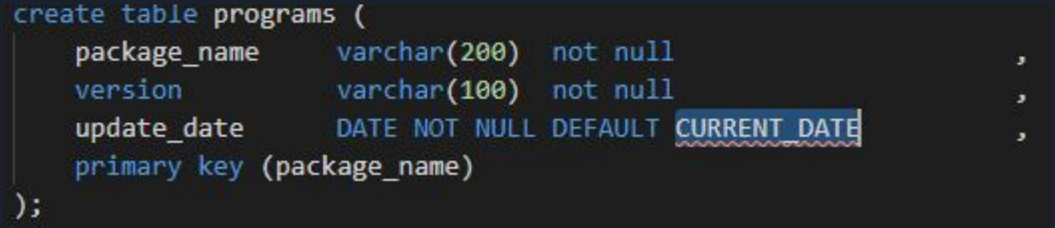
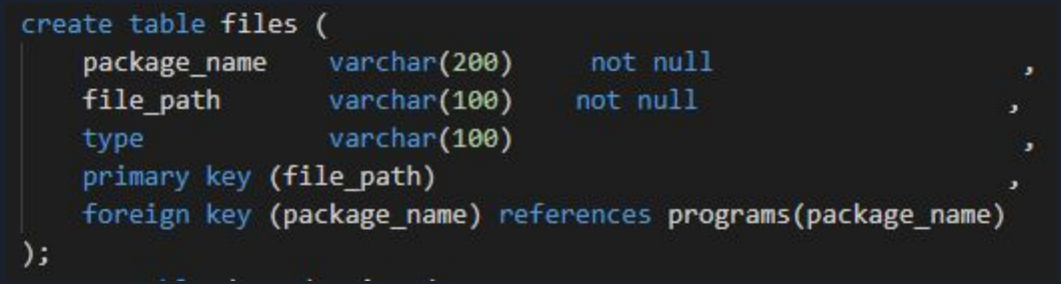
SQL menadžer paketa

Nikola Drakulić

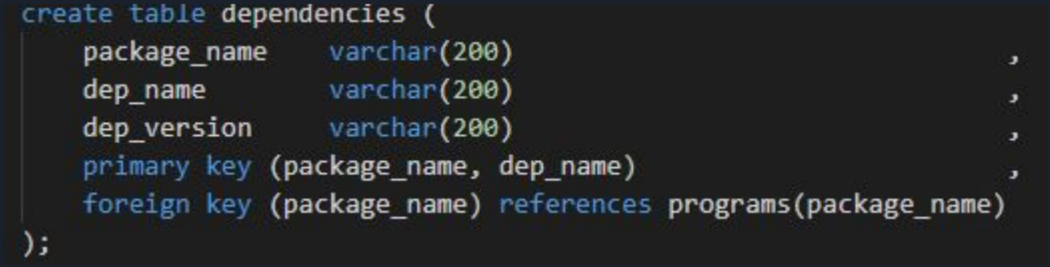
- Problem
  - Instaliranje, aržuriranje i brisanje programa na linux operativnom sistemu
  - Čuvanje fajlova i folder i verzija programa
  - Čuvanje zavisnih programa (dependency)



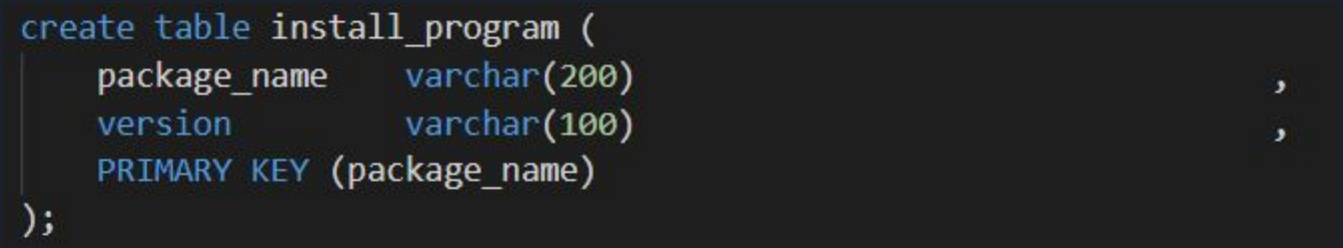
```
create table programs (
    package_name     varchar(200)  not null                        ,
    version          varchar(100)  not null                        ,
    update_date      DATE NOT NULL DEFAULT CURRENT_DATE            ,
    primary key (package_name)
);
```



