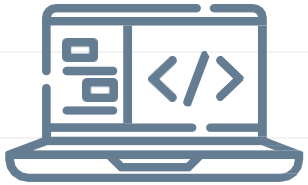
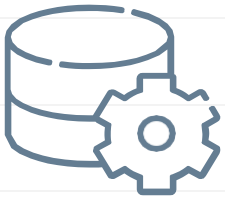
code academy



PostgreSQL

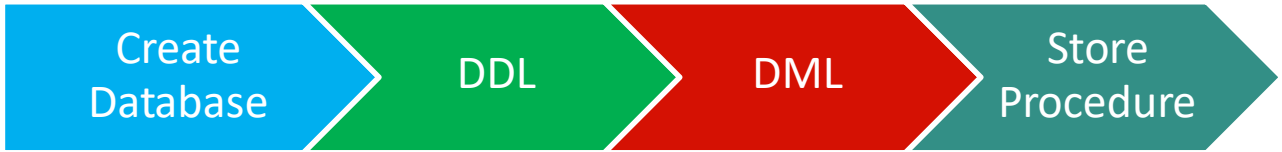
Kang Dian
Head Trainer Of CodeAcademy



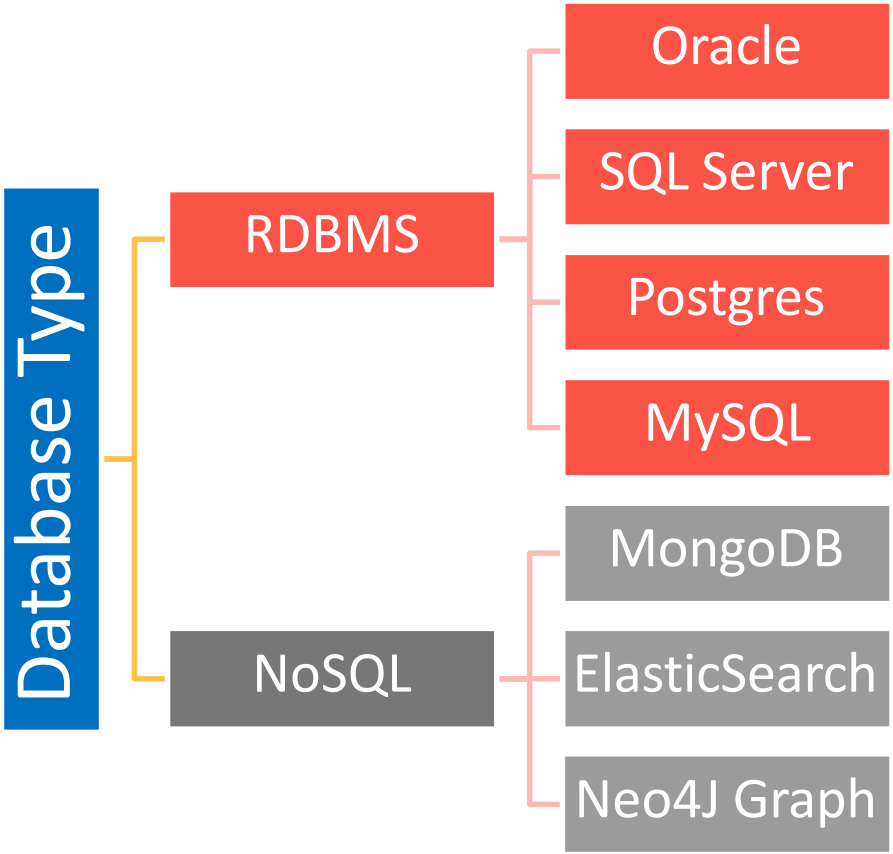
Objective

- Peserta diharapkan menguasai bagaimana create database, melakukan DDL (Data Dynamic Language) dan DML (Data Manipulation Language) menggunakan sintak SQL.

