

به نام خدا

محمد مهدی آقاجانی

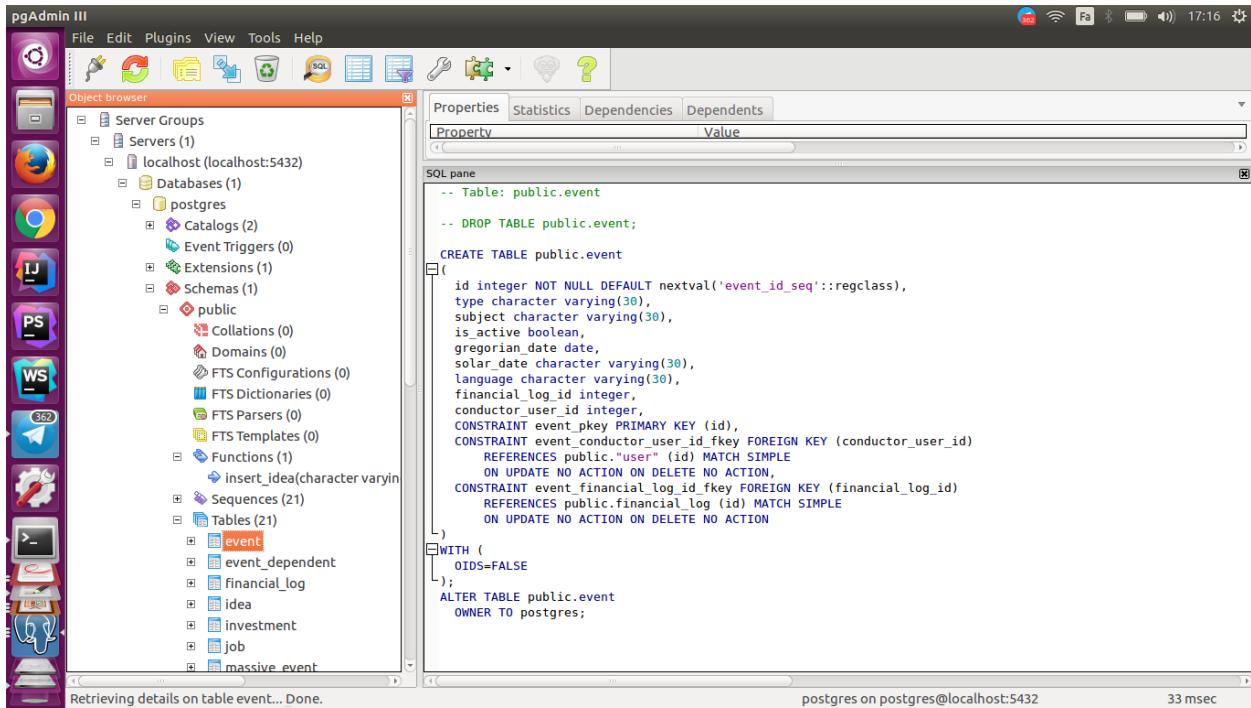
گزارش فاز اول پروژه

در ابتدا نمودار ER طراحی شد که فایل آن در فolder پروژه موجود میباشد. این نمودار شامل دو مرحله است که در مرحله دوم

multivalued attribute ها نیز تبدیل به جدول شده اند.

اما بعد از طراحی Erd جداول مربوطه ساخته شد که کد sql طراحی جداول نیز در فolder پروژه موجود است و با نام `data` شناخته می شود.

حال پرینت اسکرین جداول ساخته شده را می بینیم :

