```
CREATE OR REPLACE PROCEDURE prc_return_rental (p_rent_no IN NUMBER, p_hours_flown IN NUMBER, p_output OUT VARCHAR2) AS
    p_rental_found NUMBER;
    p_drone_id NUMBER;
BEGIN
    SELECT count(*) into p_rental_found FROM rental WHERE rent_no = p_rent_no;
    IF (p_rental_found = 0) THEN
        p_output := 'Invalid rental number, return process cancelled';
    ELSE
        IF(p_hours_flown <=0) THEN
            p_output := 'Invalid number of hours flown by the drone, the value must be a positive value';
        ELSE
            UPDATE rental SET rent_in = sysdate WHERE rent_no = p_rent_no;
            SELECT drone_id into p_drone_id FROM rental WHERE rent_no = p_rent_no;
            UPDATE DRONE SET drone_flight_time = drone_flight_time + p_hours_flown WHERE drone_id = p_drone_id;
            p_output := 'Rental ' || p_rent_no || ' has been returned, and drone '
            ||p_drone_id||' total flight time has been updated';
        END IF;
    END IF;
END;
/
```

```
--initial data
insert into drone_type values ('DIN2','DJI Inspire 2',5);
insert into drone values (100,to_date('21-02-2021','dd-mm-yyyy'),4200,0,35,0,'DIN2');
insert into rental values (101,200,sysdate,sysdate+7,null, 100);

--before value
select * from rental;

--execute the procedure
DECLARE
    output VARCHAR2(200);
BEGIN
    --call the procedure - error
    prc_return_rental(102,100,output);
    dbms_output.put_line(output);
END;
/
```

```
DECLARE
    output VARCHAR2(200);
BEGIN
    --call the procedure - error
    prc_return_rental(101,-6,output);
    dbms_output.put_line(output);
END;
/
DECLARE
    output VARCHAR2(200);
BEGIN
    --call the procedure - error
    prc_return_rental(101,40,output);
    dbms_output.put_line(output);
END;
/
--after value
select * from rental;
--closes transaction
rollback;
```