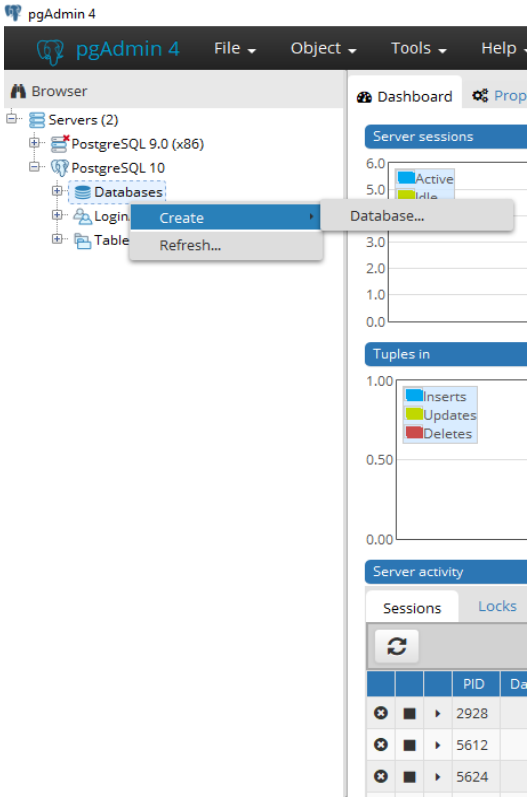
Roadmap PostgreSQL

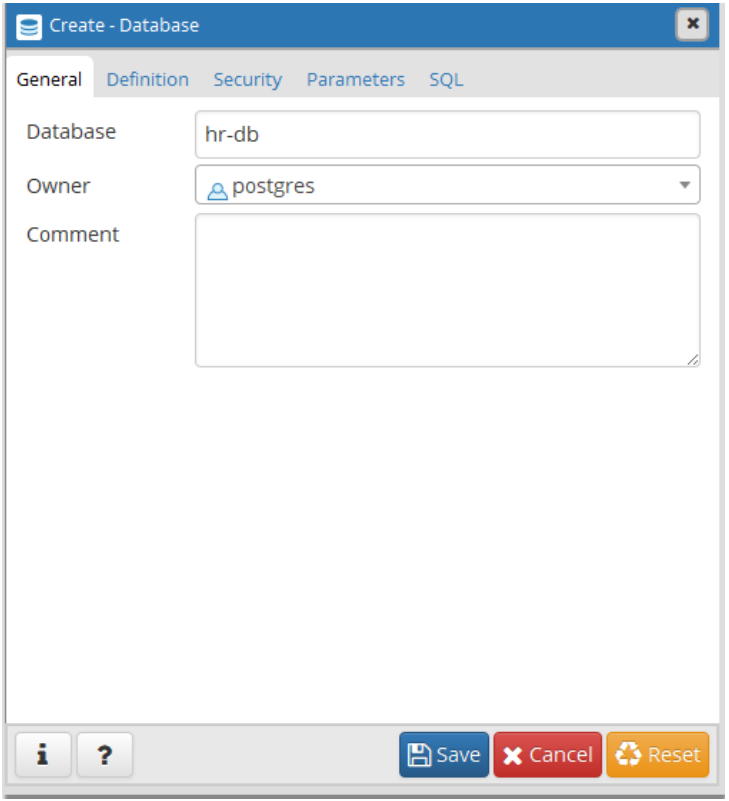


DATABASE TYPE

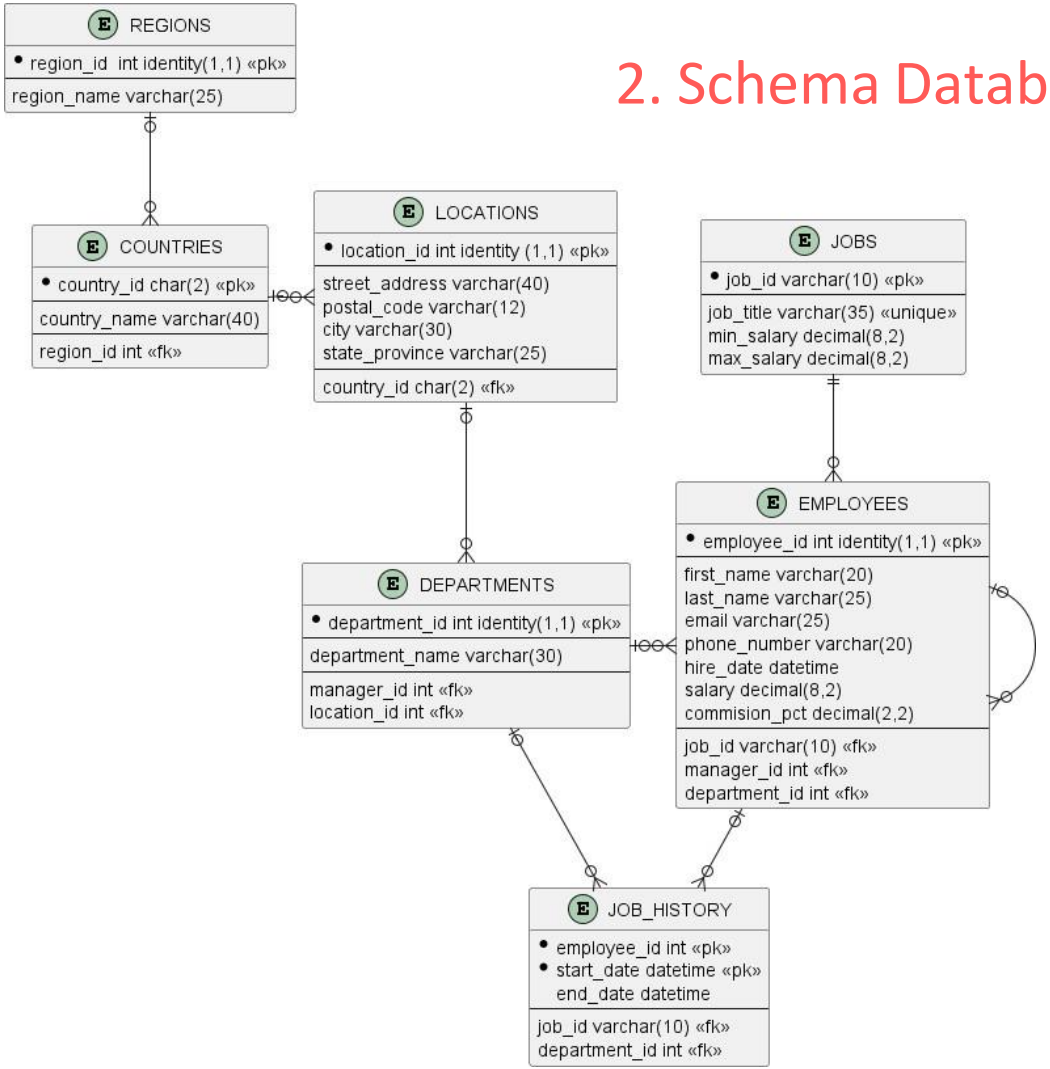


1. Create Database Server

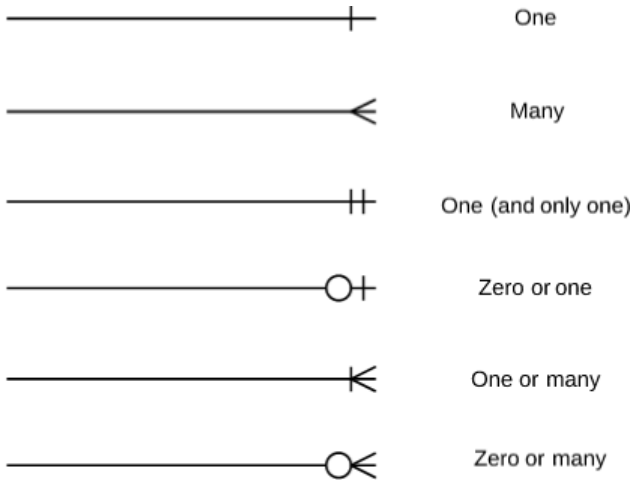