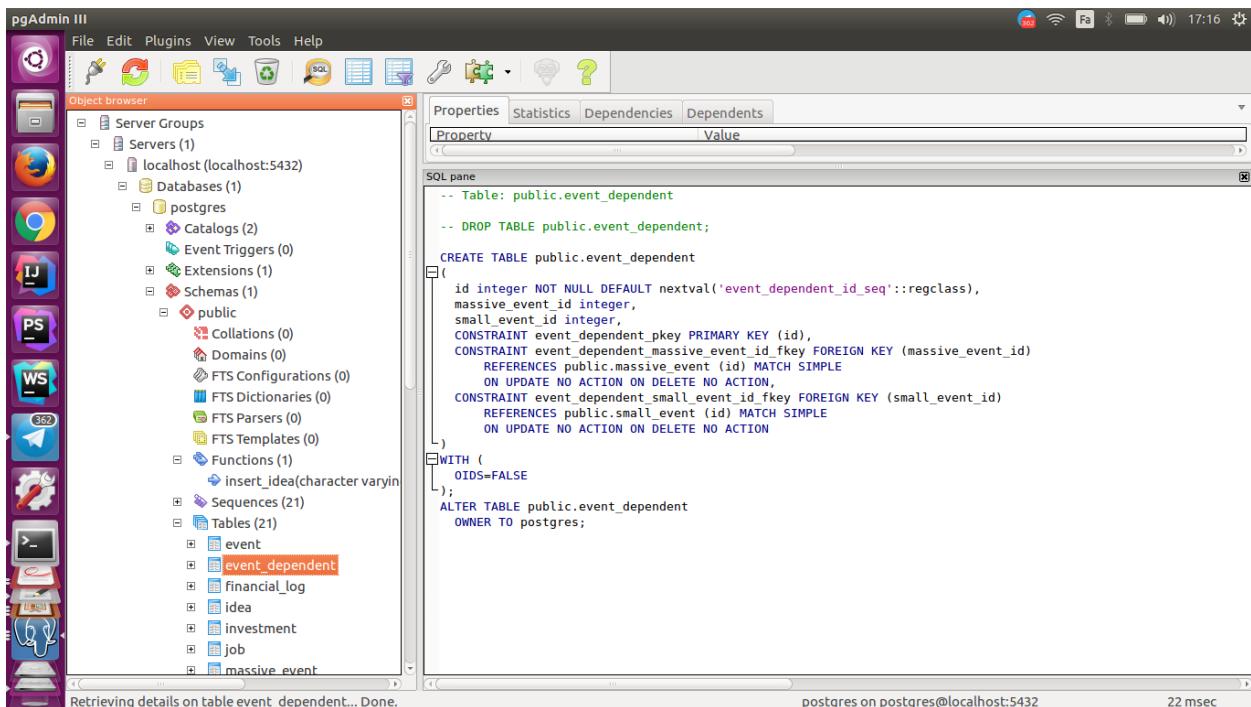


The screenshot shows the pgAdmin III interface. The left pane is the Object browser, displaying the database structure under the 'localhost' server. The right pane is the SQL pane, showing the SQL code for creating the 'public.event' table. The table has columns for id, type, subject, is_active, gregorian_date, solar_date, language, financial_log_id, conductor_user_id, and various constraints like PRIMARY KEY and FOREIGN KEY.

```

-- Table: public.event
-- DROP TABLE public.event;
CREATE TABLE public.event
(
    id integer NOT NULL DEFAULT nextval('event_id_seq'::regclass),
    type character varying(30),
    subject character varying(30),
    is_active boolean,
    gregorian_date date,
    solar_date character varying(30),
    language character varying(30),
    financial_log_id integer,
    conductor_user_id integer,
    CONSTRAINT event_pkey PRIMARY KEY (id),
    CONSTRAINT event_conductor_user_id_fkey FOREIGN KEY (conductor_user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT event_financial_log_id_fkey FOREIGN KEY (financial_log_id)
        REFERENCES public.financial_log (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.event
OWNER TO postgres;

```



The screenshot shows the pgAdmin III interface again. The left pane is the Object browser, and the right pane is the SQL pane. This time, it displays the SQL code for creating the 'public.event_dependent' table. The table has columns for id, massive_event_id, small_event_id, and various constraints involving PRIMARY KEY and FOREIGN KEY.

```

-- Table: public.event_dependent
-- DROP TABLE public.event_dependent;
CREATE TABLE public.event_dependent
(
    id integer NOT NULL DEFAULT nextval('event_dependent_id_seq'::regclass),
    massive_event_id integer,
    small_event_id integer,
    CONSTRAINT event_dependent_pkey PRIMARY KEY (id),
    CONSTRAINT event_dependent_massive_event_id_fkey FOREIGN KEY (massive_event_id)
        REFERENCES public.massive_event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT event_dependent_small_event_id_fkey FOREIGN KEY (small_event_id)
        REFERENCES public.small_event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.event_dependent
OWNER TO postgres;

```

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- Server Groups
 - Servers (1)
 - localhost (localhost:5432)
 - Databases (1)
 - postgres
 - Catalogs (2)
 - Event Triggers (0)
 - Extensions (1)

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.financial_log
-- DROP TABLE public.financial_log;

CREATE TABLE public.financial_log
(
    id integer NOT NULL DEFAULT nextval('financial_log_id_seq'::regclass),
    payment_information character varying(50),
    transaction_information character varying(50),
    CONSTRAINT financial_log_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.financial_log
    OWNER TO postgres;
```

Retrieving details on table financial_log... Done.

postgres on postgres@localhost:5432 23 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log**
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

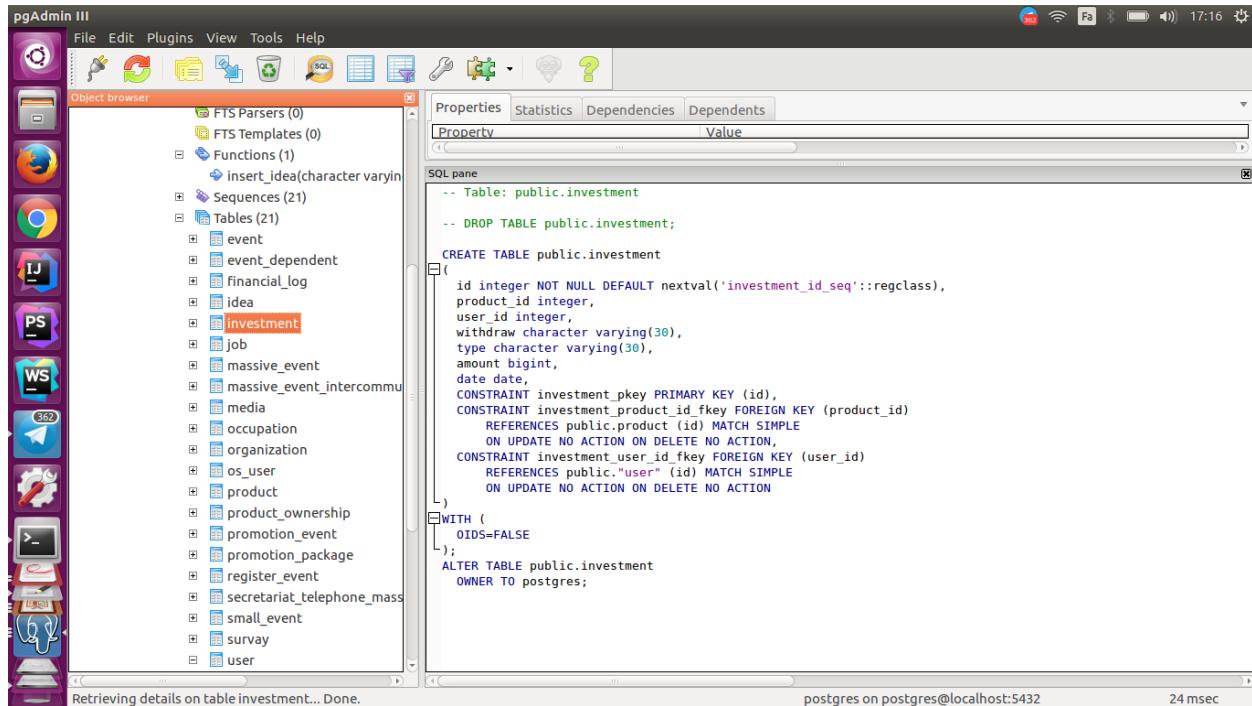
SQL pane

```
-- Table: public.idea
-- DROP TABLE public.idea;

