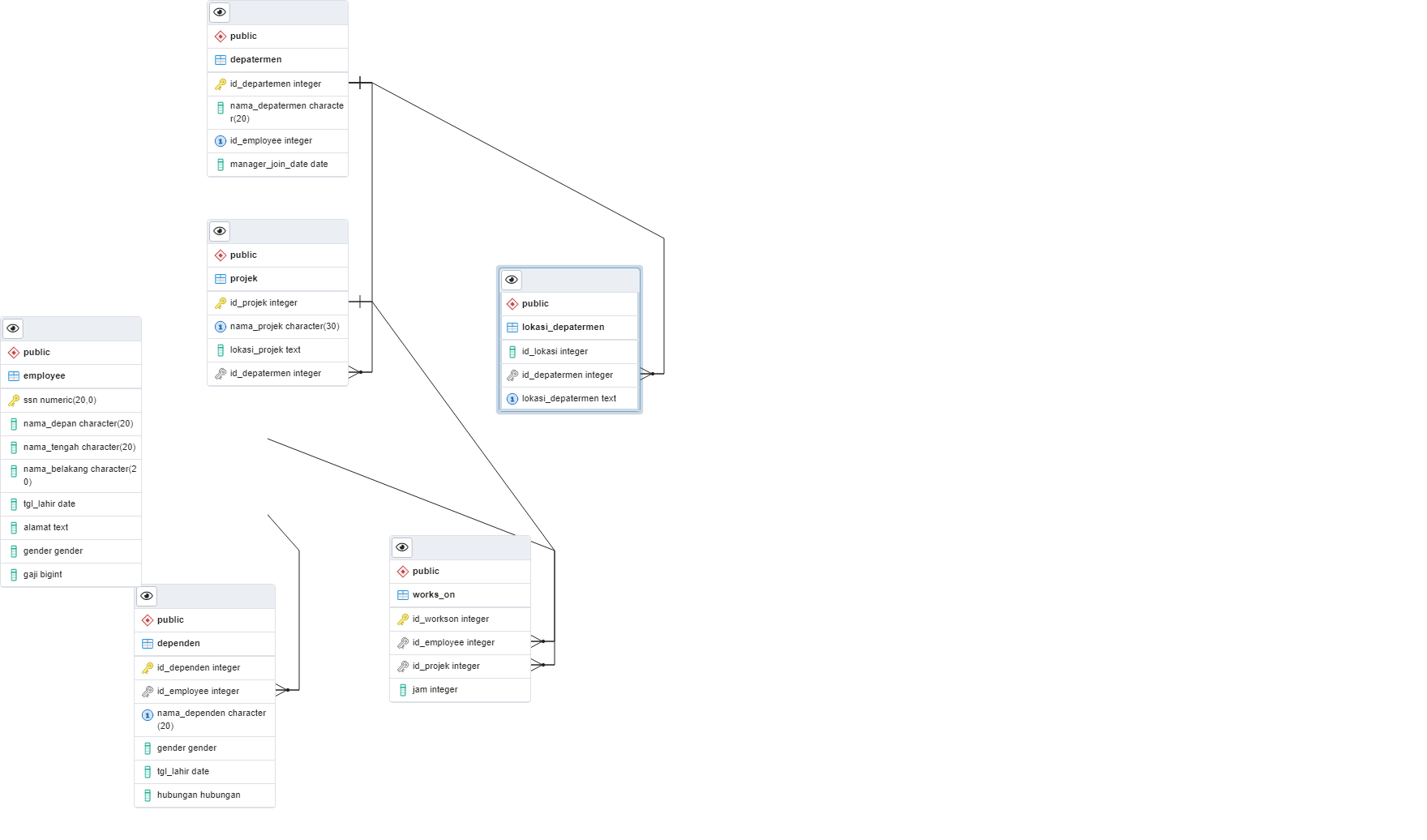
**DB\_COMPANY (PostgreSQL)**

****

Gambar 1. DB\_COMPANY (ERD)

**1. CREATE SCRIPT**

**1. depatermen**

-- Table: public.depatermen

-- DROP TABLE IF EXISTS public.depatermen;

CREATE TABLE IF NOT EXISTS public.depatermen

(

id\_departemen integer NOT NULL DEFAULT nextval('depatermen\_id\_departemen\_seq'::regclass),

nama\_depatermen character(20) COLLATE pg\_catalog."default",

id\_employee integer,

manager\_join\_date date,

CONSTRAINT depatermen\_pkey PRIMARY KEY (id\_departemen),

CONSTRAINT uq\_employee UNIQUE (id\_employee)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.depatermen

