Server [localhost]:

Database [postgres]:

Port [5432]:

Username [postgres]:

Password for user postgres:

psql (14.2)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

postgres=# \c dbda\_lab

You are now connected to database "dbda\_lab" as user "postgres".

dbda\_lab=# \d

List of relations

Schema | Name | Type | Owner

--------+---------------+-------+----------

public | agents | table | postgres

public | company6 | table | postgres

public | country | table | postgres

public | ctr | table | postgres

public | department1 | table | postgres

public | dup\_count | table | postgres

public | dup\_countries | table | postgres

public | uni | table | postgres

(8 rows)

dbda\_lab=# create table quantity(serial serial,product varchar(15),quantity numaric);

ERROR: type "numaric" does not exist

LINE 1: ...uantity(serial serial,product varchar(15),quantity numaric);

^

dbda\_lab=# create table quantity(serial serial,product varchar(15),quantity numeric);

CREATE TABLE

dbda\_lab=# insert into quantity values('suger',2);

ERROR: invalid input syntax for type integer: "suger"

LINE 1: insert into quantity values('suger',2);

^

dbda\_lab=# insert into quantity values(1,'suger',2);

INSERT 0 1

dbda\_lab=# insert into quantity values(3,'salt',3);

INSERT 0 1

dbda\_lab=# insert into quantity values(null,'tea',9);

ERROR: null value in column "serial" of relation "quantity" violates not-null constraint

DETAIL: Failing row contains (null, tea, 9).

dbda\_lab=# insert into quantity values(2,'tea',9);

INSERT 0 1

dbda\_lab=# insert into quantity values(4,'oil',5);

INSERT 0 1

dbda\_lab=# insert into quantity values(8,'nuts',7);

INSERT 0 1

dbda\_lab=# table quantity

dbda\_lab-# ;

serial | product | quantity

--------+---------+----------

1 | suger | 2

3 | salt | 3

2 | tea | 9

4 | oil | 5

8 | nuts | 7

(5 rows)

dbda\_lab=# alter table quantity add column no. serial;

ERROR: syntax error at or near "."

LINE 1: alter table quantity add column no. serial;

^

dbda\_lab=# alter table quantity add column 'no.' serial;

ERROR: syntax error at or near "'no.'"

LINE 1: alter table quantity add column 'no.' serial;

^

dbda\_lab=# alter table quantity add column no. serial;

ERROR: syntax error at or near "."

LINE 1: alter table quantity add column no. serial;

^

dbda\_lab=# alter table quantity add no serial;

ALTER TABLE

dbda\_lab=# table quantity;

serial | product | quantity | no

--------+---------+----------+----

1 | suger | 2 | 1

3 | salt | 3 | 2

2 | tea | 9 | 3

4 | oil | 5 | 4

8 | nuts | 7 | 5

(5 rows)

dbda\_lab=# create function add(a int,b int) returns int as $$ begin return a+b ;end ; language plpgsql;

dbda\_lab$# create function add(a int,b int) returns int as $$ begin return a+b ;end $$ ; language plpgsql;

dbda\_lab$#

dbda\_lab$# ;

dbda\_lab$# $$

dbda\_lab-# ;

ERROR: syntax error at or near "return"

LINE 2: ...function add(a int,b int) returns int as $$ begin return a+b...

^

dbda\_lab=# create function add(a int,b int) returns int as $$ begin return a+b ;end $$ ; language plpgsql;

ERROR: no language specified

ERROR: syntax error at or near "language"

LINE 1: language plpgsql;

^

dbda\_lab=# create function add(a int,b int) returns int language plpgsql as $$ begin return a+b ;end $$ ;

CREATE FUNCTION

dbda\_lab=# select\*from add(5,7);

add

-----

12

(1 row)

dbda\_lab=# create function math(a int,b int) returns int language plpgsql as $$ begin return a+b,return a-b,return a\*b ,return a/b ;end $$ ;

ERROR: syntax error at or near "-"

LINE 1: ...t language plpgsql as $$ begin return a+b,return a-b,return ...

