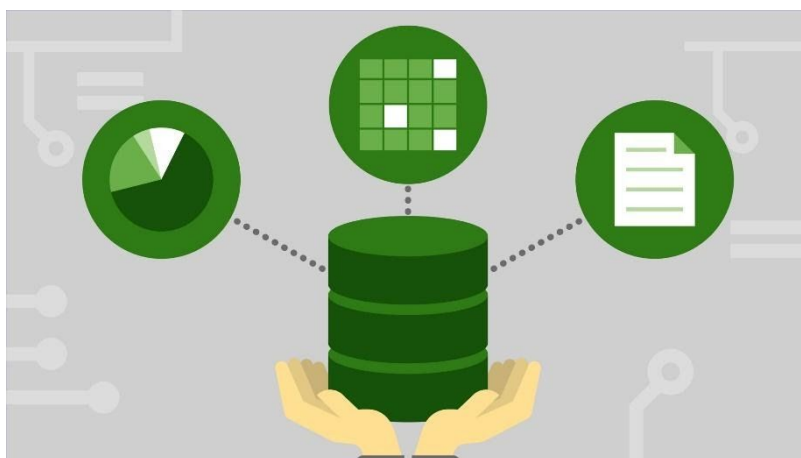


به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده  
دستور کار شماره ۱

شماره دانشجویی

۸۱۰۱۹۶۶۰۴

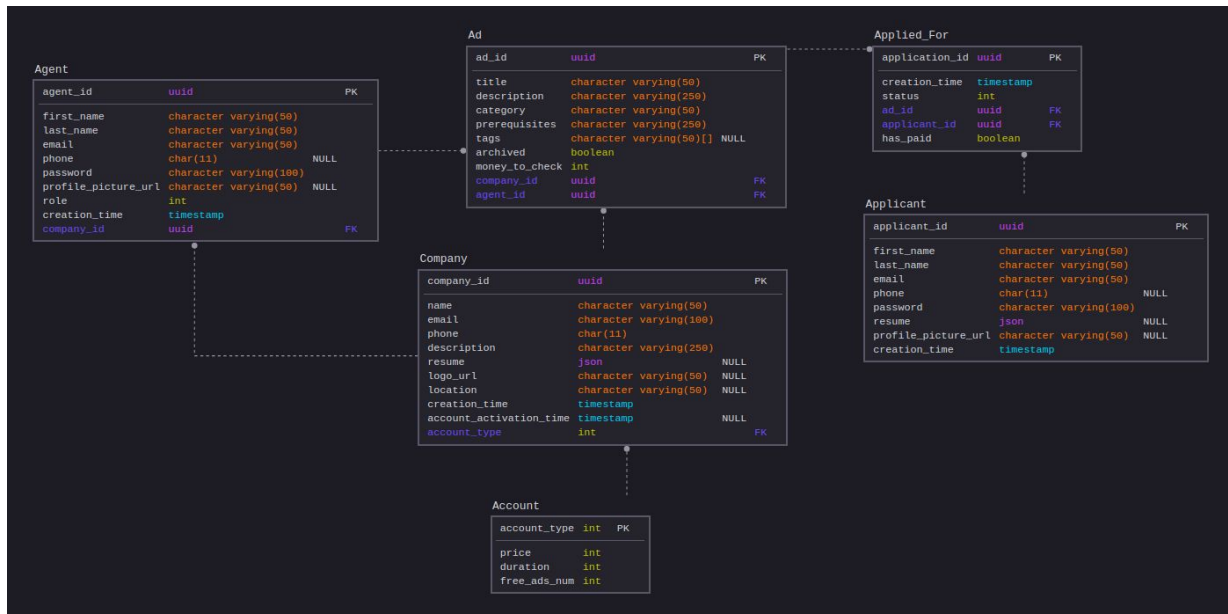
مهر ۹۹

نام و نام خانوادگی

سید پارسا حسینی نژاد

## گزارش فعالیت‌های انجام شده

ابتدا با استفاده از سایت <https://sqldbm.com> نمودار مربوط به سایت جاب اینجا را میکشیم.



