insert into positions values (3, 'manager')

select \* from positions

select \* from ingredients

select \* from employee order by id ASC

select \* from menu

select \* from dishes

select \* from list\_of\_ordered\_dishes

select \* from prepared\_dishes

select \* from measures

select \* from warehouse

select \* from orders

select \* from recipe

insert into dishes values (1, 'Water', 10, 0.5, 2)

insert into menu values (1, 1)

ALTER TABLE orders DROP COLUMN name;

select \* from ingredients order by id asc

select \* from employee INNER JOIN positions ON (employee.position\_id = positions.id)

insert into employee values (3, 'Sydorov', 'Sydor', '01-01-2003', +380661237, 2, 800)

update employee set salary =100000 where id=1

drop table menu

drop table menu\_set

ALTER TABLE dishes ADD COLUMN measure integer

ALTER TABLE dishes DROP COLUMN weight

ALTER TABLE employee ADD CONSTRAINT fk\_id\_positions FOREIGN KEY (position\_id) REFERENCES positions (id) MATCH FULL;

ALTER TABLE employee DROP CONSTRAINT fk\_id\_positions

ALTER TABLE dishes ADD CONSTRAINT weight\_fkey\_check CHECK (weight > 0)

ALTER TABLE menu RENAME COLUMN id TO dish\_number

CREATE TABLE public.measures

(

id integer NOT NULL,

name character varying NOT NULL,

CONSTRAINT measures\_pkey PRIMARY KEY (id)

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.recipe

(

id integer NOT NULL,

id\_dish integer NOT NULL,

id\_ingredients integer NOT NULL,

CONSTRAINT recipe\_pkey PRIMARY KEY (id),

CONSTRAINT pecipe\_id\_dish\_fkey FOREIGN KEY (id\_dish)

REFERENCES public.dishes (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT recipe\_id\_ingredients\_fkey FOREIGN KEY (id\_ingredients)

REFERENCES public.ingredients (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.list\_of\_ordered\_dishes

(

id integer NOT NULL,

id\_order integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT list\_pkey PRIMARY KEY (id),

CONSTRAINT list\_id\_order\_fkey FOREIGN KEY (id\_order)

REFERENCES public.orders (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT list\_id\_dish\_fkey FOREIGN KEY (id\_dish)

REFERENCES public.dishes (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.orders

(

id integer NOT NULL,

id\_waiter integer NOT NULL,

table\_number integer NOT NULL,

name character varying NOT NULL,

order\_date date,

CONSTRAINT orders\_pkey PRIMARY KEY (id),

CONSTRAINT orders\_id\_waiter\_fkey FOREIGN KEY (id\_waiter)

REFERENCES public.employee (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.list\_of\_ordered\_dishes

(

id integer NOT NULL,

id\_order integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT list\_pkey PRIMARY KEY (id),

CONSTRAINT list\_id\_order\_fkey FOREIGN KEY (id\_order)

REFERENCES public.orders (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT list\_id\_dish\_fkey FOREIGN KEY (id\_dish)

REFERENCES public.dishes (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.prepared\_dishes

(

id integer NOT NULL,

id\_dishes integer NOT NULL,

id\_cooker integer NOT NULL,

id\_order integer NOT NULL,

order\_date date,

CONSTRAINT prepared\_dishes\_pkey PRIMARY KEY (id),

CONSTRAINT prepared\_dishes\_id\_cooker\_fkey FOREIGN KEY (id\_cooker)

REFERENCES public.employee (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT prepared\_dishes\_id\_dishes\_fkey FOREIGN KEY (id\_dishes)

REFERENCES public.dishes (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT prepared\_dishes\_id\_order\_fkey FOREIGN KEY (id\_order)

REFERENCES public.orders (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.menu

(

id integer NOT NULL,

name character varying NOT NULL,

CONSTRAINT menu\_pkey PRIMARY KEY (id)

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.menu\_set

(

id\_menu integer NOT NULL,

id\_dish integer NOT NULL,

CONSTRAINT menu\_set\_pkey PRIMARY KEY (id\_menu, id\_dish),

CONSTRAINT menu\_set\_id\_menu FOREIGN KEY (id\_menu)

REFERENCES public.menu (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT menu\_set\_id\_dish FOREIGN KEY (id\_dish)

REFERENCES public.dishes (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.menu\_set (

id\_menu INTEGER NOT NULL REFERENCES public.menu(id),

id\_dish INTEGER NOT NULL REFERENCES public.dishes(id),

PRIMARY KEY (id\_menu, id\_dish)

);

CREATE TABLE public.dishes

(

id integer NOT NULL,

name character varying NOT NULL,

price double precision,

weight integer,

CONSTRAINT dishes\_pkey PRIMARY KEY (id),

CONSTRAINT dishes\_price\_check CHECK (price > 0::double precision),

CONSTRAINT dishes\_weight\_check CHECK (weight > 0)

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.ingredients

(

id integer NOT NULL,

name character varying NOT NULL,

CONSTRAINT ingredients\_pkey PRIMARY KEY (id)

)

WITH (

OIDS=FALSE

);

CREATE TABLE public.warehouse

(

id integer NOT NULL,

id\_ingredient integer NOT NULL,

quantity integer,

measure character varying(10) NOT NULL,

CONSTRAINT warehouse\_pkey PRIMARY KEY (id),

CONSTRAINT warehouse\_id\_ingredient\_fkey FOREIGN KEY (id\_ingredient)

REFERENCES public.ingredients (id) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT warehouse\_id\_ingredient\_key UNIQUE (id\_ingredient)

)

WITH (

OIDS=FALSE

);

insert into warehouse values (10, 10, 11,2)

select warehouse.id, ingredients.name, quantity, measures.name from warehouse INNER JOIN measures ON (warehouse.measure = measures.id)

INNER JOIN ingredients ON (warehouse.id\_ingredient = ingredients.id)

select \* from measures

select \* from warehouse

select \* from ingredients order by id asc

insert into measures values (3,'piece')