^

dbda\_lab=# create function math(in a int,in b int,out c,out d,out e,out f) returns int language plpgsql as $$ begin return c=a+b,return d=a-b,return e=a\*b ,return f=a/b ;end $$ ;

ERROR: type c does not exist

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) returns int language plpgsql as $$ begin return c=a+b,return d=a-b,return e=a\*b ,return f=a/b ;end $$ ;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) returns int language plpgsql as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ ;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ ;

ERROR: no language specified

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ ; language plpgsql;

ERROR: no language specified

ERROR: syntax error at or near "language"

LINE 1: language plpgsql;

^

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ ; Language plpgsql;

ERROR: no language specified

ERROR: syntax error at or near "Language"

LINE 1: Language plpgsql;

^

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

ERROR: assignment source returned 4 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b,e=a\*b ,f=a/b"

PL/pgSQL function math(integer,integer) line 1 at assignment

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c:=a+b,d:=a-b,e:=a\*b ,f:=a/b ;end $$ Language plpgsql;

ERROR: function "math" already exists with same argument types

dbda\_lab=# create or replace function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c:=a+b,d:=a-b,e:=a\*b ,f:=a/b ;end $$ Language plpgsql;

ERROR: syntax error at or near ":="

LINE 1: ...ut d int,out e int,out f int) as $$ begin c:=a+b,d:=a-b,e:=a...

^

dbda\_lab=# create or replace function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# create or replace function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin c=a+b,d=a-b,e=a\*b ,f=a/b ;end ;$$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

ERROR: assignment source returned 4 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b,e=a\*b ,f=a/b"

PL/pgSQL function math(integer,integer) line 1 at assignment

dbda\_lab=# create function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin return c=a+b; return d=a-b; return e=a\*b; return f=a/b; end $$ Language plpgsql;

ERROR: function "math" already exists with same argument types

dbda\_lab=# select\*from math(5,10);

ERROR: assignment source returned 4 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b,e=a\*b ,f=a/b"

PL/pgSQL function math(integer,integer) line 1 at assignment

dbda\_lab=# create or replace function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin return c=a+b; return d=a-b; return e=a\*b; return f=a/b; end $$ Language plpgsql;

ERROR: RETURN cannot have a parameter in function with OUT parameters

LINE 1: ...out d int,out e int,out f int) as $$ begin return c=a+b; ret...

^

dbda\_lab=# create or replace function math(a int, b int,out c int,out d int,out e int,out f int) as $$ begin return c=a+b; return d=a-b; return e=a\*b; return f=a/b; end $$ Language plpgsql;

ERROR: RETURN cannot have a parameter in function with OUT parameters

LINE 1: ...out d int,out e int,out f int) as $$ begin return c=a+b; ret...

^

dbda\_lab=# create or replace function math(a int, b int,out c int,out d int,out e int,out f int) as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

ERROR: RETURN cannot have a parameter in function with OUT parameters

LINE 1: ...out d int,out e int,out f int) as $$ begin return a+b; retur...

^

dbda\_lab=# create or replace function math(in a int, in b int,out c int,out d int,out e int,out f int) as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

ERROR: RETURN cannot have a parameter in function with OUT parameters

LINE 1: ...out d int,out e int,out f int) as $$ begin return a+b; retur...

^

dbda\_lab=# create or replace function math(a int,b int) as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

ERROR: function result type must be specified

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

HINT: Use DROP FUNCTION math(integer,integer) first.

dbda\_lab=# drop function math;

DROP FUNCTION

dbda\_lab=# create function math(a int,b int) returns int as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

math

------

15

(1 row)

dbda\_lab=# create or replace function math(in a int,in b int,out c int,out d int,out e int,out f int) as $$ begin return c=a+b, return d=a-b; return e=a\*b; return f=a/b; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

HINT: Use DROP FUNCTION math(integer,integer) first.

dbda\_lab=# drop function math;

DROP FUNCTION

dbda\_lab=# create function math(a int,b int) returns int as $$ begin return a+b; return a-b; return a\*b; return a/b; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

math

------

15

(1 row)

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b, return a-b; return a\*b; return a/b; end $$ Language plpgsql;

ERROR: syntax error at or near "-"

LINE 1: ... int) returns int as $$ begin return a+b, return a-b; return...

^

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b, a-b; return a\*b; return a/b; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

ERROR: query returned 2 columns

CONTEXT: query: a+b, a-b

PL/pgSQL function math(integer,integer) line 1 at RETURN

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b ,return a-b; end $$ Language plpgsql;

ERROR: syntax error at or near "-"

LINE 1: ... int) returns int as $$ begin return a+b ,return a-b; end $$...