2. Schema Database Human Resource

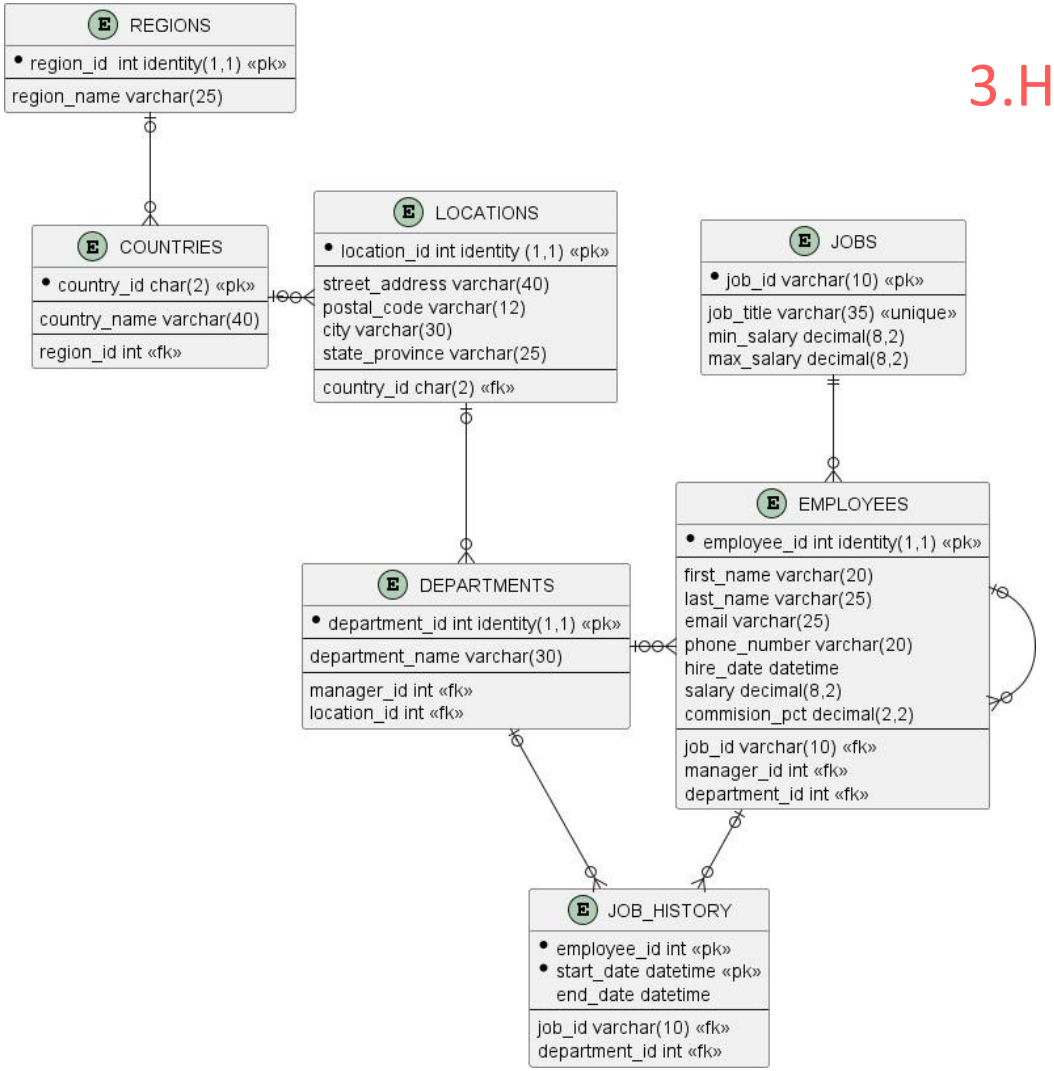


Notation Entity Relation (ER) Diagram



- Primary Key ditandai dengan tanda * dan <<pk>>
- Foreign key ditandai dengan tanda <<fk>>

