

Procedure Ulazno/Izlazni Parametri

Primer 1

- Potrebno je napisati proceduru koja će da proveri da li sa unetog računa može da se umanjí željeni iznos i ako na datom računu ima dovoljno novca potrebno je ukupan iznos umanjiti za tu vrednost.
- Procedura treba da vrati status transakcije
 - True ako je izvršena
 - False ako nije izvršena

```
create table tblbalance (id int auto_increment primary key, balance decimal(7,2))
```

```
insert into tblbalance(balance) values (8000) , (2000), (1578.45)
```

```
select * from tblbalance
```

Procedure Ulazno/Izlazni Parametri

Primer

delimiter //

```
create procedure p_withdrawal(in id_usr int, in balance_val decimal(7,2) , out stat bool)
begin
  declare current_balance decimal(7,2) default 0.0;
  select balance into current_balance from tblbalance where id=id_usr;
  if current_balance >= balance_val
  then
    set stat=true;
    update tblbalance set balance=balance-balance_val where id=id_usr;
  else
    set stat=false;
  end if;
end//
```

delimiter ;

Procedure Ulazno/Izlazni Parametri

Primer

- Proceduri saopštavamo dva ulazna parametra i jedan izlazni parametar
 - id računa sa kojeg oduzimamo iznos *id_usr*
 - vrednost koja se skida sa računa *balance_val*
 - *stat* je izlaz na osnovu kojeg određujemo da li je iznos umanjen ili nije umanjen sa računa i vraća true ili false

```
create procedure p_withdrawal(in id_usr int, in balance_val decimal(7,2) , out stat bool)
```

Procedure Ulazno/Izlazni Parametri

Primer

- U proceduri definišemo lokalnu promenjivu *current_balance* u koju upisujemo iznos na računu sa koga želimo da skinemo novac

```
declare current_balance decimal(7,2) default 0.0;
```

```
select balance into current_balance  
from tblbalance  
where id=id_usr;
```

Procedure Ulazno/Izlazni Parametri

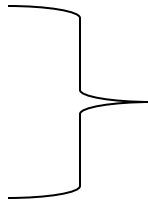
Primer

- Proveravamo da li na unetom računu ima dovoljno novca
 - Ukoliko ima novca više od iznosa koji se skida, iznos na računu se umanjuje za tu vrednos i u promenjivu status se upisuje true
 - Ukoliko ima manje novca na računu od iznosa koji se skida tada se promenjiva status menja u false

```
if current_balance >= balance_val
then
    set stat=true;
    update tblbalance set balance=balance-balance_val where id=id_usr;
else
    set stat=false;
end if;
```

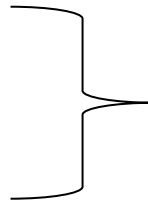
Pozvianje Procedure Primer

```
call p_withdrawal(2,100, @stat);
```



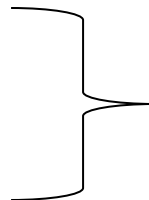
- Pozivamo proceduru saopštavajući joj broj računa, iznos koji se povalči i promenjivu gde čuvamo status

```
select @stat;
```



- Ukoliko je status 1, prosleđeni iznos je skinut sa računa a ukoliko je 0, prosleđeni iznos nije skinut sa računa

```
select * from tblbalance;
```



- Prikazujemo stanje tabele sa iznosima na računima

Pozvianje Procedure – Transakcije Zaključavanje

Primer

```
create procedure p_withdrawal(in id_usr int, in balance_val decimal(7,2) , out stat bool)
```

```
Begin
```

```
    declare current_balance decimal(7,2) default 0.0;
```

```
Start transaction;
```

```
    select balance into current_balance from tblbalance where id=id_usr for update;
```

```
        if current_balance >= balance_val
```

```
            then
```

```
                set stat=true;
```

```
                update tblbalance set balance=balance-balance_val where id=id_usr;
```

```
            else
```

```
                set stat=false;
```

```
            end if;
```

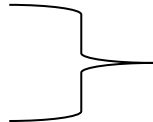
```
Commit;
```

```
end//
```

Upravljanje Greškama

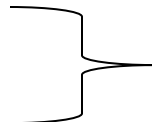
Primer

```
show errors;
```



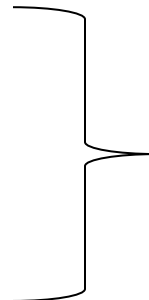
- Ukoliko se pri izvršenju komande javi greška, ova komanda nam ispisuje šta je problem

```
show warnings;
```



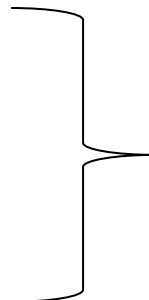
- Ukoliko se pri izvršenju komande javi upozorenje, ova komanda nam ispisuje šta je problem

```
declare exit handler for sqlexception  
begin  
    show errors;  
end;
```



- Kod za upravljanje greškama se unosi u proceduru i ukoliko se javi greška izlazi se iz procedure i prikazuje razlog javljanja greške

```
declare exit handler for sqlwarning  
begin  
    show warnings;  
end;
```



- Kod za upravljanje upozorenjima

Procedura za Generisanje podataka

- Potrebno je napisati proceduru koja će popuniti tabelu slučajno izabranim podacima.
- Proceduri je potrebno saopštiti samo broj redova koje želimo da dodamo u tabeli.

```
create table randomusers(id int auto_increment primary key,  
                           name varchar(30),  
                           created_at date,  
                           activation bool);
```

Procedura za Generisanje podataka

```
delimiter $$
create procedure insertdata (in broj int)
begin
    declare broj_redova int;
    declare ime varchar(30);
    declare datum date default date(now());
    declare stanje bool;
    select broj into broj_redova;
    while broj_redova > 0 do
        set ime = concat('User',broj_redova);
        set datum = date(now()) - interval floor(1000*rand()) day;
        set stanje = round(rand());
        insert into randomusers(name,created_at,activation) values (ime,datum,stanje);
        set broj_redova=broj_redova-1;
    end while;
end$$
delimiter ;
```

Poziv Procedure za generisanje 100 redova

```
call insertdata(100);
```