CREATE TABLE public.idea
(
    id integer NOT NULL DEFAULT nextval('idea_id_seq'::regclass),
    product_id integer,
    CONSTRAINT idea_pkey PRIMARY KEY (id),
    CONSTRAINT idea_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.idea
    OWNER TO postgres;
REVOKE ALL ON TABLE public.idea FROM public;
REVOKE ALL ON TABLE public.idea FROM postgres;
```

Retrieving details on table idea... Done.

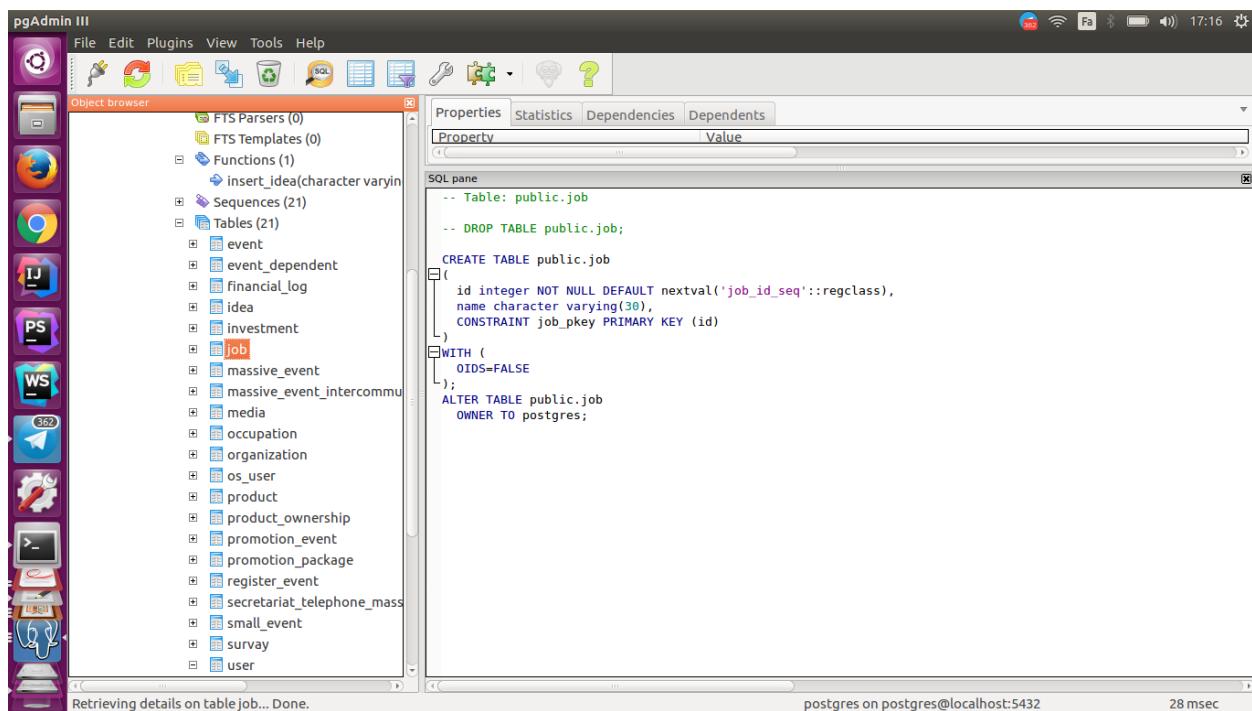
postgres on postgres@localhost:5432 1 msec



The screenshot shows the pgAdmin III interface with the following details:

- Object browser:** Shows a tree view of database objects including FTS Parsers, Functions, Sequences, Tables, and other schema items.
- Properties tab:** Shows the properties for the selected object, which is currently empty.
- SQL pane:** Displays the SQL code for creating the `public.investment` table. The code includes constraints for primary key (`id`), foreign key (`product_id` referencing `public.product`), and foreign key (`user_id` referencing `public.user`). It also includes a `WITH OIDS=FALSE` clause and an `ALTER TABLE` command to set the owner to `postgres`.

```
-- Table: public.investment
-- DROP TABLE public.investment;
CREATE TABLE public.investment
(
    id integer NOT NULL DEFAULT nextval('investment_id_seq'::regclass),
    product_id integer,
    user_id integer,
    withdraw character varying(30),
    type character varying(30),
    amount bigint,
    date date,
    CONSTRAINT investment_pkey PRIMARY KEY (id),
    CONSTRAINT investment_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT investment_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.investment
OWNER TO postgres;
```



The screenshot shows the pgAdmin III interface with the following details:

- Object browser:** Shows a tree view of database objects including FTS Parsers, Functions, Sequences, Tables, and other schema items.
- Properties tab:** Shows the properties for the selected object, which is currently empty.
- SQL pane:** Displays the SQL code for creating the `public.job` table. The code includes constraints for primary key (`id`) and a `WITH OIDS=FALSE` clause. It also includes an `ALTER TABLE` command to set the owner to `postgres`.

```
-- Table: public.job
-- DROP TABLE public.job;
CREATE TABLE public.job
(
    id integer NOT NULL DEFAULT nextval('job_id_seq'::regclass),
    name character varying(30),
    CONSTRAINT job_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.job
OWNER TO postgres;
```

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.massive_event
-- DROP TABLE public.massive_event;
CREATE TABLE public.massive_event
(
    id integer NOT NULL DEFAULT nextval('massive_event_id_seq'::regclass),
    event_id integer,
    subtitle character varying(30),
    basis_of_event character varying(50),
    registry_deadline date,
    media_deadline date,
    judgment_deadline date,
    judgment_level character varying(30),
    country character varying(30),
    city character varying(30),
    location character varying(30),
    secretariat_mobile integer,
    type_of_payment character varying(30),
    type_of_presentation character varying(30),
    CONSTRAINT massive_event_pkey PRIMARY KEY (id),
    CONSTRAINT massive_event_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.massive_event
OWNER TO postgres;
```

Retrieving details on table massive_event... Done.