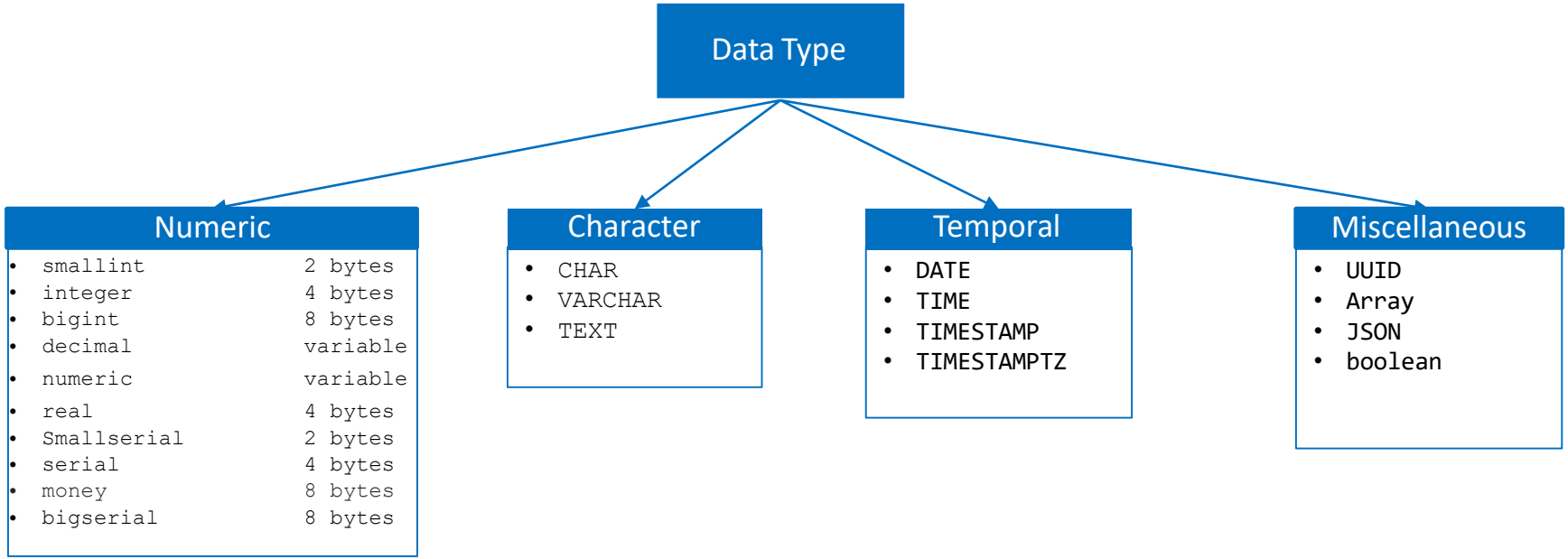
3.How To Create Tables



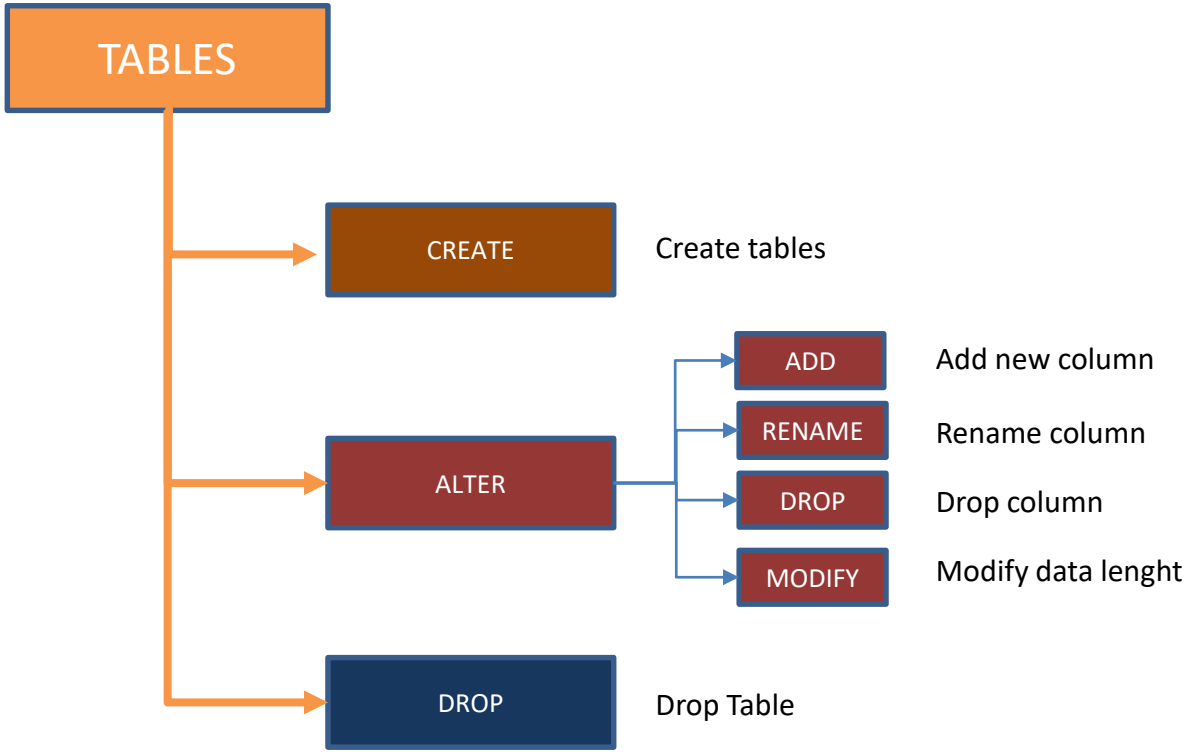
Urutan Create Table

- Mulai dari table master (parent table) atau table referensi terlebih dahulu, contoh:
  - Regions > Countries > Locations
  - Departments > Employees > Jobs atau Job_History
- Create table with scripts

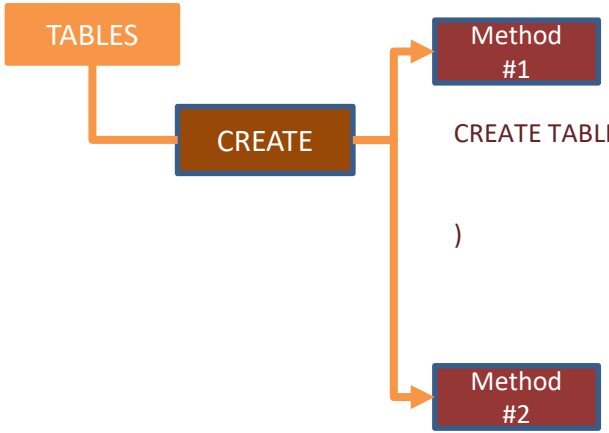
3.1. Data Type PostgreSQL



3.2. Data Definition Language (DDL)



3.2.1. CREATE TABLE




```
CREATE TABLE REGIONS(
                    REGION_ID SERIAL PRIMARY KEY,
                    REGION_NAME CHARACTER VARYING(25)
)
```

Tipe data Serial digunakan untuk autoincrement value

Steps :

1. `CREATE SEQUENCE REGION_ID_SEQ START 1;`

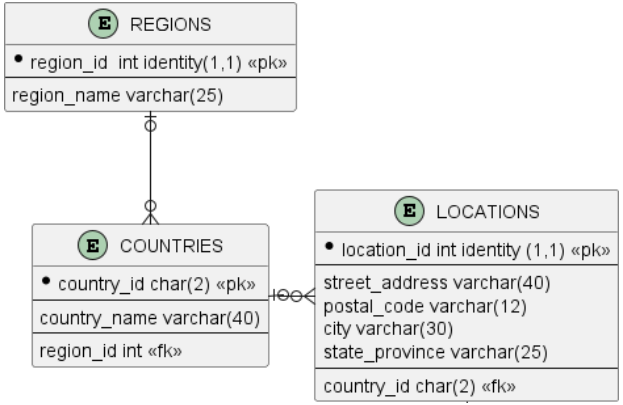
Kita juga bisa gunakan autoincrement menggunakan sequence.

2.
```
CREATE TABLE REGIONS(
                    REGION_ID INT CONSTRAINT REGION_ID_PK PRIMARY KEY DEFAULT
NEXTVAL('REGION_ID_SEQ'),
                    REGION_NAME CHARACTER VARYING(25)
)
```

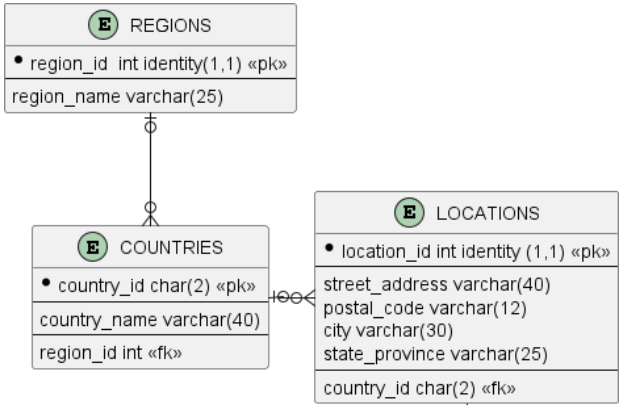
3.2.2. Syntax Create Tables Without Contraints Name

```
CREATE TABLE regions (
    region_id SERIAL PRIMARY KEY,
    region_name VARCHAR (25) DEFAULT NULL
);

CREATE TABLE countries (
    country_id CHAR (2) PRIMARY KEY,
    country_name VARCHAR (40) DEFAULT
    NULL, region_id INT NOT NULL,
    FOREIGN KEY (region_id) REFERENCES regions
    (region_id) ON DELETE CASCADE ON UPDATE CASCADE
);
```