^

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b ,a-b; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

ERROR: query returned 2 columns

CONTEXT: query: a+b ,a-b

PL/pgSQL function math(integer,integer) line 1 at RETURN

dbda\_lab=# create or replace function math(a int,b int,out c int, out d int) returns int as $$ begin c=a+b ,d=a-b; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric, out d numeric) returns int as $$ begin c=a+b ;d=a-b; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric, out d numeric) returns int as $$ begin c=a+b ,d=a-b; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric, out d numeric) returns int as $$ begin c=a+b d=a-b; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(in a numeric,in b numeric,out c numeric, out d numeric) returns int as $$ begin c=a+b d=a-b; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric, out d numeric) returns int as $$ begin c=a+b ; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric, out d numeric) returns numeric as $$ begin c=a+b ; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function math(a numeric,b numeric,out c numeric) returns numeric as $$ begin c=a+b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from math(5,10);

ERROR: query returned 2 columns

CONTEXT: query: a+b ,a-b

PL/pgSQL function math(integer,integer) line 1 at RETURN

dbda\_lab=# create or replace function math(a int,b int) returns int as $$ begin return a+b ,a-b; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# drop function math;

ERROR: function name "math" is not unique

HINT: Specify the argument list to select the function unambiguously.

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric) returns numeric as $$ begin c=a+b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c

----

15

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) returns numeric as $$ begin c=a+b;d=a-b ; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) returns numeric as $$ begin c=a+b,d=a-b ; end $$ Language plpgsql;

ERROR: function result type must be record because of OUT parameters

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) as $$ begin c=a+b,d=a-b ; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

HINT: Use DROP FUNCTION mathss(numeric,numeric) first.

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) as $$ begin c=a+b,d=a-b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

ERROR: assignment source returned 2 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b"

PL/pgSQL function mathss(numeric,numeric) line 1 at assignment

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) as $$ begin c=a+b;d=a-b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c | d

----+----

15 | -5

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric) as $$ begin c=a+b,d=a-b,e=a\*b ; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

DETAIL: Row type defined by OUT parameters is different.

HINT: Use DROP FUNCTION mathss(numeric,numeric) first.

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) as $$ begin c=a+b,d=a-b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

ERROR: assignment source returned 2 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b"

PL/pgSQL function mathss(numeric,numeric) line 1 at assignment

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric) as $$ begin c=a+b;d=a-b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c | d

----+----

15 | -5

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric) as $$ begin c=a+b,d=a-b,e=a\*b ; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

DETAIL: Row type defined by OUT parameters is different.

HINT: Use DROP FUNCTION mathss(numeric,numeric) first.

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric) as $$ begin c=a+b,d=a-b,e=a\*b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

ERROR: assignment source returned 3 columns

CONTEXT: PL/pgSQL assignment "c=a+b,d=a-b,e=a\*b"

PL/pgSQL function mathss(numeric,numeric) line 1 at assignment

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric) as $$ begin c=a+b;d=a-b;e=a\*b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c | d | e

----+----+----

15 | -5 | 50

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric,out f numeric) as $$ begin c=a+b;d=a-b;e=a\*b,f=a/b ; end $$ Language plpgsql;

ERROR: cannot change return type of existing function

DETAIL: Row type defined by OUT parameters is different.

HINT: Use DROP FUNCTION mathss(numeric,numeric) first.

dbda\_lab=# drop function mathss;

DROP FUNCTION

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric,out f numeric) as $$ begin c=a+b;d=a-b;e=a\*b;f=a/b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c | d | e | f

----+----+----+------------------------

15 | -5 | 50 | 0.50000000000000000000

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric,out f decimal(10,5) as $$ begin c=a+b;d=a-b;e=a\*b;f=a/b ; end $$ Language plpgsql;

dbda\_lab(# 0;

dbda\_lab(# )

dbda\_lab-# ;

ERROR: syntax error at or near "as"

LINE 1: ...c,out d numeric,out e numeric,out f decimal(10,5) as $$ begi...

^

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric,out f decimal(10,5)) as $$ begin c=a+b;d=a-b;e=a\*b;f=a/b ; end $$ Language plpgsql;

CREATE FUNCTION

dbda\_lab=# select\*from mathss(5,10);

c | d | e | f

----+----+----+------------------------

15 | -5 | 50 | 0.50000000000000000000

(1 row)

dbda\_lab=# create or replace function mathss(a numeric,b numeric,out c numeric,out d numeric,out e numeric,out f decimal(10,5)) as $$ begin return a+b;d=a-b;e=a\*b;f=a/b ; end $$ Language plpgsql;

ERROR: RETURN cannot have a parameter in function with OUT parameters

LINE 1: ...e numeric,out f decimal(10,5)) as $$ begin return a+b;d=a-b;...