postgres on postgres@localhost:5432 25 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.massive_event_intercommunity
-- DROP TABLE public.massive_event_intercommunity;
CREATE TABLE public.massive_event_intercommunity
(
    id integer NOT NULL DEFAULT nextval('massive_event_intercommunity_id_seq'::regclass),
    product_id integer,
    event_id integer,
    status_of_product character varying(30),
    point_of_product integer,
    type_of_calling character varying(30),
    CONSTRAINT massive_event_intercommunity_pkey PRIMARY KEY (id),
    CONSTRAINT massive_event_intercommunity_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.massive_event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT massive_event_intercommunity_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.massive_event_intercommunity
OWNER TO postgres;
```

Retrieving details on table massive_event_intercommunity... Done.

postgres on postgres@localhost:5432 39 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.media
-- DROP TABLE public.media;
CREATE TABLE public.media
(
    id integer NOT NULL DEFAULT nextval('media_id_seq'::regclass),
    product_id integer,
    registry_number integer,
    registry_date date,
    registration_body character varying(30),
    certifications character varying(50),
    CONSTRAINT media_pkey PRIMARY KEY (id),
    CONSTRAINT media_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.media
OWNER TO postgres;
```

Retrieving details on table media... Done.

postgres on postgres@localhost:5432 23 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.occupation
-- DROP TABLE public.occupation;
CREATE TABLE public.occupation
(
    type_of_relation character varying(30),
    favorite_subjects character varying(50),
    skills character varying(50),
    frequent_subjects character varying(50),
    workplace character varying(30),
    expertise character varying(30),
    type_of_personality character varying(30),
    id integer NOT NULL DEFAULT nextval('occupation_id_seq'::regclass),
    user_id integer,
    job_id integer,
    CONSTRAINT occupation_pkey PRIMARY KEY (id),
    CONSTRAINT occupation_job_id_fkey FOREIGN KEY (job_id)
        REFERENCES public.job (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT occupation_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.occupation
OWNER TO postgres;
```

Retrieving details on table occupation... Done.

postgres on postgres@localhost:5432 25 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.organization
-- DROP TABLE public.organization;
CREATE TABLE public.organization
(
    code integer NOT NULL DEFAULT nextval('organization_code_seq'::regclass),
    name character varying(30),
    boss_name character varying(30),
    successor_name character varying(30),
    CONSTRAINT organization_pkey PRIMARY KEY (code)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.organization
OWNER TO postgres;
```

Retrieving details on table organization... Done.

postgres on postgres@localhost:5432 38 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.os_user
-- DROP TABLE public.os_user;
CREATE TABLE public.os_user
(
    id integer NOT NULL DEFAULT nextval('os_user_id_seq'::regclass),
    user_id integer,
    os_name character varying(30),
    CONSTRAINT os_user_pkey PRIMARY KEY (id),
    CONSTRAINT os_user_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.os_user
OWNER TO postgres;
```

Retrieving details on table os_user... Done.

postgres on postgres@localhost:5432 23 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.product
-- DROP TABLE public.product;
CREATE TABLE public.product
(
    id integer NOT NULL DEFAULT nextval('product_id_seq'::regclass),
    subject character varying(30),
    description text,
    "references" character varying(50),
    key_words character varying(50),
    CONSTRAINT product_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.product
OWNER TO postgres;
```

Retrieving details on table product... Done.

postgres on postgres@localhost:5432 22 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- FTS Parsers (0)
- FTS Templates (0)
- Functions (1)
 - insert_idea(character varying)
- Sequences (21)
- Tables (21)
 - event
 - event_dependent
 - financial_log
 - idea
 - investment
 - job
 - massive_event
 - massive_event_intercommu
 - media
 - occupation
 - organization
 - os_user
 - product
 - product_ownership
 - promotion_event
 - promotion_package
 - register_event
 - secretariat_telephone_mass
 - small_event
 - survay
 - user