OWNER to postgres;

**2. dependen**

-- Table: public.dependen

-- DROP TABLE IF EXISTS public.dependen;

CREATE TABLE IF NOT EXISTS public.dependen

(

id\_dependen integer NOT NULL DEFAULT nextval('dependent\_id\_dependent\_seq'::regclass),

id\_employee integer,

nama\_dependen character(20) COLLATE pg\_catalog."default",

gender gender,

tgl\_lahir date,

hubungan hubungan,

CONSTRAINT dependent\_pkey PRIMARY KEY (id\_dependen),

CONSTRAINT uq\_nama\_dependen UNIQUE (nama\_dependen),

CONSTRAINT fk\_employee FOREIGN KEY (id\_employee)

REFERENCES public.employee (ssn) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.dependen

OWNER to postgres;

**3. employee**

-- Table: public.employee

-- DROP TABLE IF EXISTS public.employee;

CREATE TABLE IF NOT EXISTS public.employee

(

ssn numeric(20,0) NOT NULL,

nama\_depan character(20) COLLATE pg\_catalog."default" NOT NULL,

nama\_tengah character(20) COLLATE pg\_catalog."default",

nama\_belakang character(20) COLLATE pg\_catalog."default",

tgl\_lahir date,

alamat text COLLATE pg\_catalog."default",

gender gender,

gaji bigint,

CONSTRAINT employee\_pkey PRIMARY KEY (ssn)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.employee

OWNER to postgres;

**4. lokasi departemen**

-- Table: public.lokasi\_depatermen

-- DROP TABLE IF EXISTS public.lokasi\_depatermen;

CREATE TABLE IF NOT EXISTS public.lokasi\_depatermen

(

id\_lokasi integer NOT NULL,

id\_depatermen integer NOT NULL,

lokasi\_depatermen text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT uq\_lokasi\_depatermen UNIQUE (lokasi\_depatermen),

CONSTRAINT fk\_depatermen FOREIGN KEY (id\_depatermen)

REFERENCES public.depatermen (id\_departemen) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.lokasi\_depatermen

OWNER to postgres;

**5. projek**

-- Table: public.projek

-- DROP TABLE IF EXISTS public.projek;

CREATE TABLE IF NOT EXISTS public.projek

(

id\_projek integer NOT NULL DEFAULT nextval('projek\_id\_projek\_seq'::regclass),

nama\_projek character(30) COLLATE pg\_catalog."default",

lokasi\_projek text COLLATE pg\_catalog."default",

id\_depatermen integer,

CONSTRAINT projek\_pkey PRIMARY KEY (id\_projek),

CONSTRAINT uq\_nama\_projek UNIQUE (nama\_projek),

CONSTRAINT fk\_depatermen FOREIGN KEY (id\_depatermen)

REFERENCES public.depatermen (id\_departemen) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.projek

OWNER to postgres;

**6. works on**

-- Table: public.works\_on

-- DROP TABLE IF EXISTS public.works\_on;

CREATE TABLE IF NOT EXISTS public.works\_on

(

id\_workson integer NOT NULL,

id\_employee integer NOT NULL,

id\_projek integer NOT NULL,

jam integer NOT NULL,

CONSTRAINT works\_on\_pkey PRIMARY KEY (id\_workson),

CONSTRAINT uq\_together\_workson UNIQUE (id\_employee, id\_projek)

INCLUDE(id\_employee, id\_projek),

CONSTRAINT fk\_employee FOREIGN KEY (id\_employee)

REFERENCES public.employee (ssn) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_projek FOREIGN KEY (id\_projek)

REFERENCES public.projek (id\_projek) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.works\_on

OWNER to postgres;

**2. INSERT DATA**

insert into employee values

(1, 'chandra', 'jackson', 'aja', '2022-01-01', 'jln. perkantoran no.8', 'male', 536352),

(2, 'nami', '', 'aja', '2022-02-01', 'jln. perkantoran no.9', 'female', 236352),

(3, 'robin', '', '', '2022-03-01', 'jln. perkantoran no.10', 'female', 136352),

(4, 'usop', 'jackson', 'aja', '2022-04-01', 'jln. perkantoran no.11', 'male', 936352)

insert into dependen values

(1, 4, 'chandra', 'male', '2022-01-01', 'internal'),

(2, 4, 'mihawk', 'male', '2021-02-01', 'eksternal'),

(3, 2, 'law', 'male', '2020-01-01', 'eksternal'),

(4, 1, 'roger', 'male', '2019-01-01', 'internal')

