

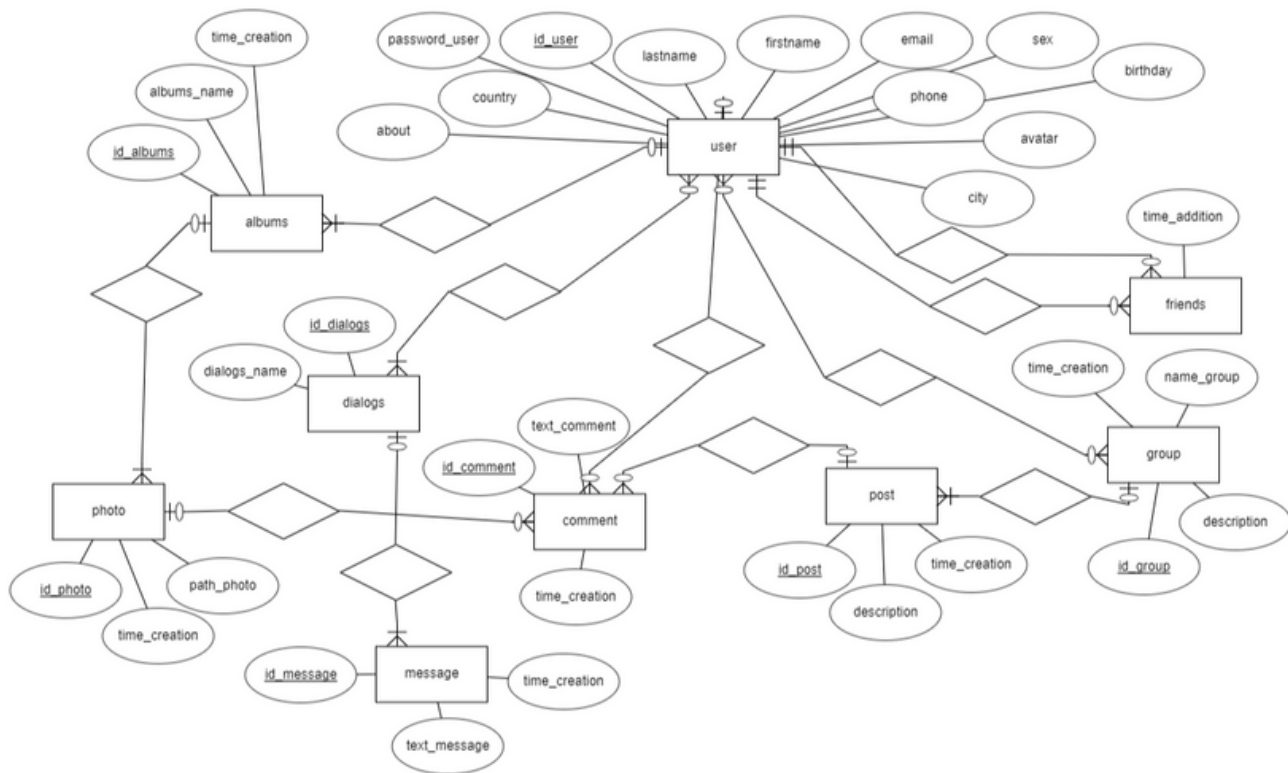


FB

# DATABÁZA FACEBOOK

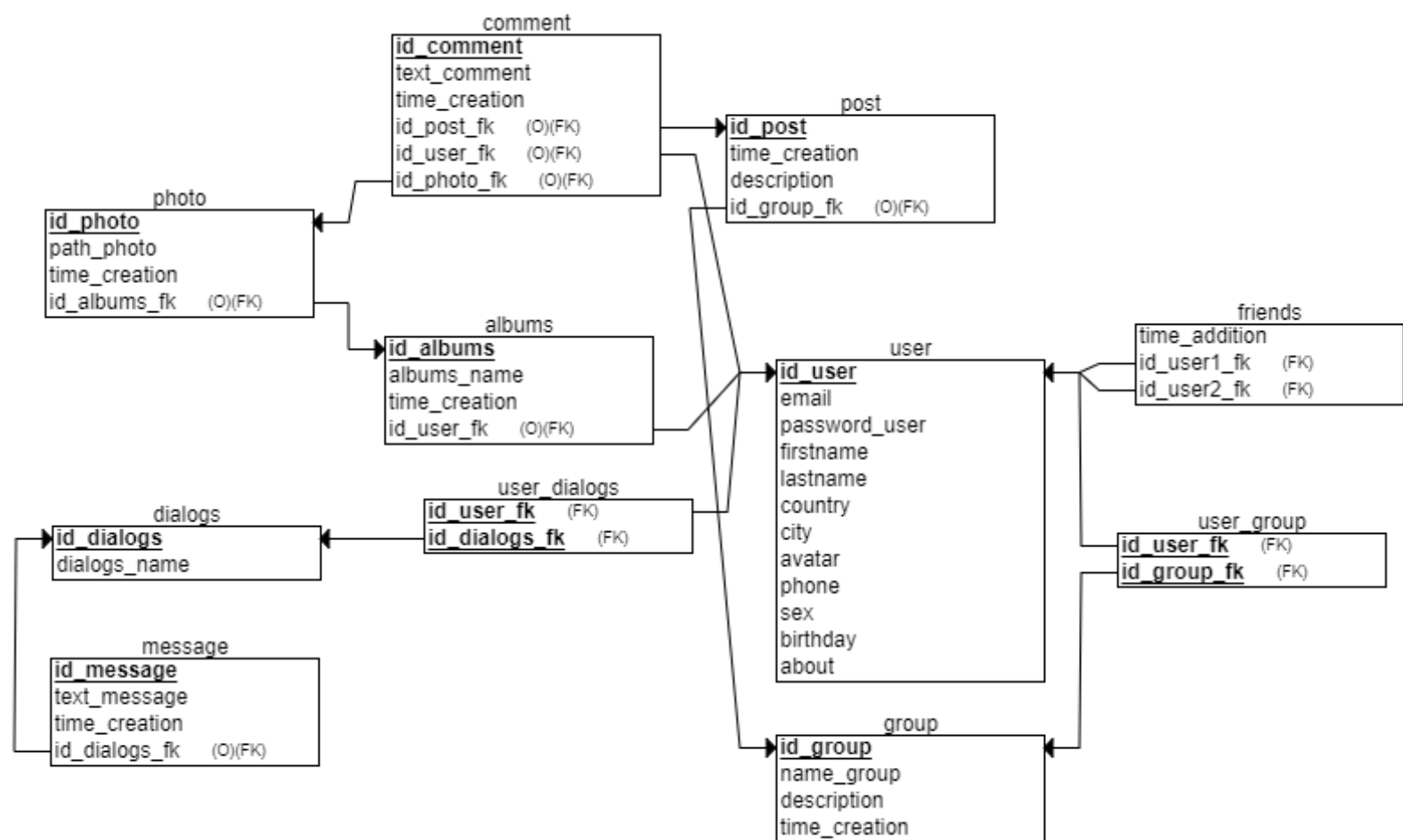
Yelizaveta Halchenko





# Entitno-relačný model

Entities	Link type	Link content
user-albums	M-1	jeden používateľ môže mať veľa albumov
user-dialogs	M-M	používateľ môže mať veľa dialógov, dialóg môže obsahovať veľa používateľov
user-comment	1-M	používateľ môže mať veľa komentárov, jeden komentár môže patriť jednému používateľovi
user-group	M-M	skupina môže obsahovať veľa používateľov, používateľ môže byť v mnohých skupinách
user-friends	1-M	používateľ môže mať veľa priateľov
comment-post	M-1	príspevok môže obsahovať veľa komentárov
comment-photo	M-1	obrázok môže obsahovať veľa komentárov
albums-photo	1-M	jeden album môže obsahovať veľa fotografií
dialogs-message	1-M	jeden dialóg môže obsahovať veľa správ
group-post	1-M	v skupine môže byť veľa príspevkov, konkrétny príspevok môže byť iba v jednej skupine