Properties Statistics Dependencies Dependents

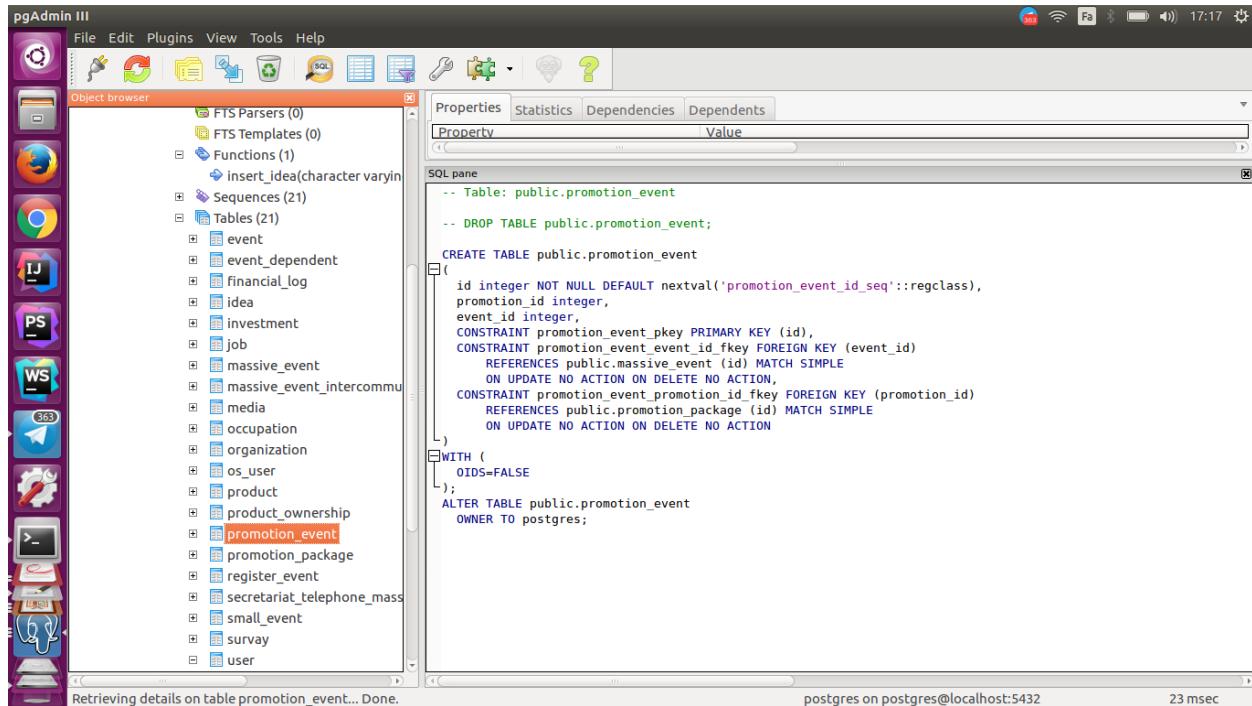
Property Value

SQL pane

```
-- Table: public.product_ownership
-- DROP TABLE public.product_ownership;
CREATE TABLE public.product_ownership
(
    id integer NOT NULL DEFAULT nextval('product_ownership_id_seq'::regclass),
    product_id integer,
    user_id integer,
    percentage integer,
    CONSTRAINT product_ownership_pkey PRIMARY KEY (id),
    CONSTRAINT product_ownership_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT product_ownership_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.product_ownership
OWNER TO postgres;
```

Retrieving details on table product_ownership... Done.

postgres on postgres@localhost:5432 34 msec



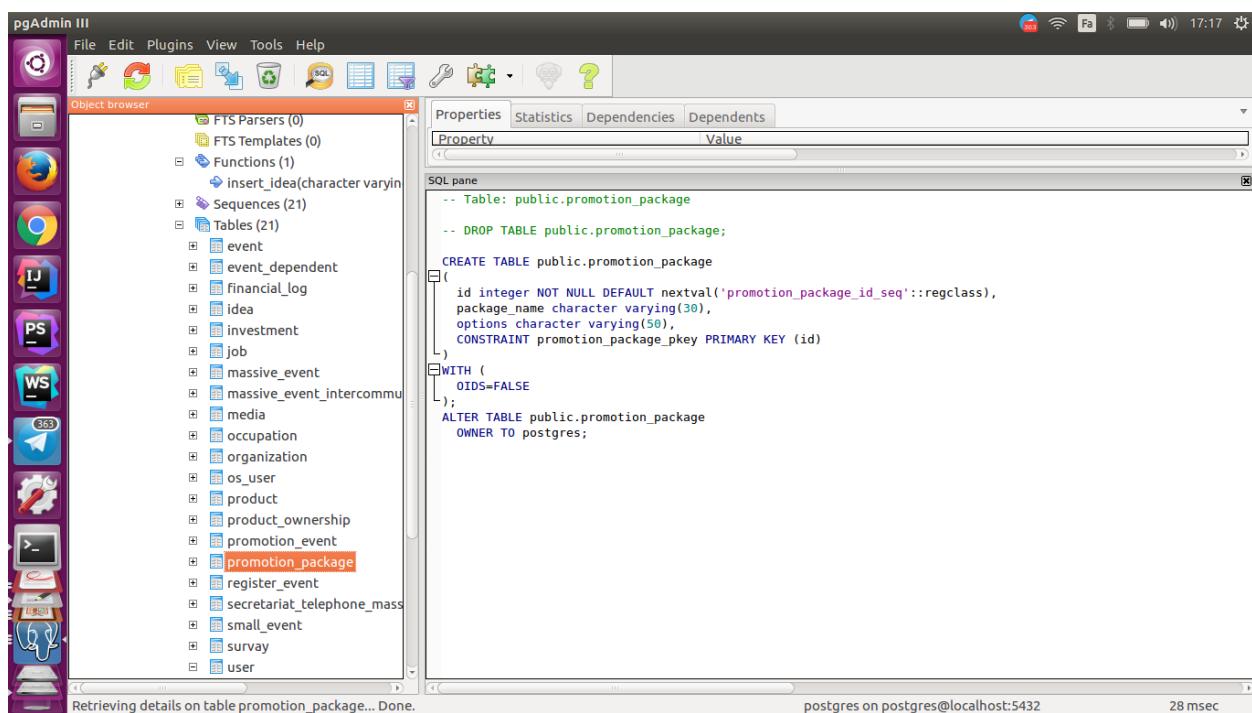
The screenshot shows the pgAdmin III interface. The left pane is the Object browser, displaying a tree structure of database objects. The 'Tables' node is expanded, showing various tables like event, event_dependent, financial_log, idea, investment, job, massive_event, massive_event_intercommu, media, occupation, organization, os_user, product, product_ownership, promotion_event, promotion_package, register_event, secretariat_telephone_mass, small_event, survay, and user. The 'promotion_event' table is selected and highlighted with a red box. The right pane is the SQL pane, which contains the SQL code for creating the table:

```

-- Table: public.promotion_event
-- DROP TABLE public.promotion_event;
CREATE TABLE public.promotion_event
(
    id integer NOT NULL DEFAULT nextval('promotion_event_id_seq'::regclass),
    promotion_id integer,
    event_id integer,
    CONSTRAINT promotion_event_pkey PRIMARY KEY (id),
    CONSTRAINT promotion_event_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.massive_event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT promotion_event_promotion_id_fkey FOREIGN KEY (promotion_id)
        REFERENCES public.promotion_package (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.promotion_event
    OWNER TO postgres;

```

The status bar at the bottom indicates "postgres on postgres@localhost:5432" and "23 msec".



This screenshot is similar to the previous one, showing the pgAdmin III interface. The Object browser on the left shows the same list of tables, with 'promotion_package' selected and highlighted by a red box. The SQL pane on the right contains the SQL code for creating the table:

```

-- Table: public.promotion_package
-- DROP TABLE public.promotion_package;
CREATE TABLE public.promotion_package
(
    id integer NOT NULL DEFAULT nextval('promotion_package_id_seq'::regclass),
    package_name character varying(30),
    options character varying(50),
    CONSTRAINT promotion_package_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.promotion_package
    OWNER TO postgres;

```

The status bar at the bottom indicates "postgres on postgres@localhost:5432" and "28 msec".