^

dbda\_lab=# \db

List of tablespaces

Name | Owner | Location

------------+----------+----------

pg\_default | postgres |

pg\_global | postgres |

(2 rows)

dbda\_lab=# \d

List of relations

Schema | Name | Type | Owner

--------+---------------------+----------+----------

public | agents | table | postgres

public | company6 | table | postgres

public | country | table | postgres

public | ctr | table | postgres

public | department1 | table | postgres

public | dup\_count | table | postgres

public | dup\_countries | table | postgres

public | quantity | table | postgres

public | quantity\_no\_seq | sequence | postgres

public | quantity\_serial\_seq | sequence | postgres

public | uni | table | postgres

(11 rows)

dbda\_lab=# \d

List of relations

Schema | Name | Type | Owner

--------+---------------------+----------+----------

public | agents | table | postgres

public | company6 | table | postgres

public | country | table | postgres

public | ctr | table | postgres

public | department1 | table | postgres

public | dup\_count | table | postgres

public | dup\_countries | table | postgres

public | quantity | table | postgres

public | quantity\_no\_seq | sequence | postgres

public | quantity\_serial\_seq | sequence | postgres

public | uni | table | postgres

(11 rows)

dbda\_lab=# create table job(job\_id numeric,job\_title varchar(15) default ' ',min\_salary numeric default 8000,max\_salary numeric default null);

CREATE TABLE

dbda\_lab=# table job

dbda\_lab-# ;

job\_id | job\_title | min\_salary | max\_salary

--------+-----------+------------+------------

(0 rows)

dbda\_lab=# \d+ job;

Table "public.job"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+-----------------------+-----------+----------+------------------------+----------+-------------+--------------+-------------

job\_id | numeric | | | | main | | |

job\_title | character varying(15) | | | ' '::character varying | extended | | |

min\_salary | numeric | | | 8000 | main | | |

max\_salary | numeric | | | | main | | |

Access method: heap

dbda\_lab=# alter table job alter column job\_title type text;

ALTER TABLE

dbda\_lab=# \d+ job;

Table "public.job"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+------------------------+----------+-------------+--------------+-------------

job\_id | numeric | | | | main | | |

job\_title | text | | | ' '::character varying | extended | | |

min\_salary | numeric | | | 8000 | main | | |

max\_salary | numeric | | | | main | | |

Access method: heap

dbda\_lab=# begin

dbda\_lab-# ;

BEGIN

dbda\_lab=\*# insert into job values(1,'data eng',,100000),(2,'jr eng',25000,50000),(3,'data scientist',,1000000);

ERROR: syntax error at or near ","

LINE 1: insert into job values(1,'data eng',,100000),(2,'jr eng',250...

^

dbda\_lab=!# insert into job(job\_id,job\_title,max\_salary) values(1,'data eng',100000),(2,'jr eng',50000),(3,'data scientist',1000000);

ERROR: current transaction is aborted, commands ignored until end of transaction block

dbda\_lab=!# rollback

dbda\_lab-!# ;

ROLLBACK

dbda\_lab=# insert into job(job\_id,job\_title,max\_salary) values(1,'data eng',100000),(2,'jr eng',50000),(3,'data scientist',1000000);

INSERT 0 3

dbda\_lab=# table job

dbda\_lab-# ;

job\_id | job\_title | min\_salary | max\_salary

--------+----------------+------------+------------

1 | data eng | 8000 | 100000

2 | jr eng | 8000 | 50000

3 | data scientist | 8000 | 1000000

(3 rows)

dbda\_lab=# insert into job(job\_id,max\_salary) values(4,100000),(5,null);

INSERT 0 2

dbda\_lab=# table job;

job\_id | job\_title | min\_salary | max\_salary

--------+----------------+------------+------------

1 | data eng | 8000 | 100000

2 | jr eng | 8000 | 50000

3 | data scientist | 8000 | 1000000

4 | | 8000 | 100000

5 | | 8000 |

(5 rows)

dbda\_lab=# alter table job add primary key (job\_title);

ERROR: could not create unique index "job\_pkey"

DETAIL: Key (job\_title)=( ) is duplicated.