# Logický relačný model

# SQL skript

```
-- user --
create table "user"
(
    id_user    serial
        constraint user_pk
            primary key,
    email      varchar(255) not null,
    password_user varchar(16) not null,
    firstname  varchar(16) not null,
    lastname   varchar(16) not null,
    country    varchar(50),
    city       varchar(50),
    avatar     varchar(255) not null,
    phone      varchar(20) not null,
    sex        varchar(15) not null,
    birthday   date        not null,
    about      varchar(255)
);

create unique index user_id_user_uindex
on "user" (id_user);

-- friends --
create table friends
(
    id_user1_fk integer not null
        constraint friends_user_id_user_fk
            references "user",
    id_user2_fk integer not null
        constraint friends_user_id_user_fk_2
            references "user",
    time_addition date    not null
);

-- dialogs --
create table dialogs
(
    id_dialogs serial
        constraint dialogs_pk
            primary key,
    dialogs_name varchar(50) not null
);

create unique index dialogs_id_dialogs_uindex
on dialogs (id_dialogs);

-- message --
create table message
(
    id_message serial
        constraint message_pk
            primary key,
    text_message varchar(255) not null,
    time_creation timestamp    not null,
    id_dialogs_fk integer      not null
        constraint message_dialogs_id_dialogs_fk
            references dialogs
);

create unique index message_id_message_uindex
on message (id_message);

-- user_dialogs --
create table user_dialogs
(
    id_user_fk integer not null
        constraint user_dialogs_user_id_user_fk
            references "user",
    id_dialogs_fk integer not null
        constraint user_dialogs_dialogs_id_dialogs_fk
            references dialogs
);
```

# SQL skript

```
-- group --
create table "group"
(
    id_group    serial
        constraint group_pk
            primary key,
    name_group  varchar(25) not null,
    description  varchar(255),
    time_creation date      not null
);

create unique index group_id_group_uindex
on "group" (id_group);

-- user_group --
create table user_group
(
    id_user_fk integer not null
        constraint user_group_user_id_user_fk
            references "user",
    id_group_fk integer not null
        constraint user_group_group_id_group_fk
            references "group"
);

-- post --
create table post
(
    id_post    serial
        constraint post_pk
            primary key,
    time_creation timestamp not null,
    description  varchar(255) not null,
    id_group_fk integer      not null
        constraint post_group_id_group_fk
            references "group"
);

create unique index post_id_post_uindex
on post (id_post);

-- albums --
create table albums
(
    id_albums    serial
        constraint albums_pk
            primary key,
    albums_name  varchar(20) not null,
    time_creation date      not null,
    id_user_fk   integer      not null
        constraint albums_user_id_user_fk
            references "user"
);

create unique index albums_id_albums_uindex
on albums (id_albums);

-- photo --
create table photo
(
    id_photo    serial
        constraint photo_pk
            primary key,
    path_photo  varchar(255) not null,
    time_creation date      not null,
    id_albums_fk integer      not null
        constraint photo_albums_id_albums_fk
            references albums
);

create unique index photo_id_photo_uindex
on photo (id_photo);
```

# SQL skript

```
-- comment --
create table comment
(
    id_comment serial
        constraint comment_pk
            primary key,
    text_comment varchar(255) not null,
    time_creation timestamp not null,
    id_post_fk integer not null
        constraint comment_post_id_post_fk
            references post,
    id_user_fk integer not null
        constraint comment_user_id_user_fk
            references "user",
    id_photo_fk integer not null
        constraint comment_photo_id_photo_fk
            references photo
);

create unique index comment_id_comment_uindex
on comment (id_comment);
```