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- event_dependent
- financial_log
- idea
- investment
- job
- massive_event
- massive_event_intercommu
- media
- occupation
- organization
- os_user
- product
- product_ownership
- promotion_event
- promotion_package
- register_event**
- secretariat_telephone_massive
- small_event
- survay
- user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.register_event
-- DROP TABLE public.register_event;
CREATE TABLE public.register_event
(
    id integer NOT NULL DEFAULT nextval('register_event_id_seq'::regclass),
    event_id integer,
    user_id integer,
    CONSTRAINT register_event_pkey PRIMARY KEY (id),
    CONSTRAINT register_event_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT register_event_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.register_event
    OWNER TO postgres;
```

Retrieving details on table register_event... Done.

postgres on postgres@localhost:5432 24 msec

pgAdmin III

File Edit Plugins View Tools Help

Object browser

- event_dependent
- financial_log
- idea
- investment
- job
- massive_event
- massive_event_intercommu
- media
- occupation
- organization
- os_user
- product
- product_ownership
- promotion_event
- promotion_package
- register_event
- secretariat_telephone_massive**
- small_event
- survay
- user

Properties Statistics Dependencies Dependents

Property Value

SQL pane

```
-- Table: public.secretariat_telephone_massive_event
-- DROP TABLE public.secretariat_telephone_massive_event;
CREATE TABLE public.secretariat_telephone_massive_event
(
    id integer NOT NULL DEFAULT nextval('secretariat_telephone_massive_event_id_seq'::regclass),
    event_id integer,
    telephone integer,
    CONSTRAINT secretariat_telephone_massive_event_pkey PRIMARY KEY (id),
    CONSTRAINT secretariat_telephone_massive_event_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.massive_event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.secretariat_telephone_massive_event
    OWNER TO postgres;
```

Retrieving details on table secretariat_telephone_massive_event... Done.

postgres on postgres@localhost:5432 26 msec

The screenshot shows the pgAdmin III interface. The left pane is the Object browser, displaying a tree structure of database objects. The 'small_event' table is selected under the 'user' node. The right pane is the SQL pane, which contains the SQL code for creating the 'small_event' table. The code includes constraints for event_id, product_id, and survey_id, as well as a trigger named 'insert_trigger'.

```

-- Table: public.small_event
-- DROP TABLE public.small_event;
CREATE TABLE public.small_event
(
    id integer NOT NULL DEFAULT nextval('small_event_id_seq'::regclass),
    event_id integer,
    title character varying(30),
    date_of_triggering_date,
    date_of_notify_date,
    date_of_finish_date,
    survey_id integer,
    product_id integer,
    CONSTRAINT small_event_pkey PRIMARY KEY (id),
    CONSTRAINT small_event_event_id_fkey FOREIGN KEY (event_id)
        REFERENCES public.event (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT small_event_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.product (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT small_event_survey_id_fkey FOREIGN KEY (survey_id)
        REFERENCES public.survey (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.small_event
OWNER TO postgres;

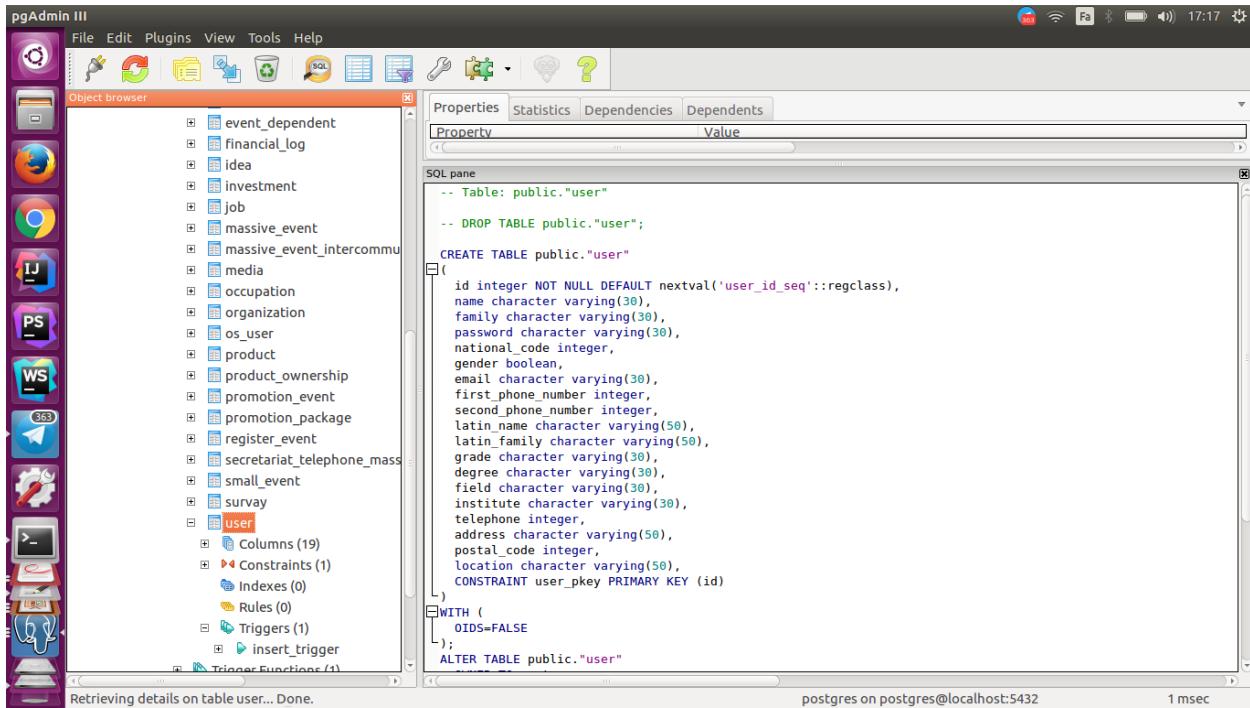
```

The screenshot shows the pgAdmin III interface again. The left pane is the Object browser, and the right pane is the SQL pane. This time, the 'survey' table is selected under the 'user' node. The SQL pane displays the creation code for the 'survey' table, which includes a primary key constraint on the 'id' column and a trigger named 'insert_trigger'.