dbda\_lab=# delete \* from where job\_id=5;

ERROR: syntax error at or near "\*"

LINE 1: delete \* from where job\_id=5;

^

dbda\_lab=# delete \* from job where job\_id=5;

ERROR: syntax error at or near "\*"

LINE 1: delete \* from job where job\_id=5;

^

dbda\_lab=# delete from job where job\_id=5;

DELETE 1

dbda\_lab=# table job;

job\_id | job\_title | min\_salary | max\_salary

--------+----------------+------------+------------

1 | data eng | 8000 | 100000

2 | jr eng | 8000 | 50000

3 | data scientist | 8000 | 1000000

4 | | 8000 | 100000

(4 rows)

dbda\_lab=# alter table job add primary key (job\_title);

ALTER TABLE

dbda\_lab=# table job;

job\_id | job\_title | min\_salary | max\_salary

--------+----------------+------------+------------

1 | data eng | 8000 | 100000

2 | jr eng | 8000 | 50000

3 | data scientist | 8000 | 1000000

4 | | 8000 | 100000

(4 rows)

dbda\_lab=# \d+ job;

Table "public.job"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+------------------------+----------+-------------+--------------+-------------

job\_id | numeric | | | | main | | |

job\_title | text | | not null | ' '::character varying | extended | | |

min\_salary | numeric | | | 8000 | main | | |

max\_salary | numeric | | | | main | | |

Indexes:

"job\_pkey" PRIMARY KEY, btree (job\_title)

Access method: heap

dbda\_lab=# insert into job(job\_id,job\_title) values(5,'data analyst);

dbda\_lab'# '

dbda\_lab(# )

dbda\_lab-# ;

INSERT 0 1

dbda\_lab=# \d+ job;

Table "public.job"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+------------------------+----------+-------------+--------------+-------------

job\_id | numeric | | | | main | | |

job\_title | text | | not null | ' '::character varying | extended | | |

min\_salary | numeric | | | 8000 | main | | |

max\_salary | numeric | | | | main | | |

Indexes:

"job\_pkey" PRIMARY KEY, btree (job\_title)

Access method: heap

dbda\_lab=# table job;

job\_id | job\_title | min\_salary | max\_salary

--------+----------------+------------+------------

1 | data eng | 8000 | 100000

2 | jr eng | 8000 | 50000

3 | data scientist | 8000 | 1000000

4 | | 8000 | 100000

5 | data analyst);+| 8000 |

| | |

(5 rows)

dbda\_lab=# \d

List of relations

Schema | Name | Type | Owner

--------+---------------------+----------+----------

public | agents | table | postgres

public | company6 | table | postgres

public | country | table | postgres

public | ctr | table | postgres

public | department1 | table | postgres

public | dup\_count | table | postgres

public | dup\_countries | table | postgres

public | job | table | postgres

public | quantity | table | postgres

public | quantity\_no\_seq | sequence | postgres

public | quantity\_serial\_seq | sequence | postgres

public | uni | table | postgres

(12 rows)

dbda\_lab=# create table countries('country\_id' serial unique,'country\_name' text default 'N/A');

ERROR: syntax error at or near "'country\_id'"

LINE 1: create table countries('country\_id' serial unique,'country\_n...

^

dbda\_lab=# create table countries('country\_id' numeric serial unique,'country\_name' text default 'N/A');

ERROR: syntax error at or near "'country\_id'"

LINE 1: create table countries('country\_id' numeric serial unique,'c...

^

dbda\_lab=# create table countries('country\_id' serial,'country\_name' text default 'N/A');

ERROR: syntax error at or near "'country\_id'"

LINE 1: create table countries('country\_id' serial,'country\_name' te...

^

dbda\_lab=# create table countries(country\_id serial unique,country\_name text default 'N/A');

CREATE TABLE

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

--------------+---------+-----------+----------+-----------------------------------------------+----------+-------------+--------------+-------------

country\_id | integer | | not null | nextval('countries\_country\_id\_seq'::regclass) | plain | | |

country\_name | text | | | 'N/A'::text | extended | | |

Indexes:

"countries\_country\_id\_key" UNIQUE CONSTRAINT, btree (country\_id)

Access method: heap

dbda\_lab=# insert into countries(country\_name) values(INDIA);

ERROR: column "india" does not exist

LINE 1: insert into countries(country\_name) values(INDIA);

^

dbda\_lab=# insert into countries(country\_name) values('INDIA');