3.2.3. Syntax Create Tables With Contraints Name




```
CREATE TABLE regions (
    region_id SERIAL,
    region_name VARCHAR (25) DEFAULT NULL,
    CONSTRAINT pk_region_id PRIMARY KEY(region_id)
);
```

```
CREATE TABLE countries (
    country_id CHAR (2),
    country_name VARCHAR (40) DEFAULT NULL,
    region_id INT NOT NULL,
    CONSTRAINT pk_country_id PRIMARY KEY(country_id),
    CONSTRAINT fk_region_id FOREIGN KEY (region_id) REFERENCES regions (region_id) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

3.3. ALTER TABLE

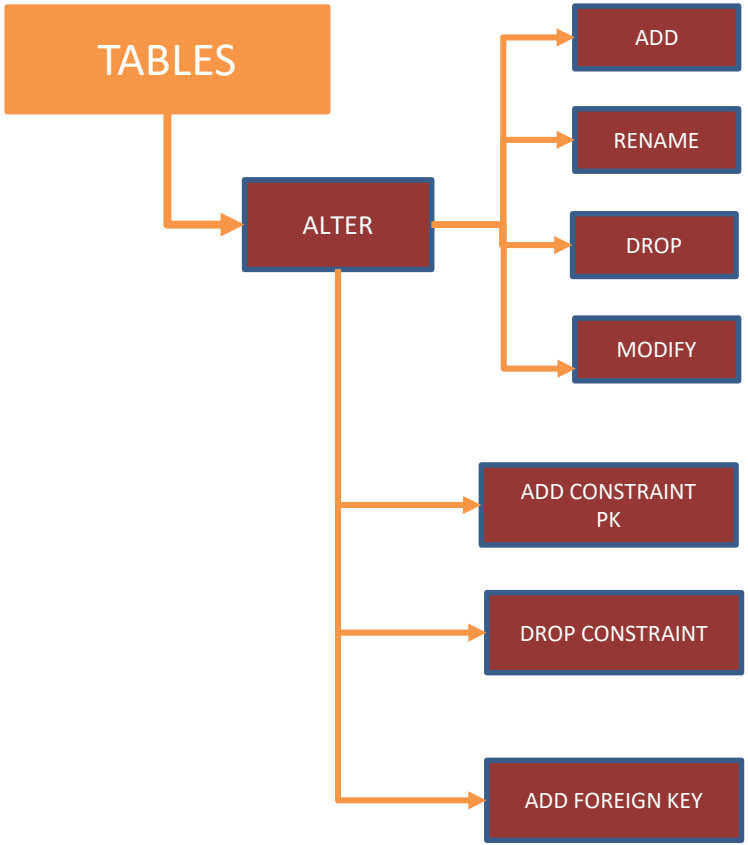


Table name

Column name

```
ALTER TABLE regions ADD COLUMN region_x  VARCHAR(25)
```

```
ALTER TABLE regions RENAME COLUMN region_x TO region_xx
```

New column name

```
ALTER TABLE regions DROP COLUMN region_x
```

```
ALTER TABLE regions ALTER COLUMN region_xx TYPE VARCHAR(30)
```

```
ALTER TABLE regions ADD CONSTRAINT region_id_pk PRIMARY KEY (region_id)
```

Primary key name

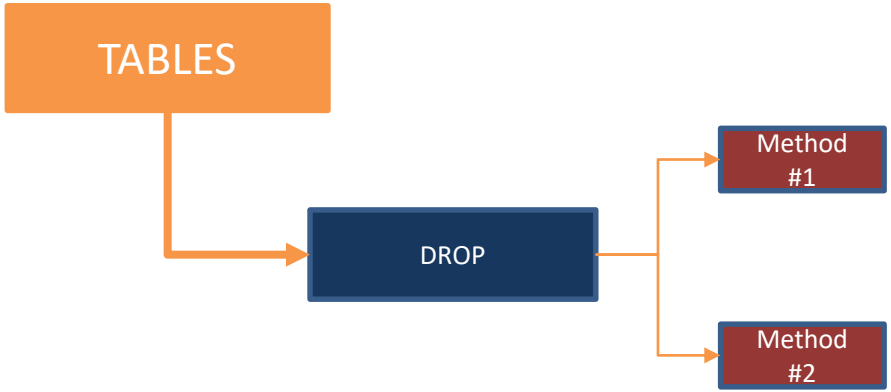
```
ALTER TABLE regions DROP CONSTRAINT region_id_pk
```

Foreign key name

```
ALTER TABLE country ADD CONSTRAINT country_region_id_fk
FOREIGN KEY (region_id)
REFERENCES regions(region_id);
```

Master table

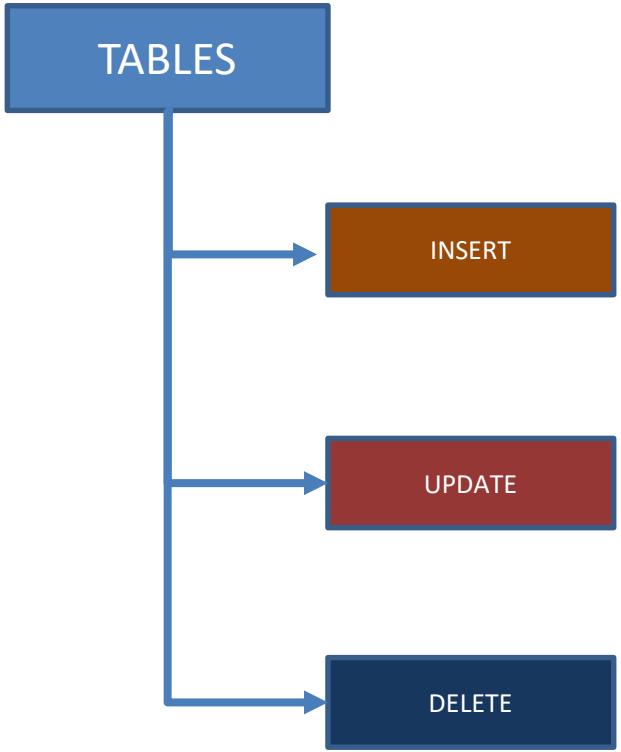
3.3.1. DROP TABLE



DROP TABLE regions

DROP TABLE regions CASCADE;

3.3.2. Insert, Update, Delete