توضیح مربوط به هر table:

1. **Company** : اطلاعات مربوط به یک شرکت را نگه می‌دارد و در آن تاریخ فعال کردن اکانت هم نگهداری می‌شود. علاوه بر این، به جدول **Account** نیز متصل است که در آن نوع اشتراک مشخص است.
2. **Account** : بسته‌های اشتراک مختلف را مشخص می‌کند که در آن قیمت، مدت زمان بسته و تعداد تبلیغ‌های مجانی که به واسطه‌ی آن بسته قابل انجام است نگهداری می‌شود. دقت شود اگر بسته‌ی خاصی رزرو نشده باشد، نوع دیگری از بسته در نظر می‌گیریم که قیمت آن صفر است، زمان آن نامحدود است و تعداد تبلیغ‌های مجانی نیز صفر است.
3. **Agent** : افراد مختلف مربوط به یک شرکت نظیر کارفرما، مدیر بخش فروش و ... است که در آن نوع کاربر نیز نگهداری می‌شود. هر **Agent** مربوط به یک شرکت است و رابطه‌ی یک به یک دارد.
4. **Ad** : آگهی است که مربوط به یک شرکت و یک **agent** خاص است. یک فیلد **money\_to\_check** نیز نگهداری می‌شود که نشانگر پولی است که متقاضی پرداخت می‌کند تا درخواستش حتما بررسی شود. هر تبلیغ یک **Agent** دارد و مربوط به یک شرکت خاص است.
5. **Applicant** : متقاضی شغل است که اطلاعاتش در این جدول نگهداری می‌شود و **resume** نیز در قالب JSON نگهداری می‌شود.
6. **Applied\_For** : اطلاعات مربوط به هر درخواست کاری را نگه می‌دارد که **agent** مربوطه می‌تواند **status** هر درخواست را تغییر دهد. این جدول در واقع یک **relation** بین تبلیغ و کارجو است و رابطه‌ی بین این دو در این جدول نگهداری می‌شود.

حال با استفاده از ابزار forward engineer در همین سایت، کدهای ایجاد جداول مربوطه را دریافت میکنیم که ضمیمه شده و در زیر نیز مشاهده می شود.

```
CREATE TABLE "Account"
(
  "account_type" int NOT NULL,
  "price" int NOT NULL,
  "duration" int NOT NULL,
  "free_ads_num" int NOT NULL,
  CONSTRAINT "PK_account" PRIMARY KEY ( "account_type" )
);

CREATE TABLE "Applicant"
(
  "applicant_id" uuid NOT NULL,
  "first_name" character varying(50) NOT NULL,
  "last_name" character varying(50) NOT NULL,
  "email" character varying(50) NOT NULL,
  "phone" char(11) NULL,
  "password" character varying(100) NOT NULL,
  "resume" json NULL,
  "profile_picture_url" character varying(50) NULL,
  "creation_time" timestamp NOT NULL,
  CONSTRAINT "PK_employer" PRIMARY KEY ( "applicant_id" )
);

CREATE TABLE "Company"
(
  "company_id" uuid NOT NULL,
  "name" character varying(50) NOT NULL,
  "email" character varying(100) NOT NULL,
  "phone" char(11) NOT NULL,
  "description" character varying(250) NOT NULL,
  "resume" json NULL,
  "logo_url" character varying(50) NULL,
  "location" character varying(50) NULL,
  "creation_time" timestamp NOT NULL,
  "account_activation_time" timestamp NULL,
  "account_type" int NOT NULL,
  CONSTRAINT "PK_company" PRIMARY KEY ( "company_id" ),
  CONSTRAINT "FK_44" FOREIGN KEY ( "account_type" ) REFERENCES "Account" ( "account_type" )
);

CREATE INDEX "fkIdx_44" ON "Company"
(
  "account_type"
);
```

```

CREATE TABLE "Agent"
(
    "agent_id"          uuid NOT NULL,
    "first_name"        character varying(50) NOT NULL,
    "last_name"         character varying(50) NOT NULL,
    "email"             character varying(50) NOT NULL,
    "phone"             char(11) NULL,
    "password"          character varying(100) NOT NULL,
    "profile_picture_url" character varying(50) NULL,
    "role"              int NOT NULL,
    "creation_time"     timestamp NOT NULL,
    "company_id"        uuid NOT NULL,
    CONSTRAINT "PK_employer" PRIMARY KEY ( "agent_id" ),
    CONSTRAINT "FK_55" FOREIGN KEY ( "company_id" ) REFERENCES "Company" ( "company_id" )
);

CREATE INDEX "fkIdx_55" ON "Agent"
(
    "company_id"
);

CREATE TABLE "Ad"
(
    "ad_id"            uuid NOT NULL,
    "title"            character varying(50) NOT NULL,
    "description"       character varying(250) NOT NULL,
    "category"         character varying(50) NOT NULL,
    "prerequisites"    character varying(250) NOT NULL,
    "tags"             character varying(50)[] NULL,
    "archived"         boolean NOT NULL,
    "money_to_check"   int NOT NULL,
    "company_id"       uuid NOT NULL,
    "agent_id"         uuid NOT NULL,
    CONSTRAINT "PK_ad" PRIMARY KEY ( "ad_id" ),
    CONSTRAINT "FK_67" FOREIGN KEY ( "company_id" ) REFERENCES "Company" ( "company_id" ),
    CONSTRAINT "FK_70" FOREIGN KEY ( "agent_id" ) REFERENCES "Agent" ( "agent_id" )
);

CREATE INDEX "fkIdx_67" ON "Ad"
(
    "company_id"
);

CREATE INDEX "fkIdx_70" ON "Ad"
(
    "agent_id"
);