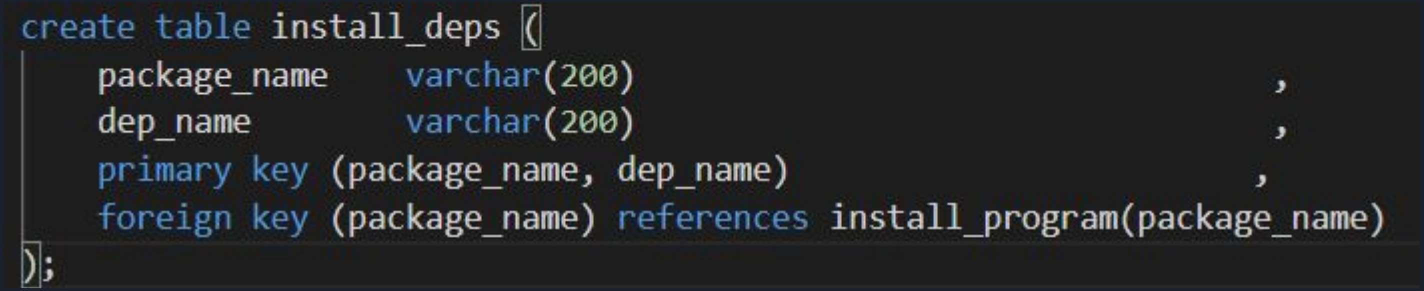
```
create table files (
    package_name    varchar(200)     not null                      ,
    file_path       varchar(100)    not null                      ,
    type            varchar(100)                                   ,
    primary key (file_path)                                       ,
    foreign key (package_name) references programs(package_name)
);
```



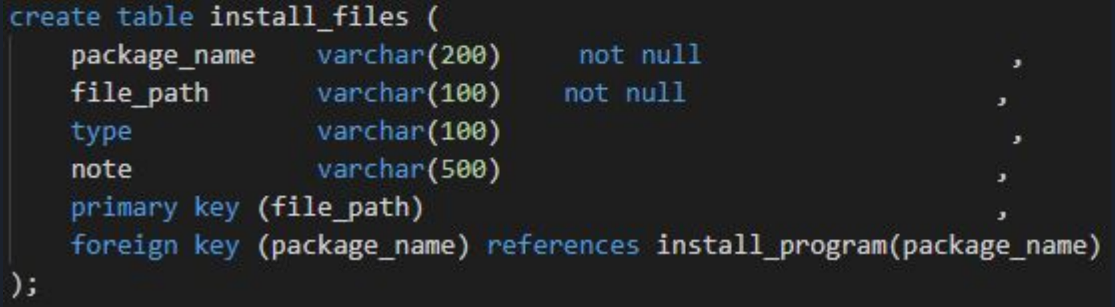
```
create table dependencies (
    package_name    varchar(200)                                  ,
    dep_name        varchar(200)                                  ,
    dep_version     varchar(200)                                  ,
    primary key (package_name, dep_name)                          ,
    foreign key (package_name) references programs(package_name)
);
```



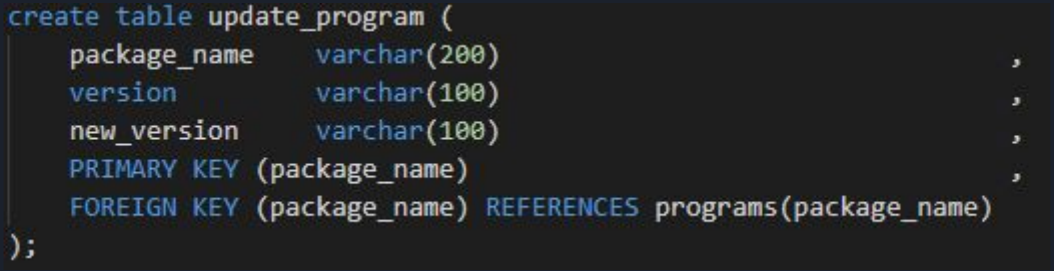
```
create table install_program (
    package_name    varchar(200)                                    ,
    version         varchar(100)                                    ,
    PRIMARY KEY (package_name)
);
```



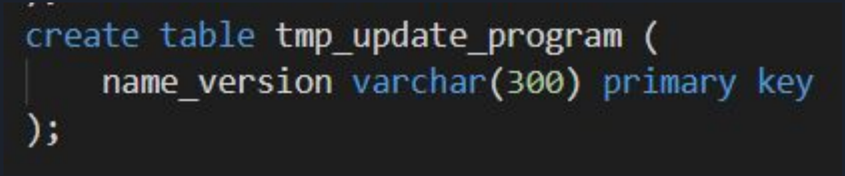
```
create table install_deps (
    package_name    varchar(200)                                    ,
    dep_name        varchar(200)                                    ,
    primary key (package_name, dep_name)                           ,
    foreign key (package_name) references install_program(package_name)
);
```