```
insert regions (region_id,region_name) values (5,'Antartic')
```

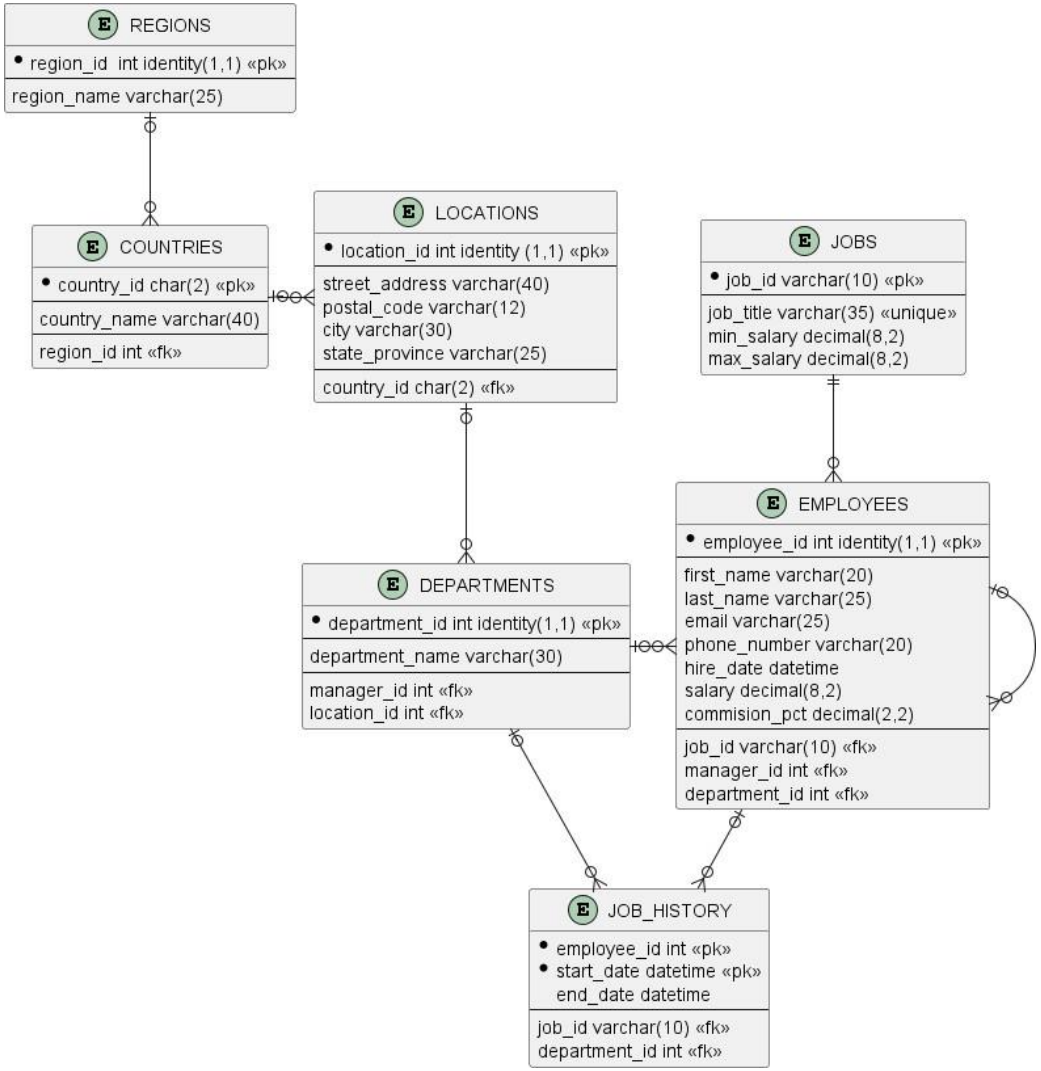
```
insert regions (region_name) values ('Antartic')
```

```
update regions
set region_name ='Antartika'
where region_id=5
```

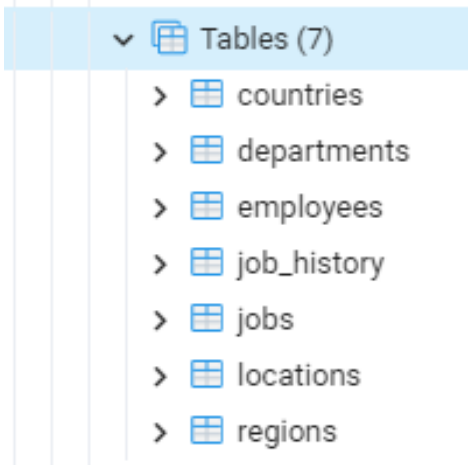
```
delete from regions where region_id=5
```

Practise

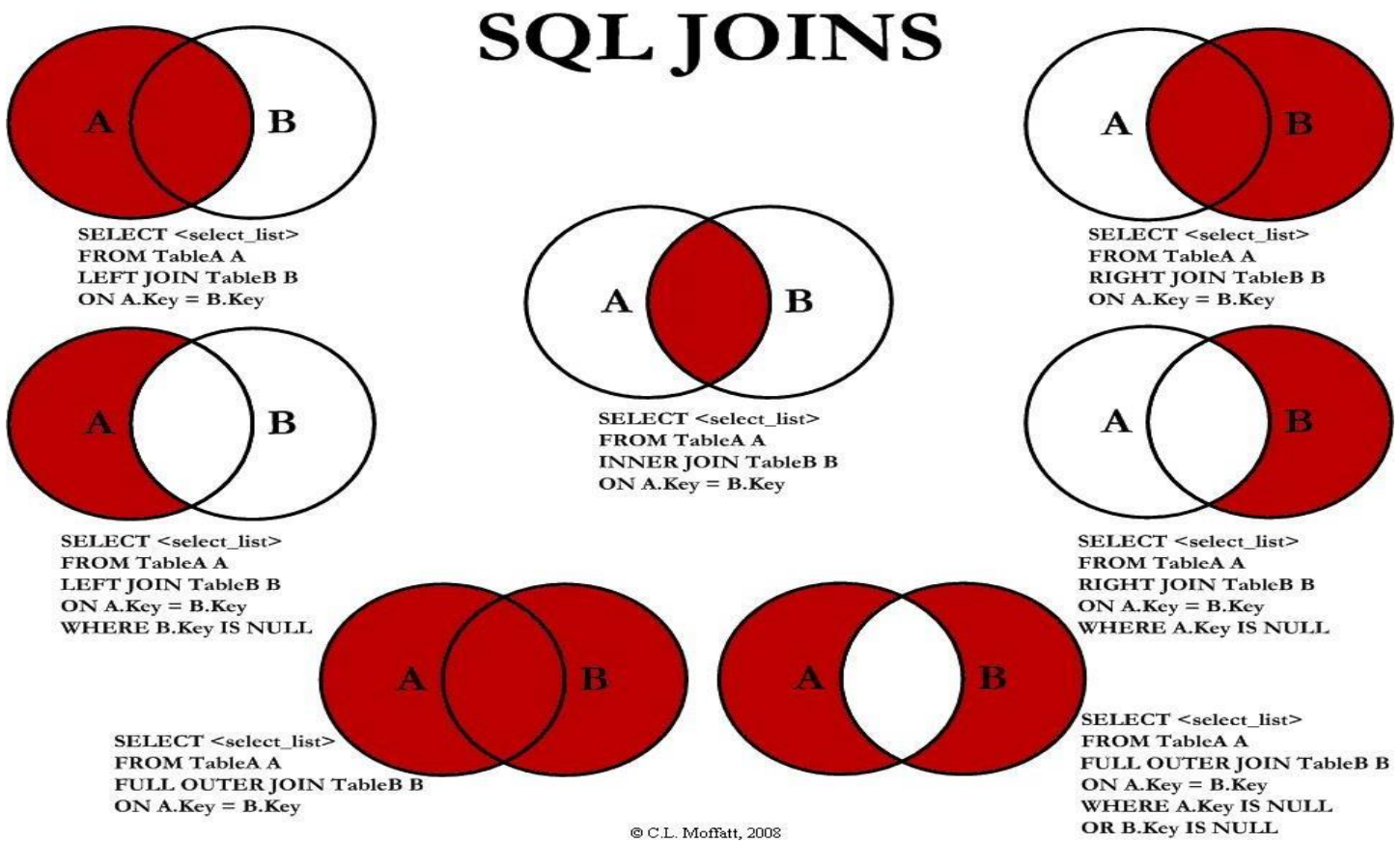
Practise-01



- Create database hr-db
- Create table with constraint name
- Insert data into table
- Int identity = Serial
- Decimal = Numeric



8. SQL JOINS


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

3.3 Inner Join

Method #1 ~ Paralel Execution method #1