INSERT 0 1

dbda\_lab=# insert into countries(country\_id) values(3);

INSERT 0 1

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

--------------+---------+-----------+----------+-----------------------------------------------+----------+-------------+--------------+-------------

country\_id | integer | | not null | nextval('countries\_country\_id\_seq'::regclass) | plain | | |

country\_name | text | | | 'N/A'::text | extended | | |

Indexes:

"countries\_country\_id\_key" UNIQUE CONSTRAINT, btree (country\_id)

Access method: heap

dbda\_lab=# table countries;

country\_id | country\_name

------------+--------------

1 | INDIA

3 | N/A

(2 rows)

dbda\_lab=# insert into countries(country\_name) values('NEPAL');

INSERT 0 1

dbda\_lab=# insert into countries(country\_name) values('BANGLADESH');

ERROR: duplicate key value violates unique constraint "countries\_country\_id\_key"

DETAIL: Key (country\_id)=(3) already exists.

dbda\_lab=# table countries;

country\_id | country\_name

------------+--------------

1 | INDIA

3 | N/A

2 | NEPAL

(3 rows)

dbda\_lab=# insert into countries(country\_id) values(4);

INSERT 0 1

dbda\_lab=# table countries;

country\_id | country\_name

------------+--------------

1 | INDIA

3 | N/A

2 | NEPAL

4 | N/A

(4 rows)

dbda\_lab=# insert into countries(country\_name) values('BHUTAN');

ERROR: duplicate key value violates unique constraint "countries\_country\_id\_key"

DETAIL: Key (country\_id)=(4) already exists.

dbda\_lab=# insert into countries(country\_name) values('BHUTAN');

INSERT 0 1

dbda\_lab=# table countries;

country\_id | country\_name

------------+--------------

1 | INDIA

3 | N/A

2 | NEPAL

4 | N/A

5 | BHUTAN

(5 rows)

dbda\_lab=# drop table countries ;

DROP TABLE

dbda\_lab=# create table countries(country\_id numeric,region\_id numeric);

CREATE TABLE

dbda\_lab=# drop table countries;

DROP TABLE

dbda\_lab=# create table countries(country\_id numeric not null,region\_id numeric not null);

CREATE TABLE

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+---------+---------+-------------+--------------+-------------

country\_id | numeric | | not null | | main | | |

region\_id | numeric | | not null | | main | | |

Access method: heap

dbda\_lab=# CREATE TABLE IF NOT EXISTS countries (

dbda\_lab(# COUNTRY\_ID integer NOT NULL,

dbda\_lab(# COUNTRY\_NAME varchar(40) NOT NULL,

dbda\_lab(# REGION\_ID integer NOT NULL,

dbda\_lab(# PRIMARY KEY (COUNTRY\_ID,REGION\_ID)

dbda\_lab(# );

NOTICE: relation "countries" already exists, skipping

CREATE TABLE

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+---------+---------+-------------+--------------+-------------

country\_id | numeric | | not null | | main | | |

region\_id | numeric | | not null | | main | | |

Access method: heap

dbda\_lab=# CREATE TABLE IF EXISTS countries (

dbda\_lab(# COUNTRY\_ID integer NOT NULL,

dbda\_lab(# COUNTRY\_NAME varchar(40) NOT NULL,

dbda\_lab(# REGION\_ID integer NOT NULL,

dbda\_lab(# PRIMARY KEY (COUNTRY\_ID,REGION\_ID)

dbda\_lab(# );

ERROR: syntax error at or near "EXISTS"

LINE 1: CREATE TABLE IF EXISTS countries (

^

dbda\_lab=# );CREATE or replace countries (

ERROR: syntax error at or near ")"

LINE 1: );

^

dbda\_lab(# COUNTRY\_ID integer NOT NULL,

dbda\_lab(# COUNTRY\_NAME varchar(40) NOT NULL,

dbda\_lab(# REGION\_ID integer NOT NULL,

dbda\_lab(# PRIMARY KEY (COUNTRY\_ID,REGION\_ID)

dbda\_lab(# );

ERROR: syntax error at or near "countries"

LINE 1: CREATE or replace countries (

^

dbda\_lab=# CREATE or replace table countries (

dbda\_lab(# COUNTRY\_ID integer NOT NULL,

dbda\_lab(# COUNTRY\_NAME varchar(40) NOT NULL,

dbda\_lab(# REGION\_ID integer NOT NULL,

dbda\_lab(# PRIMARY KEY (COUNTRY\_ID,REGION\_ID)

dbda\_lab(# );