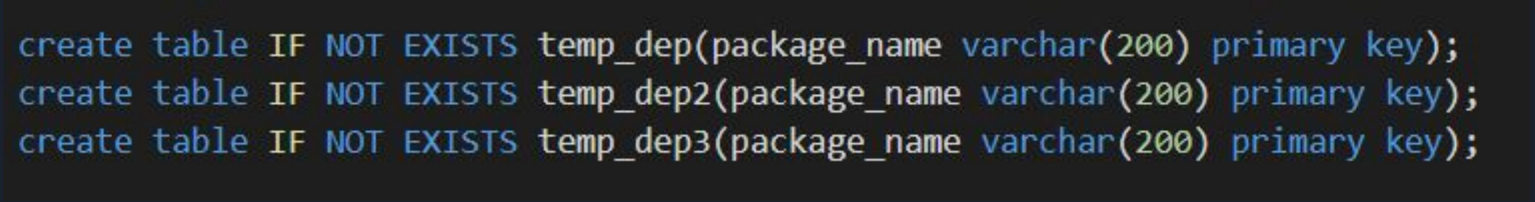
```
create table install_files (
    package_name    varchar(200)     not null                    ,
    file_path       varchar(100)    not null                    ,
    type            varchar(100)                                 ,
    note            varchar(500)                                ,
    primary key (file_path)                                     ,
    foreign key (package_name) references install_program(package_name)
);
```



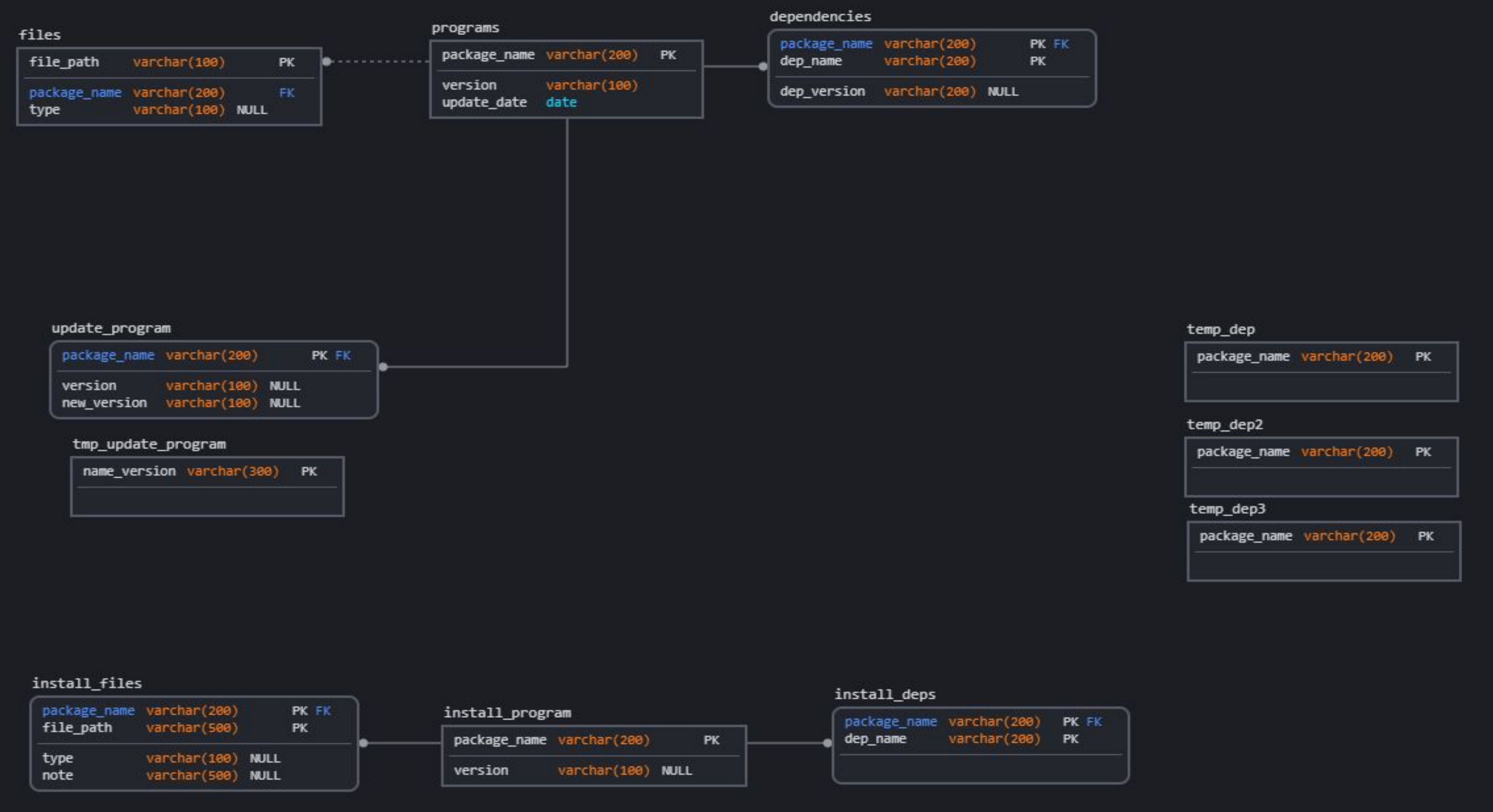
```sql
create table update_program (
    package_name    varchar(200)                                 ,
    version         varchar(100)                                 ,
    new_version     varchar(100)                                 ,
    PRIMARY KEY (package_name)                                   ,
    FOREIGN KEY (package_name) REFERENCES programs(package_name)
);
```



```sql
create table tmp_update_program (
    name_version varchar(300) primary key
);
```



```sql
create table IF NOT EXISTS temp_dep(package_name varchar(200) primary key);
create table IF NOT EXISTS temp_dep2(package_name varchar(200) primary key);
create table IF NOT EXISTS temp_dep3(package_name varchar(200) primary key);
```