(5, 1, 'leo', 'male', '2022-01-01', 'internal'),

(6, 3, 'kaido', 'male', '2021-02-01', 'eksternal'),

(7, 2, 'apoo', 'male', '2020-01-01', 'eksternal')

insert into depatermen values

(1, 'it', 34242, '2020-01-01'),

(2, 'human resource', 12242, '2021-01-01'),

(4, 'it', 84242, '2019-01-01')

insert into projek values

(6, 'jakarta-fest', 'jakarta selatan', 1),

(7, 'bandung-fest', 'bandung', 1),

(8, 'palembang-fest', 'palembang', 1),

(9, 'bali-fest', 'bali', 4),

(10, 'japan-fest', 'japan', 2)

insert into lokasi\_depatermen values

(1, 1, 'malaysia'),

(2, 4, 'indonesia'),

(3, 2, 'palembang')

Insert into works\_on values

(1, 2, 6, 20),

(2, 2, 7, 100),

(3, 2, 8, 300),

(4, 2, 9, 2),

(5, 7, 9, 2),

(6, 8, 1, 2000),

(7, 5, 6, 300),

(8, 5, 9, 2)

**3. QUERY**

3.A **Tampilkan dependent\_name dan relationship dengan employee yang namanya diawali huruf R ?**

select

d.nama\_dependen,   
e.nama\_depan from dependen as d

inner join employee as e on d.id\_employee = e.ssn

where e.nama\_depan LIKE 'r%';

3.B **Banyaknya employee yang mengerjakan project PNum = 6 ?**

SELECT COUNT(id\_employee) AS employee\_on\_p6

FROM works\_on

where id\_projek = 6;

3.C **Banyaknya employee yang memiliki salary lebih dari 900000?**

select

e.nama\_depan,

e.gaji

from employee as e

where e.gaji > 900000

3.D **Banyaknya project yang dikerjakan DNum =2 ?**

select count(p.id\_depatermen) as projek\_depatermen\_2

from projek as p

where p.id\_depatermen = 2

3.E **Hitung total dan rata-rata salary dari setiap departemen ?**

select

d.nama\_depatermen,

SUM(e.gaji) as total\_gaji,

AVG(gaji) as rata\_rata\_gaji from depatermen as d

inner join employee as e on d.id\_employee = e.ssn

group by d.nama\_depatermen

3.F **Banyaknya employee dari setiap department dan urutkan berdasarkan employee terbanyak ?**

select

d.nama\_depatermen,

COUNT(d.id\_employee) as total\_employee

from depatermen as d

inner join employee as e on d.id\_employee = e.ssn

group by d.nama\_depatermen

order by total\_employee DESC;

3.G **Total hours perweek dari semua employee untuk setiap project ?**

select

p.nama\_projek,

e.nama\_depan, SUM(w.jam) as total\_hours\_per\_week

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

inner join projek as p on w.id\_projek = p.id\_projek

group by p.nama\_projek, e.nama\_depan

3.H **Employee yang memiliki total hours perweek lebih besar dari 140 hours dan urutkan berdasarkan jumlah jam kerja terbanyak?**

select e.nama\_depan as employee,

p.nama\_projek as projek,

SUM(w.jam) as total\_jam

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

inner join projek as p on w.id\_projek = p.id\_projek

where w.jam >= 140

GROUP BY w.id\_employee, projek, employee, w.jam

ORDER BY w.jam DESC

3.I **Kelompokkan bonus employee berdasarkan jumlah jam kerjanya ? (Jika >= 200 hours, maka bonus = 50%; Jika >= 150 hours, maka bonus = 25%, Selainnya bonus = 10%)**

select e.nama\_depan as employee,

case

when SUM(jam) >= 200 then '50%'

when SUM(jam) >= 150 then '25%'

else '10%'

end as bonus

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

group by w.id\_employee, e.nama\_depan

3.J **Banyaknya project yang dikerjakan tiap employee dan urutkan dari yang terbanyak ?**