```

-- Table: public.survey
-- DROP TABLE public.survey;
CREATE TABLE public.survey
(
    id integer NOT NULL DEFAULT nextval('survey_id_seq'::regclass),
    form_name character varying(30),
    options character varying(50),
    CONSTRAINT survey_pkey PRIMARY KEY (id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.survey
OWNER TO postgres;

```



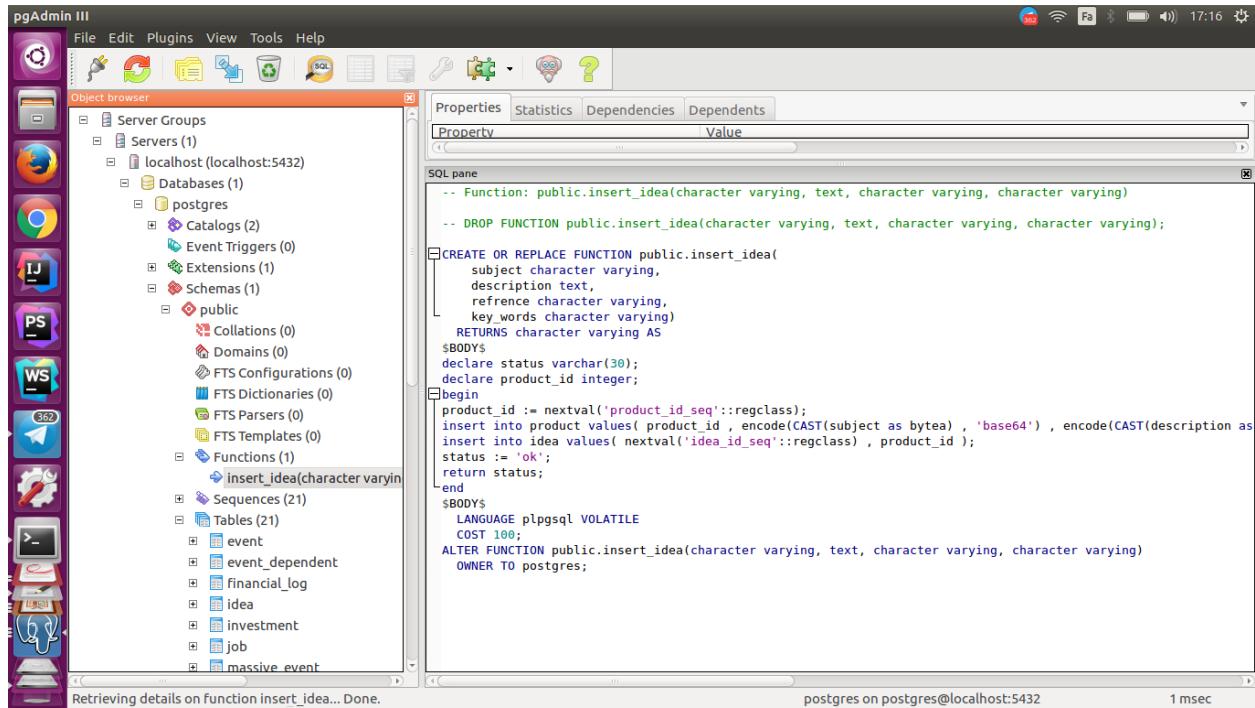
اما بعد از ساخت جداول به سراغ ایجاد شرط رمزگذاری داده های گفته شده رفتیم (ایده ها و پسورد کاربر)

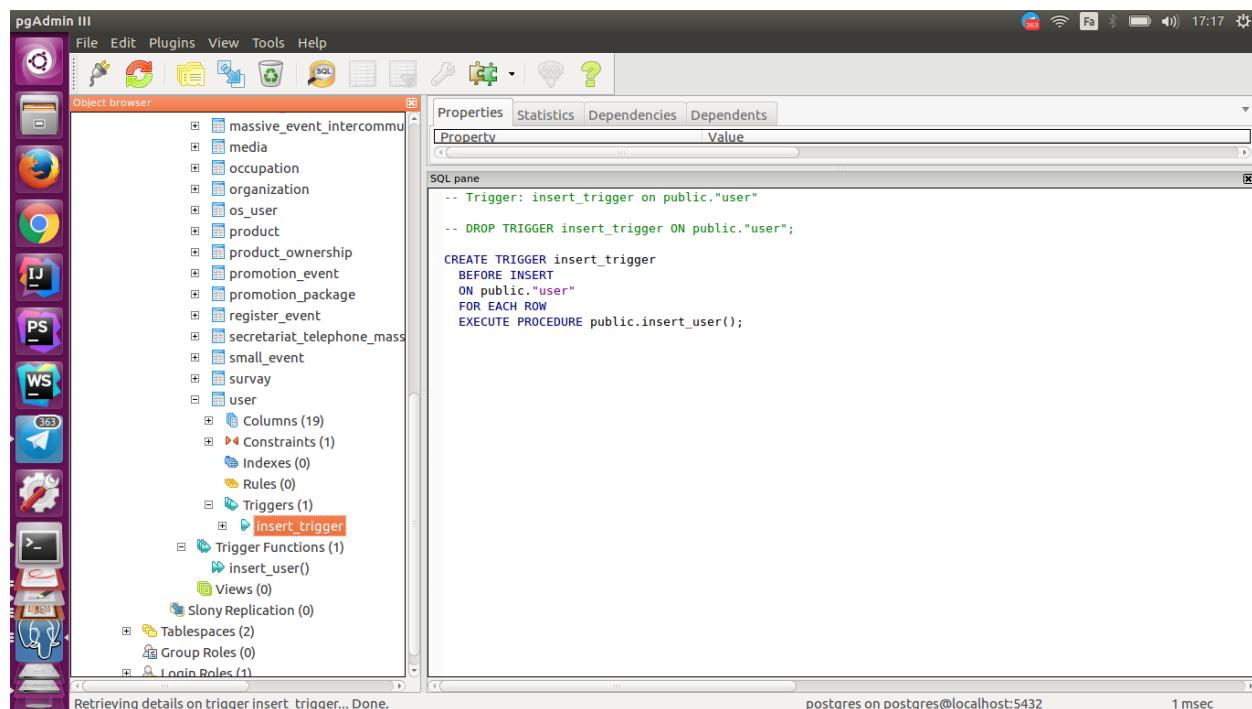
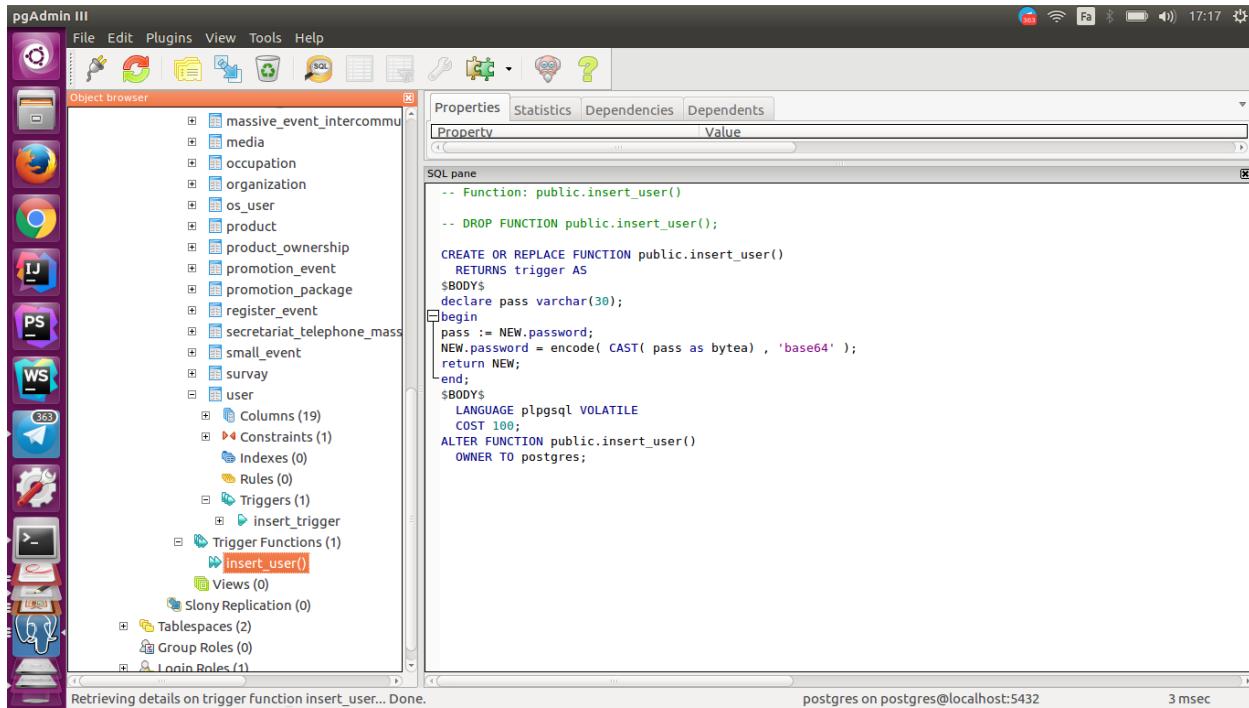
برای اینکار برای ایده ها از انجایی که از وراثت استفاده شده بود و برای اضافه کردن داده باید هم در جدول پدر و هم در جدول

فرزنده داده میشد تابع insert_idea را نوشتیم و درون همان تابع داده ها را کد میکنیم و از فرمت base64 نیز استفاده

میکنیم . اما برای جدول user از trigger استفاده میکنیم و داده پسورد را تبدیل به base64 مینماییم . البته در فاز بعدی پروژه

باید دسترسی insert بر روی جدول های product , idea بسته شود تا به صورت sql نتوان داده به آن اضافه نمود)





Edit Data - localhost (localhost:5432) - postgres - public.user

The screenshot shows a PostgreSQL client interface with the title "Edit Data - localhost (localhost:5432) - postgres - public.user". The main area displays a table with one row of data. The columns are: id [PK] serial, name character varying(30), family character varying(30), password character varying(30), national_code integer, gender boolean, email character varying(30), first_phone_number integer, second_phone_number integer, and latin