files

file_path	varchar(100)		PK
package_name	varchar(200)		FK
type	varchar(100)	NULL	

programs

package_name	varchar(200)		PK
version	varchar(100)		
update_date	date		

dependencies

package_name	varchar(200)		PK FK
dep_name	varchar(200)		PK
dep_version	varchar(200)	NULL	

update_program

package_name	varchar(200)		PK FK
version	varchar(100)	NULL	
new_version	varchar(100)	NULL	

tmp_update_program

name_version	varchar(300)	PK

temp_dep

package_name	varchar(200)	PK

temp_dep2

package_name	varchar(200)	PK

temp_dep3

package_name	varchar(200)	PK

install_files

package_name	varchar(200)		PK FK
file_path	varchar(500)		PK
type	varchar(100)	NULL	
note	varchar(500)	NULL	

install_program

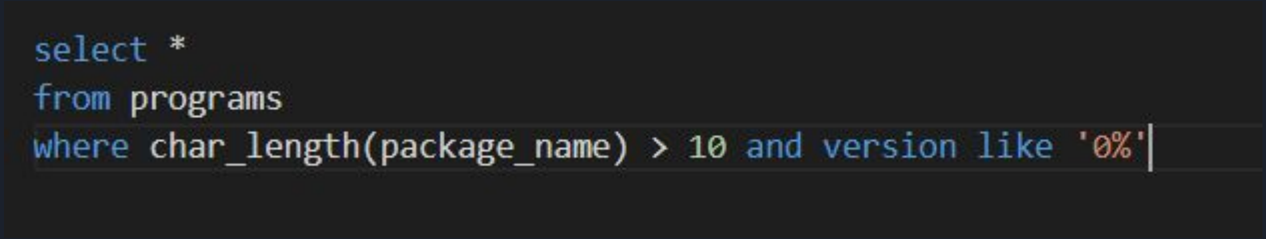
package_name	varchar(200)		PK
version	varchar(100)	NULL	