select

e.nama\_depan as employee,

COUNT(w.id\_projek) as total\_projek

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

group by w.id\_employee, e.nama\_depan

order by total\_projek DESC

3.K **Employee yang bekerja pada 4 project ?**

select

e.nama\_depan as employee,

COUNT(w.id\_projek) as total\_projek

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

group by w.id\_employee, e.nama\_depan

having COUNT(w.id\_projek) >= 4

3.L **Employee yang memiliki rata-rata hours perweek = 200 jam dan bekerja pada 1 project ?**

select

e.nama\_depan as employee,

w.jam,

COUNT(w.id\_projek) projek

from works\_on as w

inner join employee as e on w.id\_employee = e.ssn

group by e.nama\_depan, w.jam

HAVING w.jam>=200

AND COUNT(w.id\_projek) >= 1

3.M **Banyaknya Dependent berdasarkan relationship dengan employee ?**

select

count(\*) total\_dependen

from dependen as d

inner join employee as e on d.id\_dependen = e.ssn

3.N **Berapa lama Manager tiap Department sudah menjabat ?**

SELECT e.nama\_depan,

DATE\_PART('year', NOW()::date) -

DATE\_PART('year', d.manager\_join\_date::date)

as lama\_menjabat\_tahun

from depatermen as d

inner join employee as e on d.id\_employee = e.ssn

3.O **Lokasi project yang menjadi tempat lebih dari satu department?** select

d.lokasi\_depatermen as lokasi,

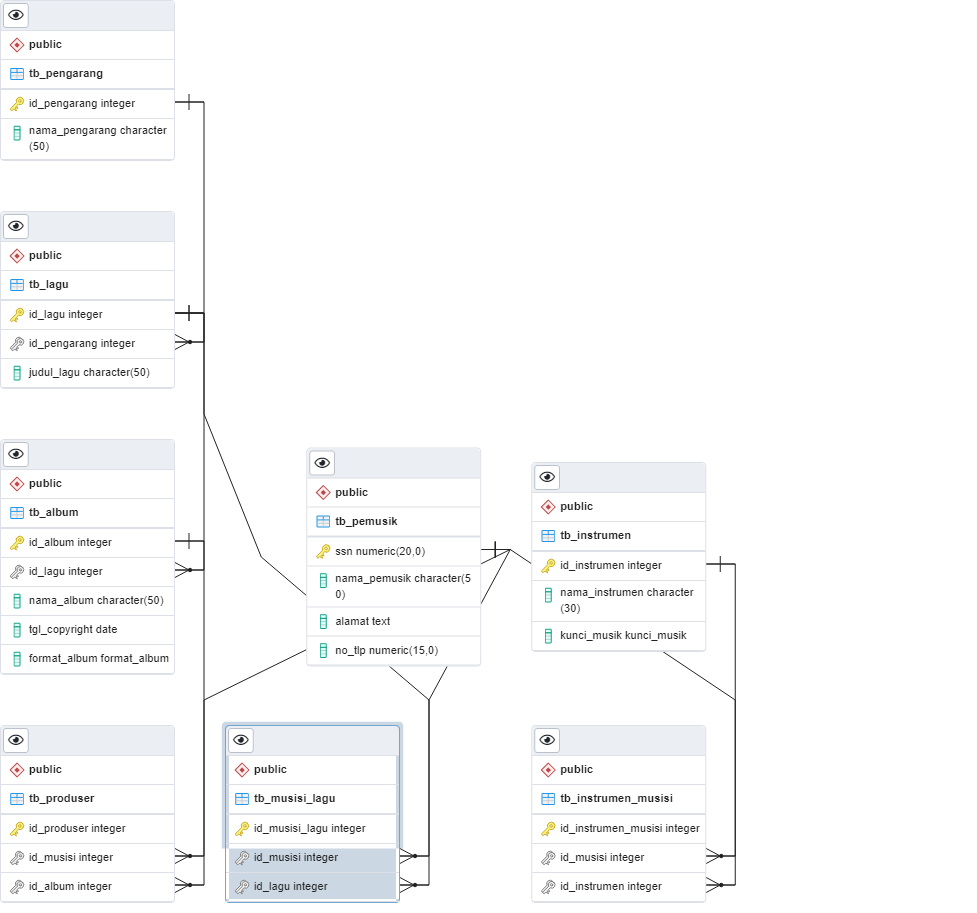
COUNT(d.id\_depatermen) as total\_departemen

from lokasi\_depatermen as d

group by lokasi

having COUNT(d.id\_depatermen) >= 1

**DB\_MELODI\_INDAH (PostgreSQL)**



Gambar 2. DB\_MELODI\_INDAH (ERD)