```

```

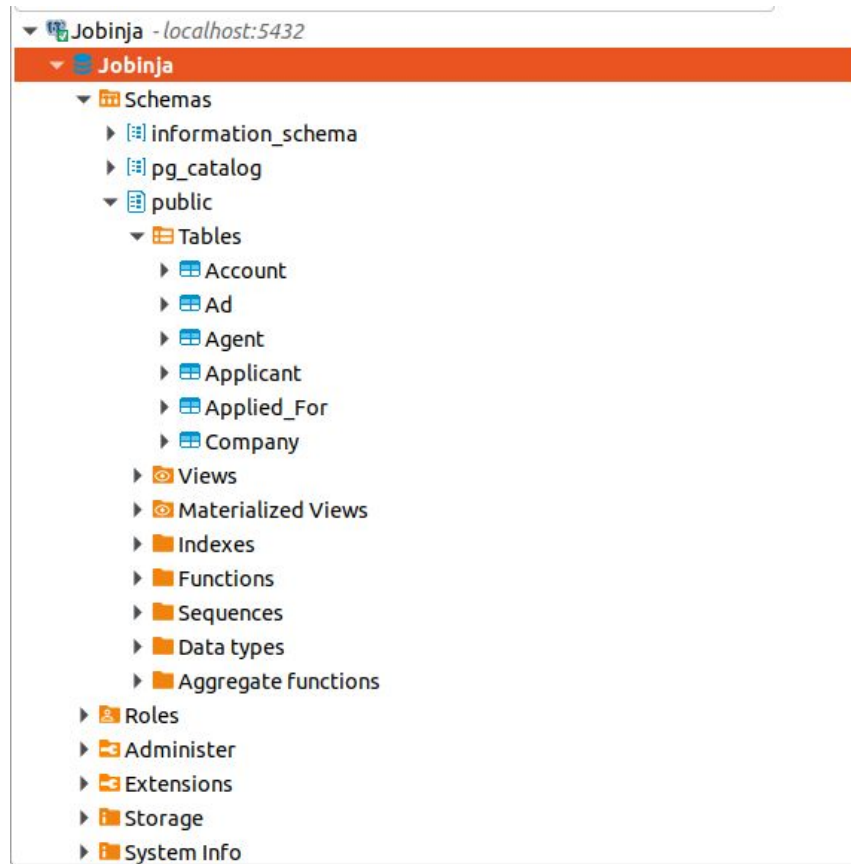
CREATE TABLE "Applied_For"
(
    "application_id"   uuid NOT NULL,
    "creation_time"    timestamp NOT NULL,
    "status"          int NOT NULL,
    "ad_id"           uuid NOT NULL,
    "applicant_id"    uuid NOT NULL,
    "has_paid"        boolean NOT NULL,
    CONSTRAINT "PK_applied_for" PRIMARY KEY ( "application_id" ),
    CONSTRAINT "FK_82" FOREIGN KEY ( "ad_id" ) REFERENCES "Ad" ( "ad_id" ),
    CONSTRAINT "FK_85" FOREIGN KEY ( "applicant_id" ) REFERENCES "Applicant" ( "applicant_id" )
);

CREATE INDEX "fkIdx_82" ON "Applied_For"
(
    "ad_id"
);

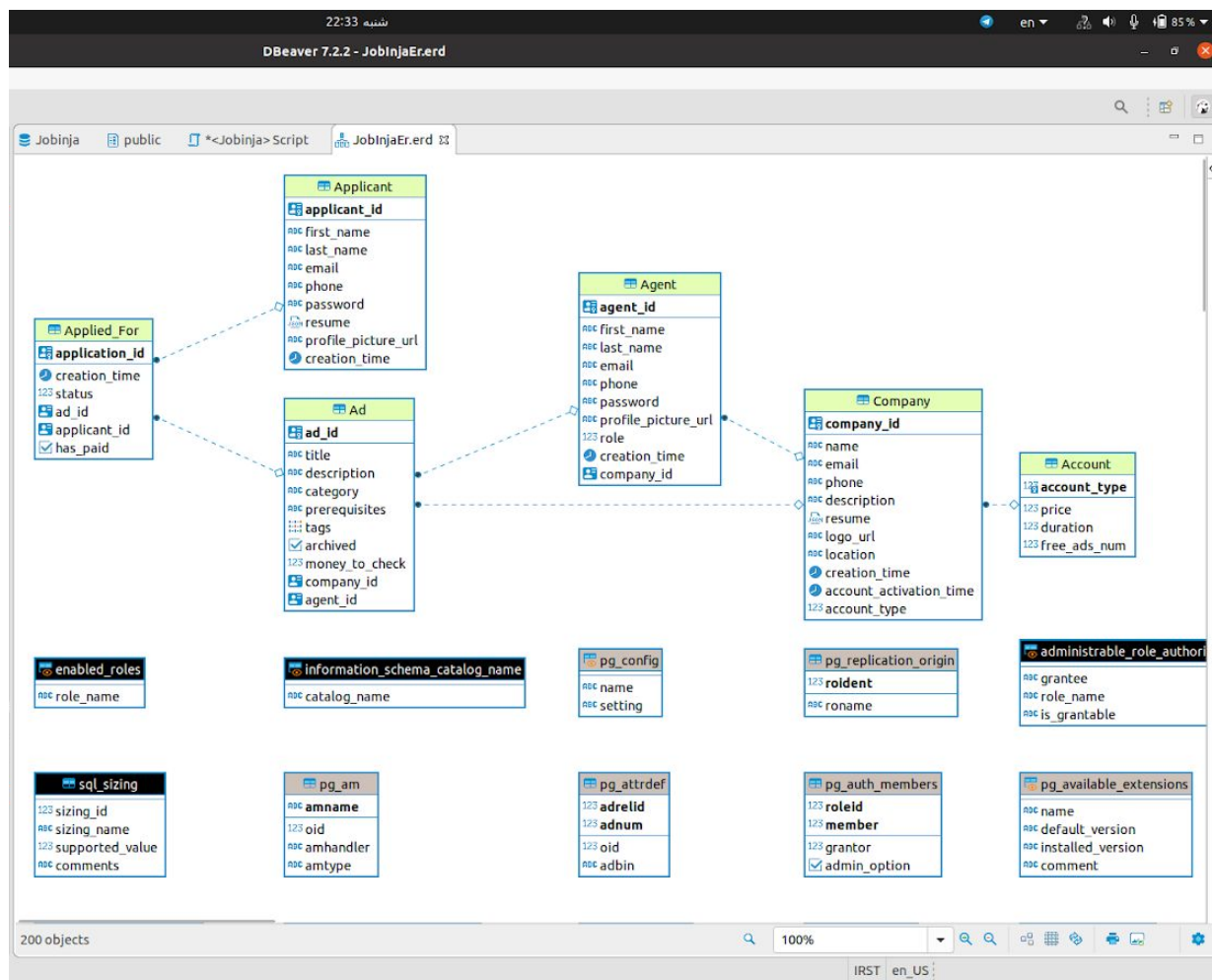
CREATE INDEX "fkIdx_85" ON "Applied_For"
(
    "applicant_id"
);

```

حال با استفاده از دستور `sudo -u postgres createdb Jobinja`، دیتابیس مربوطه را ایجاد کرده و با استفاده از نرم افزار DBeaver، به این دیتابیس متصل می‌شویم. سپس کد مربوطه را ران می‌کنیم تا جداول مربوطه مطابق شکل زیر ایجاد شوند.



حال می‌توان diagram را دوباره با استفاده از نرم افزار DBEaver استخراج کرد که به شکل زیر می‌باشد:



## مشکلات و توضیحات تکمیلی

توضیحات پروژه کافی بود و در استفاده از نرم افزار ها هم مشکلی پیش نیامد. لینک زیر می تواند در رابطه با راه اندازی postgres روی ubuntu مفید باشد:

<https://medium.com/coding-blocks/creating-user-database-and-adding-access-on-postgresql-8bfcd2f4a91e>