ERROR: syntax error at or near "table"

LINE 1: CREATE or replace table countries (

^

dbda\_lab=# drop table coutries;

ERROR: table "coutries" does not exist

dbda\_lab=# drop table countries;

DROP TABLE

dbda\_lab=# CREATE TABLE IF NOT EXISTS countries ( COUNTRY\_ID integer NOT NULL, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer NOT NULL, PRIMARY KEY(COUNTRY\_ID,REGION\_ID)

dbda\_lab(# ;

dbda\_lab(# )

dbda\_lab-# ;

ERROR: syntax error at or near ";"

LINE 2: ;

^

dbda\_lab=# CREATE TABLE IF NOT EXISTS countries ( COUNTRY\_ID integer NOT NULL, COUNTRY\_NAME varchar(40) NOT NULL, REGION\_ID integer NOT NULL, PRIMARY KEY(COUNTRY\_ID,REGION\_ID));

CREATE TABLE

dbda\_lab=# table countries;

country\_id | country\_name | region\_id

------------+--------------+-----------

(0 rows)

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

--------------+-----------------------+-----------+----------+---------+----------+-------------+--------------+-------------

country\_id | integer | | not null | | plain | | |

country\_name | character varying(40) | | not null | | extended | | |

region\_id | integer | | not null | | plain | | |

Indexes:

"countries\_pkey" PRIMARY KEY, btree (country\_id, region\_id)

Access method: heap

dbda\_lab=# alter table countries drop column country\_name;

ALTER TABLE

dbda\_lab=# \d+ countries;

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+---------+-----------+----------+---------+---------+-------------+--------------+-------------

country\_id | integer | | not null | | plain | | |

region\_id | integer | | not null | | plain | | |

Indexes:

"countries\_pkey" PRIMARY KEY, btree (country\_id, region\_id)

Access method: heap

dbda\_lab=# table countries;

country\_id | region\_id

------------+-----------

(0 rows)

dbda\_lab=# alter table countries add column name varchar(15) ;

ALTER TABLE

dbda\_lab=# alter table countries modify column name varchar(15) default 'aa' ;

ERROR: syntax error at or near "modify"

LINE 1: alter table countries modify column name varchar(15) default...

^

dbda\_lab=# alter table countries modify name varchar(15) default 'aa' ;

ERROR: syntax error at or near "modify"

LINE 1: alter table countries modify name varchar(15) default 'aa' ;

^

dbda\_lab=# alter table countries modify name varchar(15) default 'aa' ;

ERROR: syntax error at or near "modify"

LINE 1: alter table countries modify name varchar(15) default 'aa' ;

^

dbda\_lab=# alter table countries modify name varchar(15) not null default 'aa' ;

ERROR: syntax error at or near "modify"

LINE 1: alter table countries modify name varchar(15) not null defau...

^

dbda\_lab=# alter table countries modify name varchar(15) not null ;

ERROR: syntax error at or near "modify"

LINE 1: alter table countries modify name varchar(15) not null ;

^

dbda\_lab=# alter table countries alter column modify name varchar(15) not null ;

ERROR: syntax error at or near "name"

LINE 1: alter table countries alter column modify name varchar(15) n...

^

dbda\_lab=# insert into countries values(356,20),(764,28);

INSERT 0 2

dbda\_lab=# insert into countries values(356,20);

ERROR: duplicate key value violates unique constraint "countries\_pkey"

DETAIL: Key (country\_id, region\_id)=(356, 20) already exists.

dbda\_lab=# \d+ countries

Table "public.countries"

Column | Type | Collation | Nullable | Default | Storage | Compression | Stats target | Description

------------+-----------------------+-----------+----------+---------+----------+-------------+--------------+-------------

country\_id | integer | | not null | | plain | | |

region\_id | integer | | not null | | plain | | |

name | character varying(15) | | | | extended | | |

Indexes:

"countries\_pkey" PRIMARY KEY, btree (country\_id, region\_id)

Access method: heap

dbda\_lab=# table countries:

dbda\_lab-# ;

ERROR: syntax error at or near ":"

LINE 1: table countries:

^

dbda\_lab=# table countries;

country\_id | region\_id | name

------------+-----------+------

356 | 20 |

764 | 28 |

(2 rows)

dbda\_lab=#