install_deps

package_name	varchar(200)	PK FK
dep_name	varchar(200)	PK

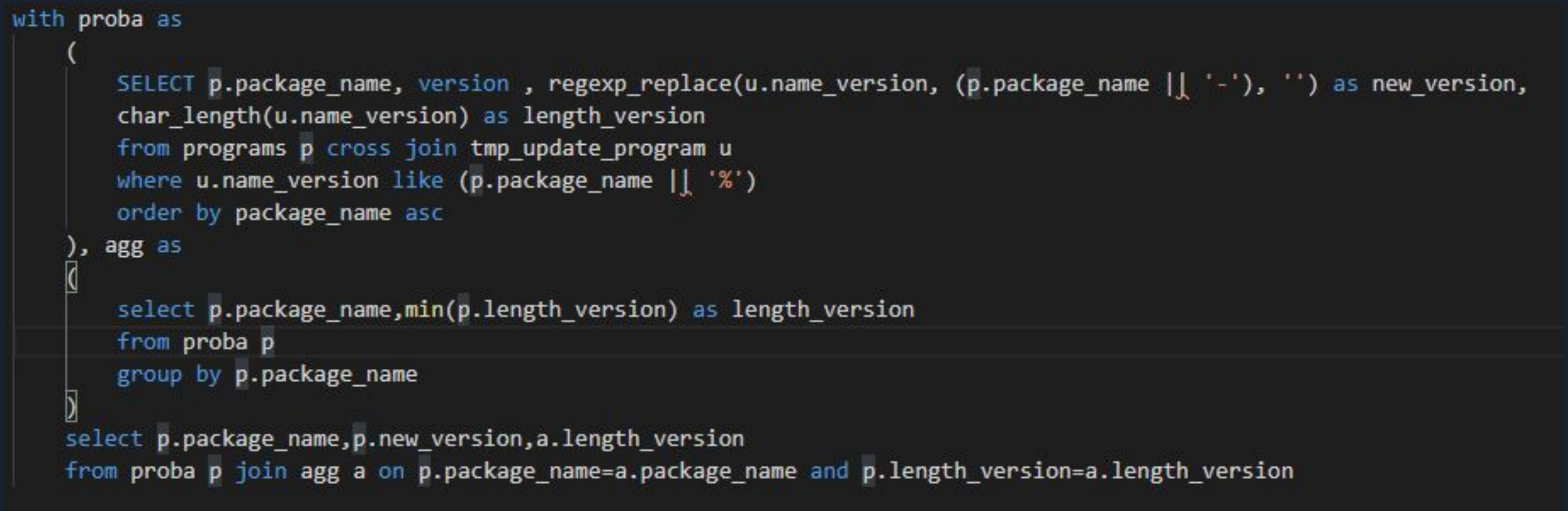
Upiti

1. Naci sve programe cije je ime duze od 10 karaktera i jos uvek su u alfa verziji (verzija im počinje sa nulom).



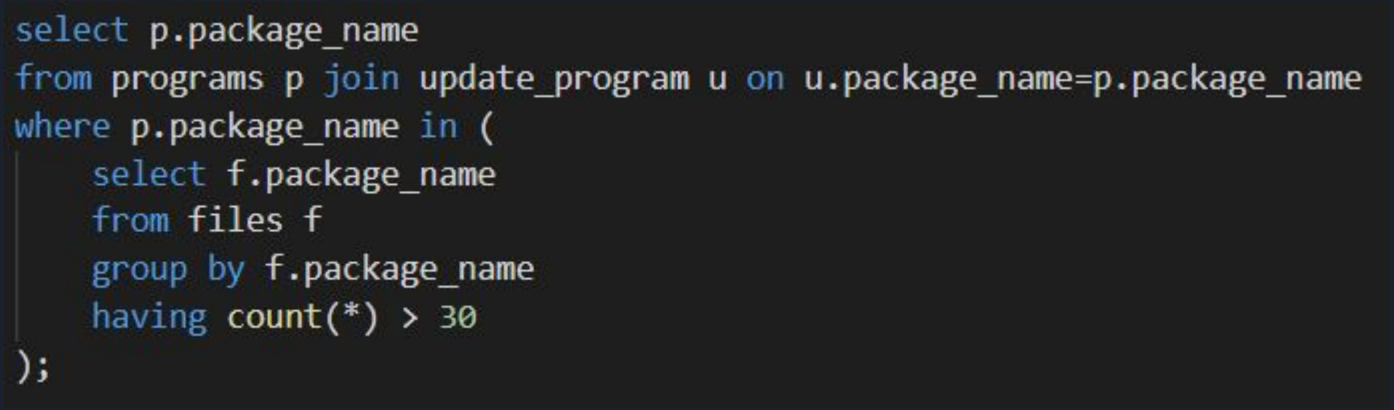
```
select *
from programs
where char_length(package_name) > 10 and version like '0%'
```

2. Name_version je u formatu package_name + “-” + version razdvojiti string package_name i version



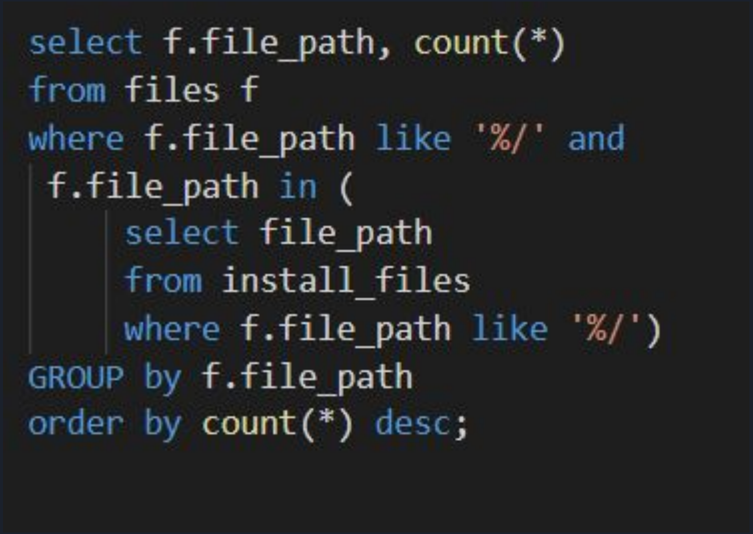
```
with proba as
    (
        SELECT p.package_name, version , regexp_replace(u.name_version, (p.package_name || '-'), '') as new_version,
        char_length(u.name_version) as length_version
        from programs p cross join tmp_update_program u
        where u.name_version like (p.package_name || '%')
        order by package_name asc
    ), agg as
    (
        select p.package_name,min(p.length_version) as length_version
        from proba p
        group by p.package_name
    )
    select p.package_name,p.new_version,a.length_version
    from proba p join agg a on p.package_name=a.package_name and p.length_version=a.length_version
```