**1. CREATE SCRIPT**

**1.1 public.tb\_pemusik**

-- DROP TABLE IF EXISTS public.tb\_pemusik;

CREATE TABLE IF NOT EXISTS public.tb\_pemusik

(

ssn numeric(20,0) NOT NULL,

nama\_pemusik character(50) COLLATE pg\_catalog."default" NOT NULL,

alamat text COLLATE pg\_catalog."default",

no\_tlp numeric(15,0) NOT NULL,

CONSTRAINT tb\_pemusik\_pkey PRIMARY KEY (ssn)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_pemusik

OWNER to postgres;

**1.2 public.tb\_instrumen**

-- DROP TABLE IF EXISTS public.tb\_instrumen;

CREATE TABLE IF NOT EXISTS public.tb\_instrumen

(

id\_instrumen integer NOT NULL DEFAULT nextval('tb\_instrumen\_id\_instrumen\_seq'::regclass),

nama\_instrumen character(30) COLLATE pg\_catalog."default",

kunci\_musik kunci\_musik,

CONSTRAINT tb\_instrumen\_pkey PRIMARY KEY (id\_instrumen)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_instrumen

OWNER to postgres;

**1.3 public.tb\_pengarang**

-- DROP TABLE IF EXISTS public.tb\_pengarang;

CREATE TABLE IF NOT EXISTS public.tb\_pengarang

(

id\_pengarang integer NOT NULL DEFAULT nextval('tb\_pengarang\_id\_pengarang\_seq'::regclass),

nama\_pengarang character(50) COLLATE pg\_catalog."default",

CONSTRAINT tb\_pengarang\_pkey PRIMARY KEY (id\_pengarang)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_pengarang

OWNER to postgres;

**1.4 public.tb\_lagu**

-- DROP TABLE IF EXISTS public.tb\_lagu;

CREATE TABLE IF NOT EXISTS public.tb\_lagu

(

id\_lagu integer NOT NULL DEFAULT nextval('tb\_lagu\_id\_lagu\_seq'::regclass),

id\_pengarang integer,

judul\_lagu character(50) COLLATE pg\_catalog."default",

CONSTRAINT tb\_lagu\_pkey PRIMARY KEY (id\_lagu),

CONSTRAINT fk\_pengarang FOREIGN KEY (id\_pengarang)

REFERENCES public.tb\_pengarang (id\_pengarang) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_lagu

OWNER to postgres;

**1.5 public.tb\_album**

-- DROP TABLE IF EXISTS public.tb\_album;

CREATE TABLE IF NOT EXISTS public.tb\_album

(

id\_album integer NOT NULL DEFAULT nextval('tb\_album\_id\_album\_seq'::regclass),

id\_lagu integer NOT NULL,

nama\_album character(50) COLLATE pg\_catalog."default" NOT NULL,

tgl\_copyright date NOT NULL,

format\_album format\_album NOT NULL,

CONSTRAINT tb\_album\_pkey PRIMARY KEY (id\_album),

CONSTRAINT fk\_lagu FOREIGN KEY (id\_lagu)

REFERENCES public.tb\_lagu (id\_lagu) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_album

OWNER to postgres;

**1.6 public.tb\_instrumen\_musisi**

-- DROP TABLE IF EXISTS public.tb\_instrumen\_musisi;

CREATE TABLE IF NOT EXISTS public.tb\_instrumen\_musisi

(

id\_instrumen\_musisi integer NOT NULL DEFAULT nextval('tb\_instrumen\_musisi\_id\_instrumen\_musisi\_seq'::regclass),

id\_musisi integer,

id\_instrumen integer,

CONSTRAINT tb\_instrumen\_musisi\_pkey PRIMARY KEY (id\_instrumen\_musisi),

CONSTRAINT fk\_instrumen FOREIGN KEY (id\_instrumen)

REFERENCES public.tb\_instrumen (id\_instrumen) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_musisi FOREIGN KEY (id\_musisi)

REFERENCES public.tb\_pemusik (ssn) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_instrumen\_musisi

OWNER to postgres;

**1.7 public.tb\_musisi\_lagu**

-- DROP TABLE IF EXISTS public.tb\_musisi\_lagu;

