Adugare pizzerie:

Import csv

Adaugare resursa

1. Alege tip res
2. Formular specializat [daca e cazul]

[adaugare user]

Adaugare resursa

Adaugare resursa unui user

RES\_COLUMNs, Id\_user

specializaation\_RES\_COLUMNs

Dep( id\_user, RES\_COLUMNs)

Dep (RES\_COLUMNs, specializaation\_RES\_COLUMNs)

formularul de specializare apare dupa formularul RES

# Cont\_Useri

## Stergere User implica

* **Atentie** La stergerea unui utilizator resurselePersonale asociate cu acesta vor fi asociate cu null.

(FK **set null** resPers:id\_user).

* La stergerea unui utilizator lista cu resursele sale favorite va fi stearsa. (FK **delete cascade** resPers: id\_user)

# RESURSA

## ADAUGARE Resursa METDOA 1 CRUD si aplicatie cu/fara utilizator

**INSERT INTO resurse r(id\_res, r.obiect) VALUES (seq\_resurse.nextval, pizzerie( … , ‘0750000’ , ‘pizzahot.ro’);**

**+TRIGGER** add\_res\_id\_user

After Insert

ON resurse

Begin

Insert into resPersonale (id\_user , id\_res) values (null, :new.id\_res)

End

--------------------------

In caz ca insertul se face din aplicatie se face un update in resPersonale din null devine id-ul userului

**UPDATE resPersonale set id\_user = gigel where id\_res= VAL**

## ADAUGARE Resursa METDOA 2 sql command

**INSERT INTO resurse r(id\_res, r.obiect) VALUES (seq\_resurse.nextval, pizzerie( … , ‘0750000’ , ‘pizzahot.ro’);**

**Insert into resPersonale (id\_user , id\_res) values (22, seq\_resurse.currval);**

## STERGERE Resursa

**Delete resurse\_oop where id\_res= 111;**

**Delete resurse\_oop where id\_user= 22;**

**Delete resurse\_oop where id\_user= 22 and id\_res = 111;**

Stergere resPers:

(FK delete cascade resPers: id\_res)

**+TRIGGER care se declanseaza la stergerea din resPers si sterge din resurse\_oop**

Stergere resFav:

(FK delete cascade resFav: id\_res, user)

## Update Resursa

# View-uri

--creem niste viewuri

Toate resursele

Pe tipuri de resurse

ResursePersonale [join pt mai multe informatii]

ResurseFavorite[join pt mai multe informatii]

~~Conturi~~

~~Tari~~

~~Categ~~

# Popularitate

--o recalc\_popularitate()

O procedura care odata executata aduce resursele la starea cureta( include tratare de exceptii)

**+Trigger**

After

Insert on resfav

* 1. popularitate(:new.id\_res)+=1

Update on resfav

1. Daca fac update pe id\_user nu se schimba nimic
2. Daca fac update pe id\_res se schimba
3. popularitate(:new.id\_res)+=1
4. popularitate(:old.id\_res)-=1

Delete on resfav

1. Daca sterg un id\_res atunci popularitate(:old.id\_res)-=1

Operatiile pe alte tabele se traduc in final tot prin acestea 3.