char. The data for the single row is: id=1, name='mamddf', family='karimi', password='ODU0OTEzNzQ=', national_code=1, gender=false, email='', first_phone_number=85491374, second_phone_number=, latin=''. The status bar at the bottom indicates "1 row.".

Query - postgres on postgres@localhost:5432 *

The screenshot shows a PostgreSQL client interface with the title "Query - postgres on postgres@localhost:5432 *". The SQL Editor pane contains the query: "insert into public."user" (name , family , password) values ('mamddf' , 'karimi' , '85491374 ')". The Output pane shows the result: "Query returned successfully: one row affected, 22 msec execution time." and "One row affected." The status bar at the bottom indicates "Unix Ln 1, Col 100, Ch 100" and "22 msec".

Query - postgres on postgres@localhost:5432 *

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
select insert_idea('iranmetro' , 'the best app' , 'www.iranmetro.ir' , 'iran , metro' );
```

Scratch pad

Output pane

Data Output Explain Messages History

	insert_idea	character varying
1	ok	

OK. Unix Ln 1, Col 89, Ch 89 1 row. 22 msec

Edit Data - localhost (localhost:5432) - postgres - public.product

File Edit View Tools Help

No limit

	id [PK] serial	subject character varying(30)	description text	references character varying(50)	key_words character varying(50)
1	6	aXJhbmlldHJv	dGhIGjlcl30gYXBwIGluIHRyYW5zcG9ydGF0aW9uIGluZHVsJ	d3d3LmlyYW5tZXRuby5pcg==	aXJhbmlldHJv
2	7	aXJhbmlldHJv	dGhIGjlcl30gYXBw	d3d3LmlyYW5tZXRuby5pcg==	aXJhbmlldHJv
*					

Scratch pad

2 rows.