3. Selektovati sve programe koji treba da se updetuju i imaju preko 30 fajlova



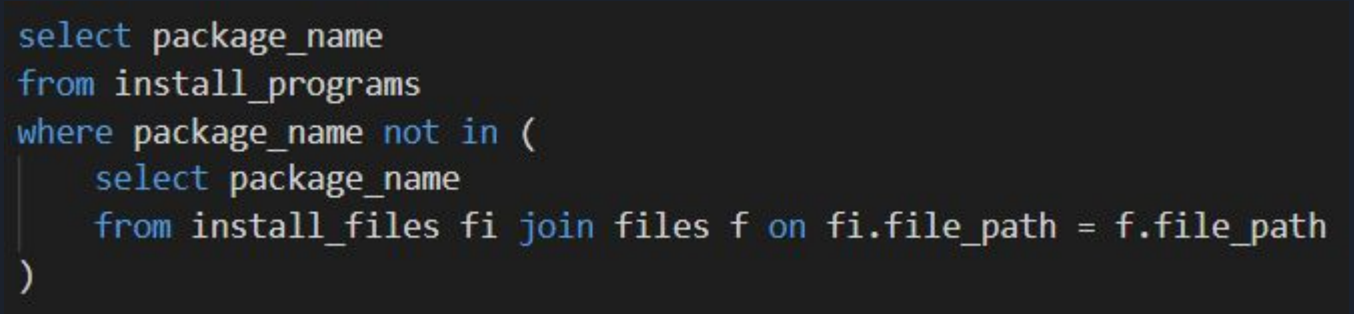
```
select p.package_name
from programs p join update_program u on u.package_name=p.package_name
where p.package_name in (
    select f.package_name
    from files f
    group by f.package_name
    having count(*) > 30
);
```

4. Prebrojati ponavljanja foldera koji se nalaze u trenutno instaliranim fajlovim i fajlovim koji treba da se instaliraju



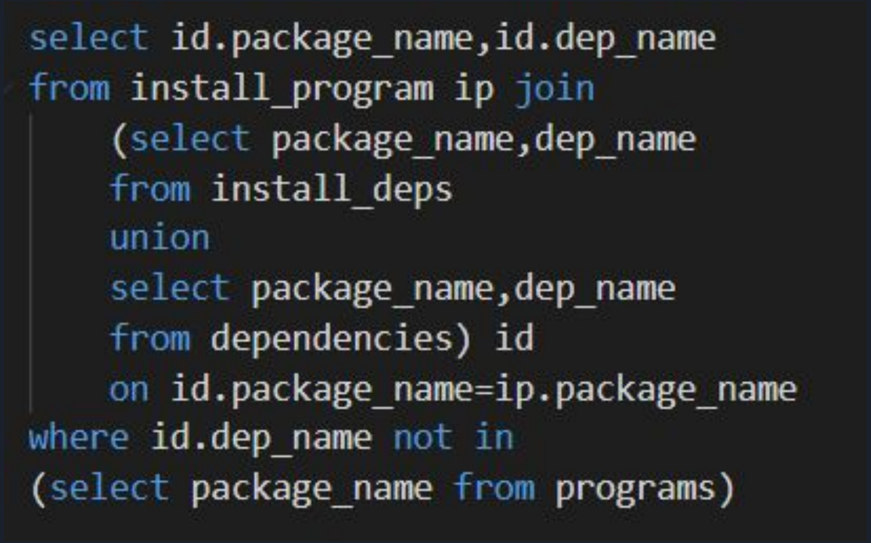
```sql
select f.file_path, count(*)
from files f
where f.file_path like '%/' and
 f.file_path in (
     select file_path
     from install_files
     where f.file_path like '%/')
GROUP by f.file_path
order by count(*) desc;
```

5. Ispisati pakete koji trebaju da se instaliraju ali se neki faljovi, koji nisu folderi, već pojavljuju u bazi



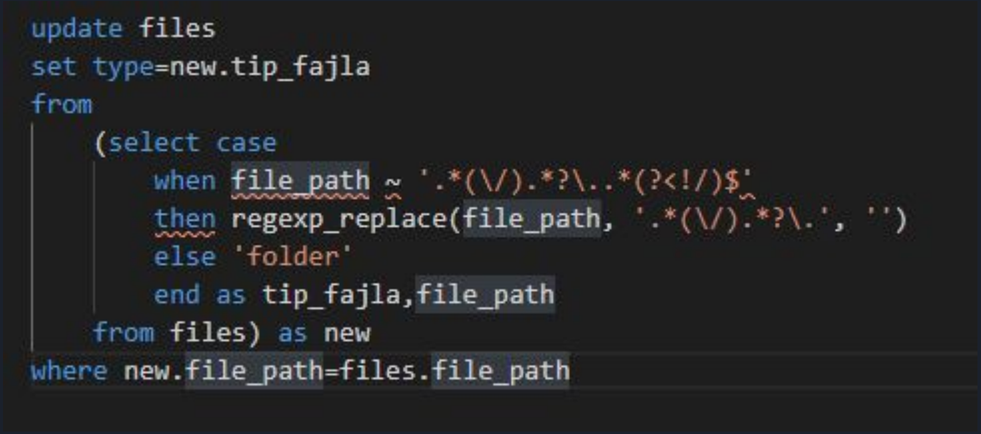
```sql
select package_name
from install_programs
where package_name not in (
    select package_name
    from install_files fi join files f on fi.file_path = f.file_path
)
```