CREATE TABLE IF NOT EXISTS public.tb\_musisi\_lagu

(

id\_musisi\_lagu integer NOT NULL DEFAULT nextval('tb\_musisi\_lagu\_id\_musisi\_lagu\_seq'::regclass),

id\_musisi integer NOT NULL,

id\_lagu integer NOT NULL,

CONSTRAINT tb\_musisi\_lagu\_pkey PRIMARY KEY (id\_musisi\_lagu),

CONSTRAINT fk\_lagu FOREIGN KEY (id\_lagu)

REFERENCES public.tb\_lagu (id\_lagu) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_musisi FOREIGN KEY (id\_musisi)

REFERENCES public.tb\_pemusik (ssn) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_musisi\_lagu

OWNER to postgres;

**1.8 public.tb\_produser**

-- DROP TABLE IF EXISTS public.tb\_produser;

CREATE TABLE IF NOT EXISTS public.tb\_produser

(

id\_produser integer NOT NULL DEFAULT nextval('tb\_produser\_id\_produser\_seq'::regclass),

id\_musisi integer NOT NULL,

id\_album integer NOT NULL,

CONSTRAINT tb\_produser\_pkey PRIMARY KEY (id\_produser),

CONSTRAINT fk\_album FOREIGN KEY (id\_album)

REFERENCES public.tb\_album (id\_album) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_musisi FOREIGN KEY (id\_musisi)

REFERENCES public.tb\_pemusik (ssn) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.tb\_produser

OWNER to postgres;

**2. INSERT DATA**

insert into tb\_pemusik values

(134658123, 'chandra', 'jln. ansi no 8', 23425643),

(234658167, 'nami', 'jln. sinsa no 9', 23455643),

(334652343, 'brook', 'jln. orange no 7', 8924643),

(434567663, 'usop', 'jln. malky no 7', 27825643),

(534634553, 'sanji', 'jln. ways no 6', 45725643)

insert into tb\_instrumen values

(1, 'gitar', 'C'),

(2, 'flute', 'A'),

(3, 'piano', 'B#')

insert into tb\_pengarang values

(1, 'jacob'),

(2, 'arial'),

(3, 'violet')

insert into tb\_lagu values

(1, 3, 'dark'),

(2, 1, 'versace on the floor'),

(3, 2, 'treasure')

insert into tb\_album values

(1, 1, 'album a', '2022-01-01', 'CD'),

(2, 1, 'album b', '2022-02-01', 'CD'),

(3, 2, 'album b', '2022-03-01', 'MC')

insert into tb\_instrumen\_musisi values

(1, 134658123, 2),

(2, 234658167, 1),

(3, 334652343, 3),

(4, 334652343, 1),

(5, 334652343, 2),

(6, 434567663, 1)

insert into tb\_musisi\_lagu values

(1, 234658167, 3),

(2, 234658167, 2),

(3, 234658167, 2),

(4, 234658167, 2)

insert into tb\_produser values

(234658167, 2),

(234658167, 2),

(334652343, 1),

(334652343, 3)

**3. QUERY**

**3.1 Judul lagu & Nama album**

select l.judul\_lagu, a.nama\_album, a.tgl\_copyright, a.format\_album

from tb\_album as a

left join tb\_lagu as l on a.id\_lagu = l.id\_lagu

**3.2. nama-nama instrumen**

select \* from tb\_instrumen

**3.3. nama-nama musisi**

select \* from tb\_pemusik

**3.4. nama-nama pengarang**

select \* from tb\_pengarang

**3.5. instrumen yang dipakai musisi**

select p.nama\_pemusik, i.nama\_instrumen from tb\_instrumen\_musisi as im

left join tb\_pemusik as p on im.id\_musisi = p.ssn

left join tb\_instrumen as i on im.id\_instrumen = i.id\_instrumen

**3.6. nama lagu dan pengarang**

select p.nama\_pengarang, l.judul\_lagu from tb\_lagu as l

left join tb\_pengarang as p on l.id\_pengarang = p.id\_pengarang

**3.7. nama musisi dan lagu**

select p.nama\_pemusik, l.judul\_lagu from tb\_musisi\_lagu as ml

left join tb\_pemusik as p on ml.id\_musisi = p.ssn

left join tb\_lagu as l on ml.id\_lagu = l.id\_lagu

**1.8. produser musik**

select p.id\_produser, z.nama\_pemusik, a.nama\_album from tb\_produser as p

left join tb\_pemusik as z on p.id\_musisi = z.ssn

left join tb\_album as a on p.id\_album = a.id\_album