SP örnekleri

1. Belirli bir departmandaki çalışanların sayısını döndüren bir prosedür yaz.

- Parametre olarak department_id almalı.
- Belirtilen departmandaki çalışanların sayısını döndürmeli.

```
create or replace procedure sel_workernumber_sp(
    p_department_id in number,
    p_worker_count out number
) as
begin
    select count(*)
    into p_worker_count
    from employees
    where department_id= p_department_id;

end sel_workernumber_sp;
```

```
DECLARE
    v_count NUMBER;
BEGIN
    sel_workernumber_sp(50, v_count);
    DBMS_OUTPUT.PUT_LINE('Çalışan Sayısı: ' || v_count);
END;
/
```

Yeni Maaş Spsi

```
create or replace PROCEDURE yeni_maas_sp(
    p_id IN VARCHAR2,  -- Virgül hatası düzeltildi
    p_maas IN NUMBER DEFAULT 0
) AS
BEGIN
    UPDATE jobs
    SET min_salary = min_salary * (1 + p_maas / 100)
   WHERE job_id = p_id;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Hata oluştu: ' || SQLERRM);
END yeni_maas_sp;
```

```
BEGIN
    yeni_maas_sp('AD_VP', 10);
END;
/
```

getAccountBalance_udf

```
create or replace function getAccountBalance_udf(
    p_customer_id number,
    p_account_id number default null

) return number is v_balance number:=0;

begin

    if p_account_id is not null then
        select nvl(balance, 0)
        into v_balance
        from accounts
        where customer_id= p_customer_id
    and account_id=p_account_id;

    else
        select nvl(sum(balance), 0)
        into v_balance
        from accounts
        where customer_id= p_customer_id;

    end if;
    return v_balance;
end;
```

SELECT getAccountBalance_udf(1, 1001) FROM dual;

2. Çalışan ekleyen bir prosedür yaz.

- Parametreler: first_name, last_name, email, phone_number, hire_date, job_id, salary, manager_id, department_id.
- Yeni bir çalışanı employees tablosuna eklemeli.

```
create or replace procedure insert_employee_sp(
    p_first_name in varchar2,
    p_last_name in varchar2,
    p_email in varchar2,
    p_phone_number in varchar2,
    p_hire_date in date,
    p_job_id in varchar2,
    p_salary in number,
    p_manager_id in number,
    p_department_id in number
) as
begin
    insert into employees(first_name, last_name, email, phone_number, hire_date, job_id,
          salary, manager_id, department_id)
    values(p_first_name, p_last_name, p_email, p_phone_number, p_hire_date, p_job_id,
          p_salary,  p_manager_id, p_department_id);

    commit;
expection
    when others then
       rollback;

end insert_employee_sp;


BEGIN
    insert_employee_sp(
       'Ali',
       'Yılmaz',
       'ali@example.com',
       '5551234567',
       SYSDATE,
       'IT_PROG',
       5000,
       101,
       60
    );
END;
/
```

3. Çalışanın maaşına zam yapan bir prosedür yaz.

- Parametreler: employee_id, percent_increase (örneğin %10 zam).
- Çalışanın mevcut maaşını güncelleyerek belirlenen yüzdelik oranda artır.

```
create or replace procedure update_employee_maas_sp(
    p_employee_id in number,
    p_percent_increase in number

) as
begin
    update employees
    set salary= salary + (salary * (p_percent_increase / 100))
    where employee_id= p_employee_id;
end update_employee_maas_sp;

commit;

begin
    update_employee_maas_sp(200,10);
end;
```

4. Bir yöneticinin kaç çalışanı olduğunu döndüren bir prosedür yaz.

- Parametre olarak manager_id almalı.
- employees tablosunda bu yöneticinin kaç çalışanı olduğunu bulmalı.
- Eğer çalışan yoksa "Bu yöneticinin altında çalışan yok." şeklinde bir mesaj döndürmeli.

```
create or replace procedure count_manager_employees_sp(
    p_manager_id in number

) as
    v_employee_count number;

begin
    select count(*)
    into v_employee_count
    from employees
    where manager_id= p_manager_id;

end count_manager_employees_sp;

BEGIN
    count_manager_employees_sp(101);
END;
/
```

Finansal CRM Projemden Sp Örnekleri

```
create or replace PROCEDURE INSERT_CUSTOMERSP(
    p_customerid IN NUMBER,
    p_firstName  IN VARCHAR2,
    p_lastName   IN VARCHAR2,
    p_type       IN VARCHAR2
)
AS
BEGIN
    INSERT INTO customers(customer_id, first_name, last_name, type)
    VALUES (p_customerid, p_firstName, p_lastName, p_type);

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END INSERT_CUSTOMERSP;
```

```
create or replace PROCEDURE DELETE_CUSTOMERSP(
    p_customerid in number
)as
begin
delete from customers
where customer_id= p_customerid;
commit;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

```
create or replace PROCEDURE SELECT_CUSTOMER_GETBYIDSP(
    p_customerid in number,
    p_firstname out varchar2,
    p_lastname out varchar2,
    p_type out varchar2
)as
begin
select first_name, last_name, type
into p_firstname, p_lastname, p_type
from customers
where customer_id = p_customerid;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_firstname := NULL;
        p_lastname := NULL;
        p_type := NULL;
    WHEN OTHERS THEN
        RAISE;
END;
```

```
create or replace PROCEDURE SELECT_ALLCUSTOMERSP(
  p_customers OUT SYS_REFCURSOR
)as
begin
open p_customers for
select customer_id, first_name, last_name, type
from customers;
end;
```

```
create or replace PROCEDURE INSERT_CUSTOMERSP(
    p_customerid IN NUMBER,
    p_firstName  IN VARCHAR2,
    p_lastName   IN VARCHAR2,
    p_type       IN VARCHAR2
)
AS
BEGIN
    INSERT INTO customers(customer_id, first_name, last_name, type)
    VALUES (p_customerid, p_firstName, p_lastName, p_type);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END INSERT_CUSTOMERSP;
```

```
create or replace PROCEDURE SELECT_TRANSACTIONBYIDSP (
    p_transactionid in number,
    p_customerid out number,
    p_amount out number,
    p_type out varchar2,
    p_transactionDate out date
) as
begin
select customer_id, amount, type, transaction_date
into p_customerid, p_amount, p_type, p_transactionDate
from transactions
where transaction_id = p_transactionid;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        p_customerid := NULL;
        p_amount := NULL;
        p_type := NULL;
        p_transactionDate := NULL;
    WHEN OTHERS THEN
        RAISE;
END;
```

```
create or replace PROCEDURE DELETE_CUSTOMERSP(
    p_customerid in number
)as
begin
delete from customers
where customer_id= p_customerid;
commit;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```

```
create or replace PROCEDURE INSERT_TRANSACTIONSP (
    p_customerid in number,
    p_amount in number,
    p_type in varchar2,
    p_transactionDate in date
) as
begin
insert into transactions(customer_id, amount, type, transaction_date)
values (p_customerid, p_amount, p_type, p_transactionDate);

commit;

exception
    when others then
    rollback;
    raise;
end insert_transactionsp;
```

```
create or replace PROCEDURE UPDATE_TRANSACTION_STATUS_SP (
    p_transaction_id IN NUMBER,
    p_type           IN VARCHAR2,
    p_amount         IN NUMBER
)
AS
BEGIN
    UPDATE transactions
    SET
        type   = p_type,
        amount = p_amount,
        status = 1
    WHERE transaction_id = p_transaction_id;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END;
```