6. Naći pakete koje je potrebno instalirati da bi paketi iz install_program mogli da se instaliraju



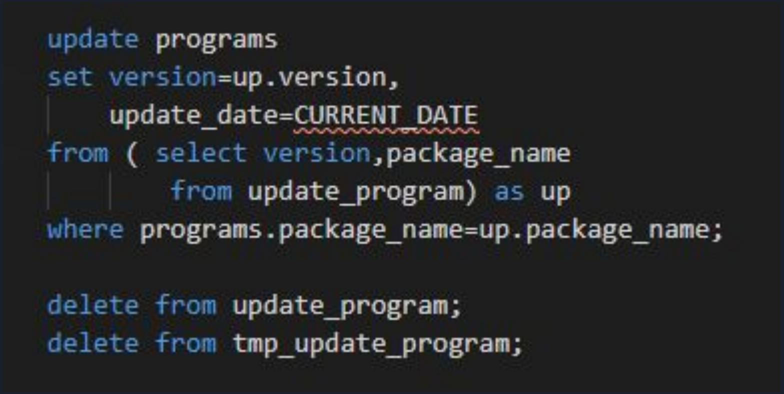
```sql
select id.package_name,id.dep_name
from install_program ip join
    (select package_name,dep_name
    from install_deps
    union
    select package_name,dep_name
    from dependencies) id
    on id.package_name=ip.package_name
where id.dep_name not in
(select package_name from programs)
```

7. Aržurirati tabelu files gde za tipove treba staviti extenziju falja ili ukoliko je folder folder



```
update files
set type=new.tip_fajla
from
    (select case
        when file_path ~ '.*(\/).*?\..*(?<!/)$'
        then regexp_replace(file_path, '.*(\/).*?\.', '')
        else 'folder'
        end as tip_fajla,file_path
    from files) as new
where new.file_path=files.file_path
```

8. Aržurirati programe i ocistiti nepotrebne podatke iz update tabele



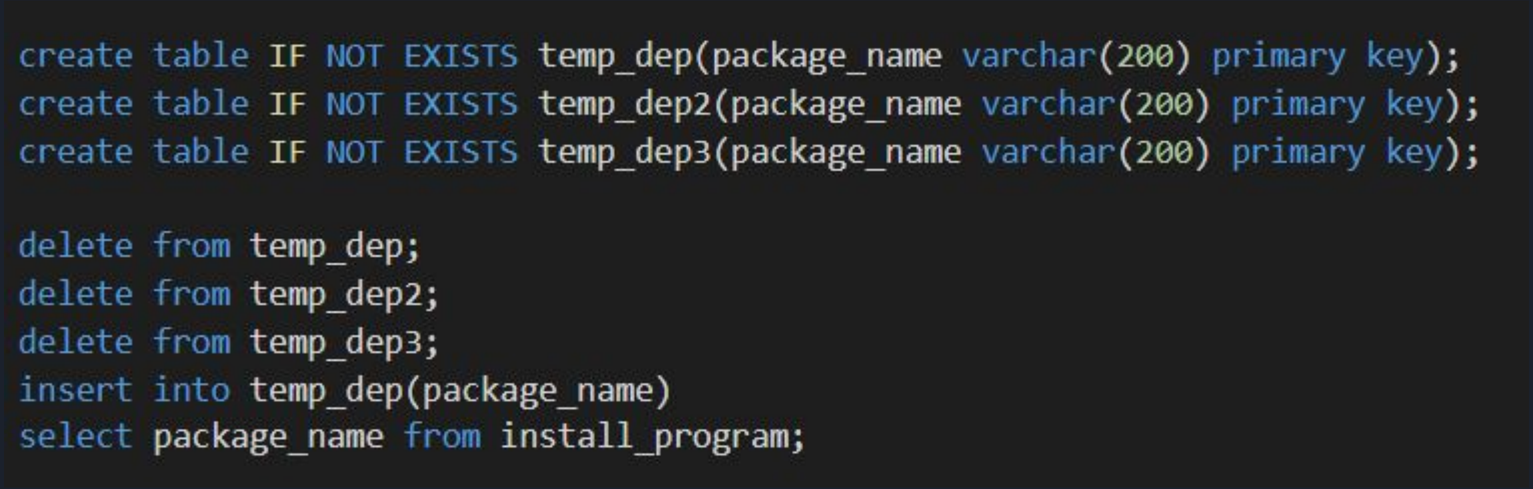
```sql
update programs
set version=up.version,
    update_date=CURRENT_DATE
from ( select version,package_name
        from update_program) as up
where programs.package_name=up.package_name;

delete from update_program;
delete from tmp_update_program;
```