```
select a.region_id,a.region_name,country_name
from regions a, countries b
where a.region_id=b.region_id
and a.region_id=1
order by a.region_id
```

Method #2 (Always do this method)

```
select a.region_id,a.region_name,country_name
from regions a inner join countries b
on a.region_id=b.region_id
where a.region_id=1
order by a.region_id
```

```
5  select a.region_id,a.region_name,country_name
6  from regions a, countries b
7  where a.region_id=b.region_id
8  and a.region_id=1
9  order by a.region_id
10
11 select a.region_id,a.region_name,country_name
12 from regions a inner join countries b
13 on a.region_id=b.region_id
14 and a.region_id=1
15 order by a.region_id
```

Data Output | Explain | Messages | Query History

	region_id integer	region_name character varying (25)	country_name character varying (40)
1	1	Europe	Belgium
2	1	Europe	Switzerland
3	1	Europe	Germany
4	1	Europe	Denmark
5	1	Europe	France
6	1	Europe	Italy
7	1	Europe	Netherlands
8	1	Europe	United Kingdom

3.4 Left & Right Join

Step #1

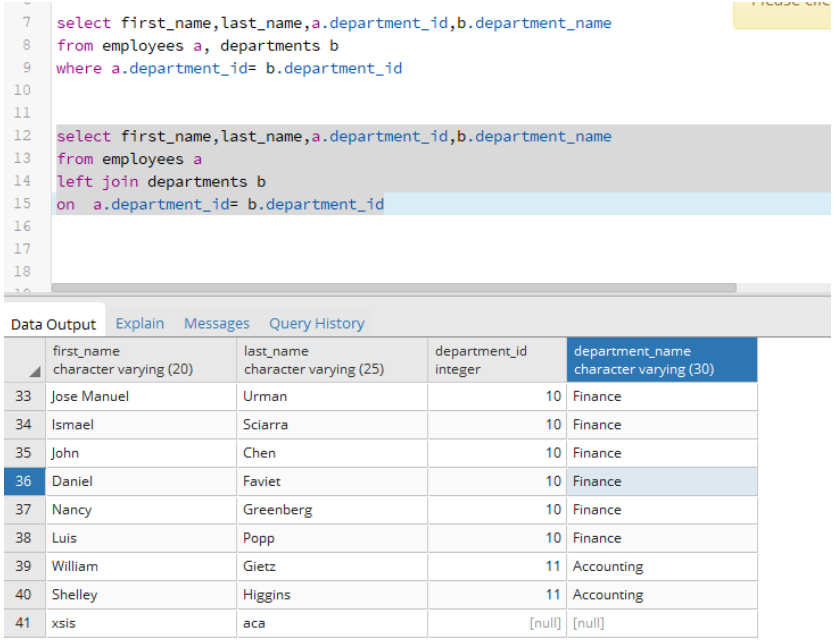
```
INSERT INTO
employees(employee_id,first_name,last_name,email,phone_number,hire_date,j
ob_id,salary,manager_id,department_id) VALUES
(99,'xsis','aca','xa@xsis.com','515.123.4567',DATE '1987-06-
17',4,24000.00,NULL,null);
```

Left Join

```
select first_name,last_name,a.department_id,b.department_name
from employees a
left join departments b
on  a.department_id= b.department_id
```

Right Join

```
select first_name,last_name,a.department_id,a.department_name
from departments a
right join employees b
on  a.department_id= b.department_id
```



```
7  select first_name,last_name,a.department_id,b.department_name
8  from employees a, departments b
9  where a.department_id= b.department_id
10
11
12 select first_name,last_name,a.department_id,b.department_name
13 from employees a
14 left join departments b
15 on  a.department_id= b.department_id
16
17
18
```

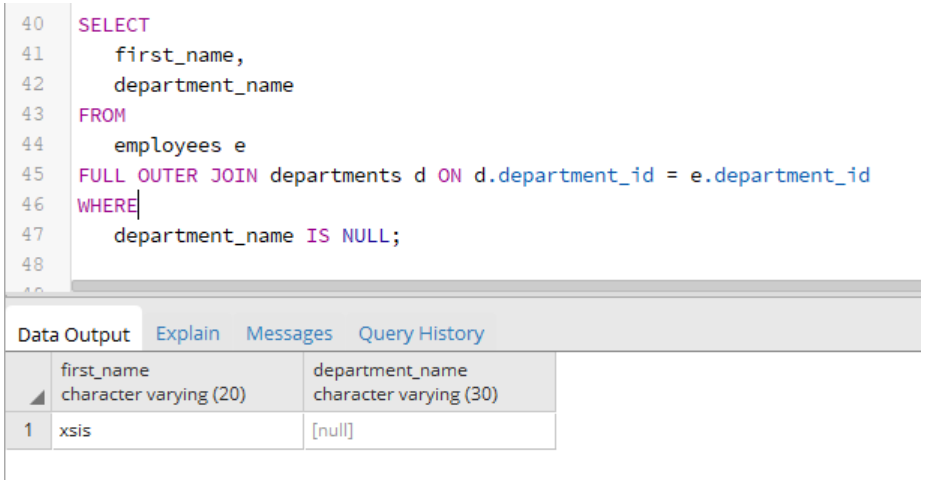
Data Output | Explain | Messages | Query History

	first_name character varying (20)	last_name character varying (25)	department_id integer	department_name character varying (30)
33	Jose Manuel	Urman	10	Finance
34	Ismael	Sciarra	10	Finance
35	John	Chen	10	Finance
36	Daniel	Faviet	10	Finance
37	Nancy	Greenberg	10	Finance
38	Luis	Popp	10	Finance
39	William	Gietz	11	Accounting
40	Shelley	Higgins	11	Accounting
41	xsis	aca	[null]	[null]

3.5 Outer Join

Query