9. Instalirati program ukoliko su instalirani svi dependecies i ne postoje konfliktni fajlovi potom ocistiti install tabele od nepotrebnih podataka

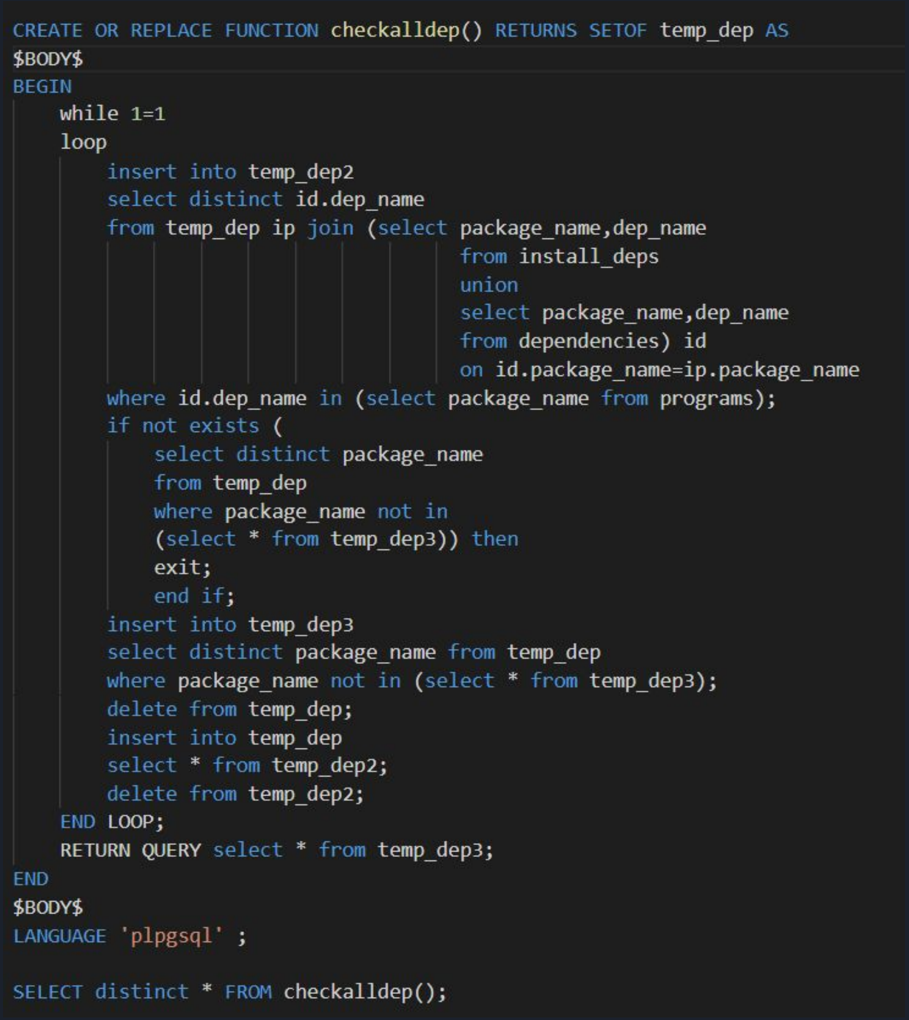
Pokazati vscode

10. Ispisati sve programe koji su potrebni da se instaliraju da bi mogli da se instaliraju svi programi iz install_package tabele



```
create table IF NOT EXISTS temp_dep(package_name varchar(200) primary key);
create table IF NOT EXISTS temp_dep2(package_name varchar(200) primary key);
create table IF NOT EXISTS temp_dep3(package_name varchar(200) primary key);

delete from temp_dep;
delete from temp_dep2;
delete from temp_dep3;
insert into temp_dep(package_name)
select package_name from install_program;
```



```
CREATE OR REPLACE FUNCTION checkalldep() RETURNS SETOF temp_dep AS
$BODY$
BEGIN
    while 1=1
    loop
        insert into temp_dep2
        select distinct id.dep_name
        from temp_dep ip join (select package_name,dep_name
                                      from install_deps
                                      union
                                      select package_name,dep_name
                                      from dependencies) id
                                      on id.package_name=ip.package_name
        where id.dep_name in (select package_name from programs);
        if not exists (
            select distinct package_name
            from temp_dep
            where package_name not in
            (select * from temp_dep3)) then
            exit;
            end if;
        insert into temp_dep3
        select distinct package_name from temp_dep
        where package_name not in (select * from temp_dep3);
        delete from temp_dep;
        insert into temp_dep
        select * from temp_dep2;
        delete from temp_dep2;
    END LOOP;
    RETURN QUERY select * from temp_dep3;
END
$BODY$
LANGUAGE 'plpgsql' ;

SELECT distinct * FROM checkalldep();
```

Nedostaci

- Ne može da postoji isti program dve različite verzije
- Ne proverava se fajlovi kod aržuriranja programa

Hvala na pažnji,
Pitanja?