```
SELECT
  first_name,
  department_name
FROM
  employees e
FULL OUTER JOIN departments d
ON d.department_id = e.department_id
WHERE
  department_name IS NULL;
```



3.6 Count

SQL

```
select manager_id,count(employee_id)
from employees
group by manager_id
```

Practise

Tampilkan nama manager dan department name

```
80  select manager_id,count(employee_id)
81  from employees
82  group by manager_id
83
84
```

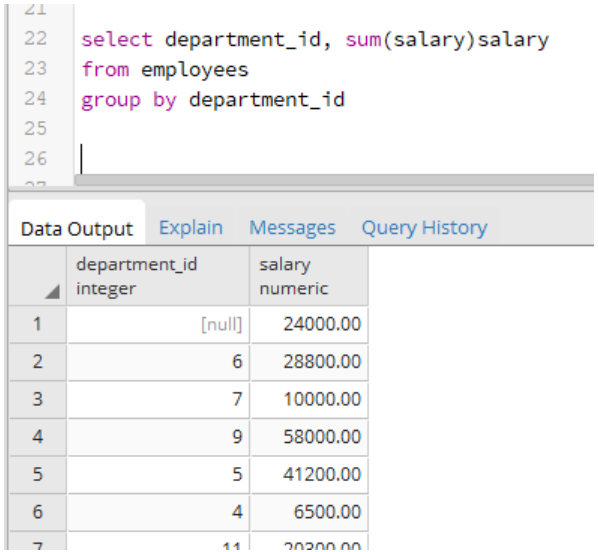
Data Output | Explain | Messages | Query History

	manager_id integer	count bigint
1	120	1
2	102	1
3	[null]	2
4	114	5
5	205	1
6	101	5
7	201	1
8	103	4
9	123	2
10	100	14
11	108	5

3.7 Having Sum & Group By

Before

```
select department_id, sum(salary)salary
from employees
group by department_id
```



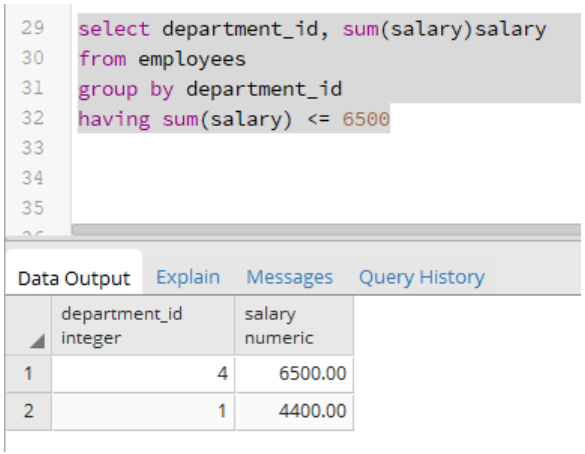
```
22  select department_id, sum(salary)salary
23  from employees
24  group by department_id
25
26
```

Data Output | Explain | Messages | Query History

	department_id integer	salary numeric
1	[null]	24000.00
2	6	28800.00
3	7	10000.00
4	9	58000.00
5	5	41200.00
6	4	6500.00

After

```
select department_id, sum(salary)salary
from employees
group by department_id
having sum(salary) <= 6500
```



```
29  select department_id, sum(salary)salary
30  from employees
31  group by department_id
32  having sum(salary) <= 6500
33
34
35
```

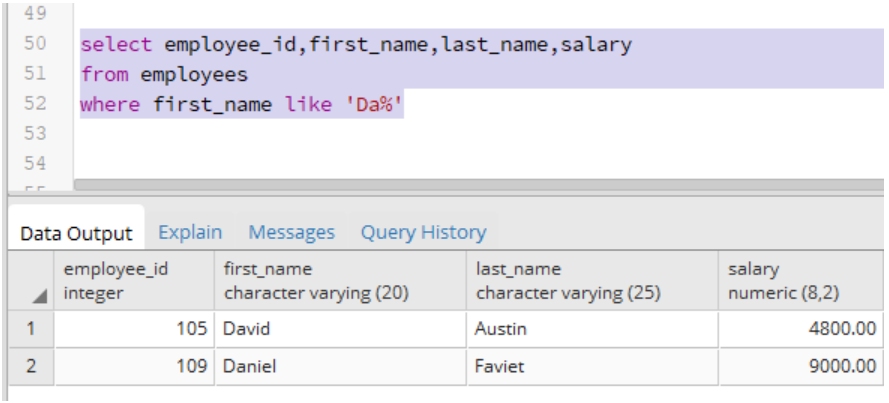
Data Output | Explain | Messages | Query History

	department_id integer	salary numeric
1	4	6500.00
2	1	4400.00

3.8 Like %

Query

```
select employee_id,first_name,last_name,salary
from employees
where first_name like 'Da%'
```



	employee_id integer	first_name character varying (20)	last_name character varying (25)	salary numeric (8,2)
1	105	David	Austin	4800.00
2	109	Daniel	Faviet	9000.00

3.9 SubQuery

Query

```
select *
from departments
where location_id in
(select location_id from locations a,countries b
where a.country_id= b.country_id
and b.region_id=1)
```

```
74
75  select * from departments where location_id in
76  (select location_id from locations a,countries b
77  where a.country_id= b.country_id
78  and b.region_id=1)
79
```

Data Output | Explain | Messages | Query History

	department_id integer	department_name character varying (30)	location_id integer
1	4	Human Resources	2400
2	7	Public Relations	2700
3	8	Sales	2500