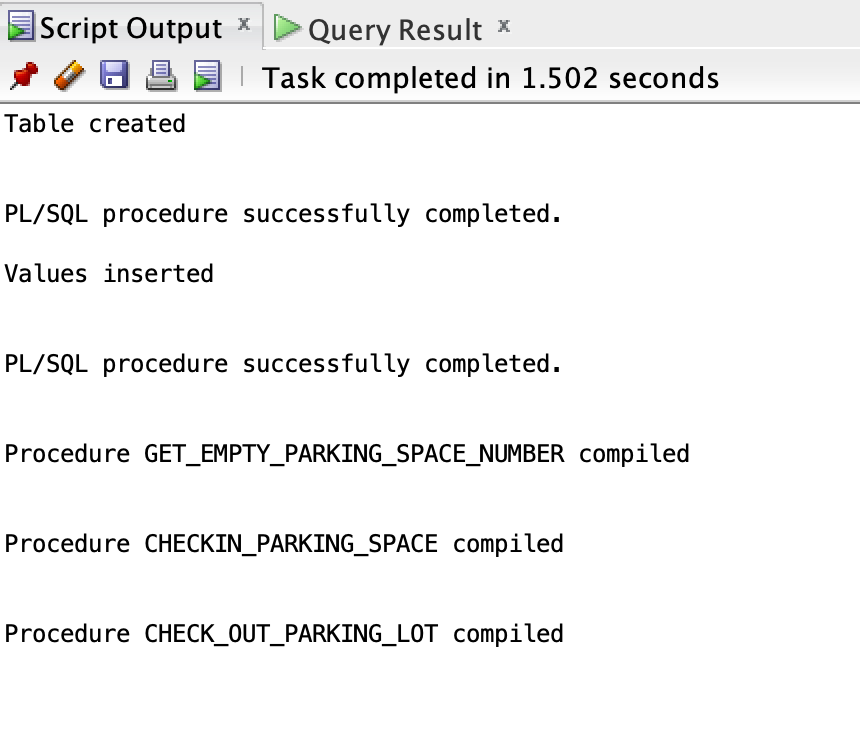
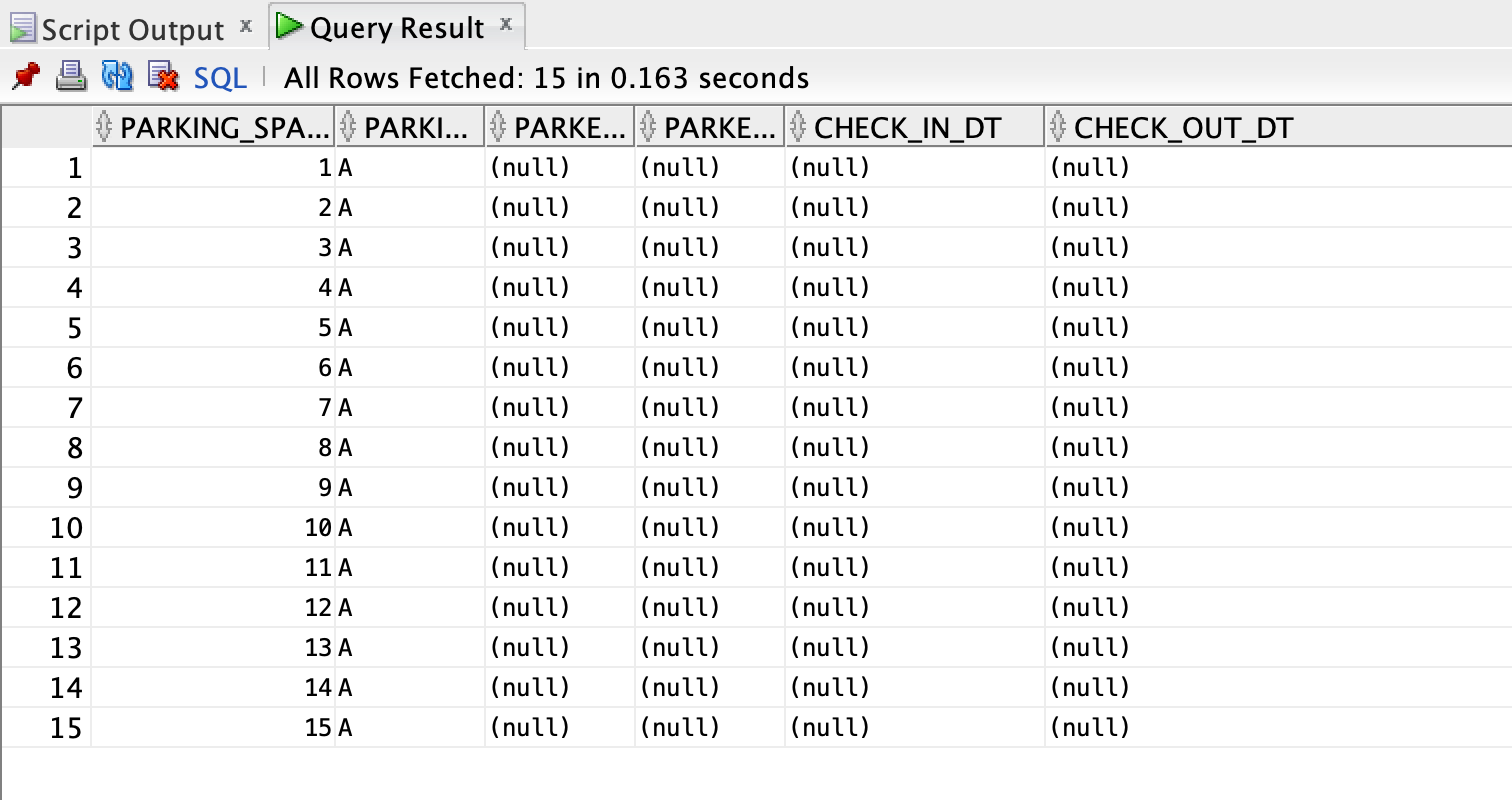
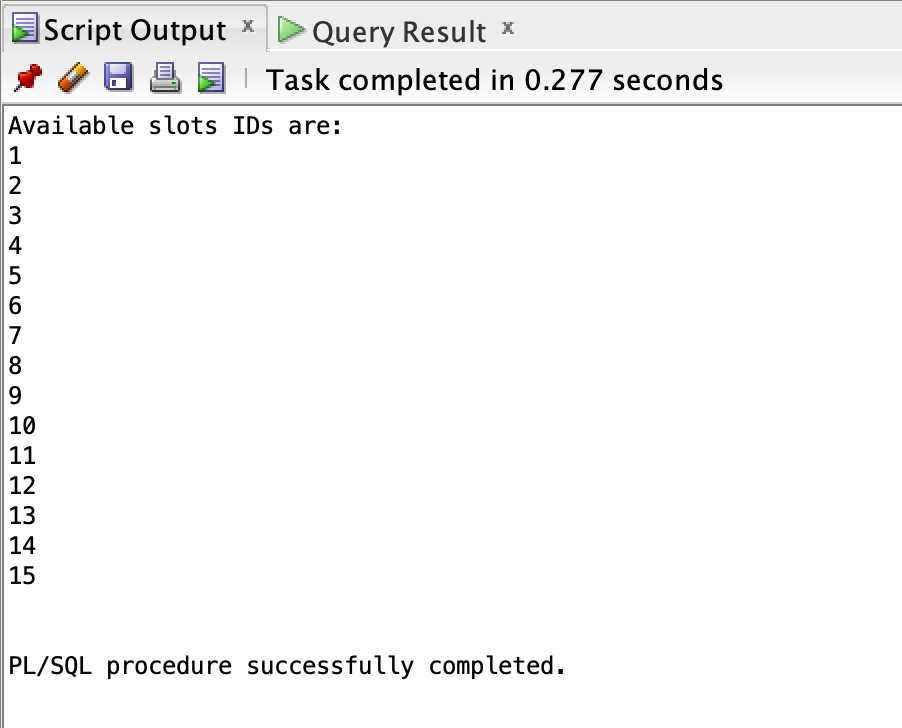
**– Tables and Stored Procedures created**





-- check available spots

exec get\_empty\_parking\_space\_number();



**– TEST CASES:**

---- 1) Insert given data

--A) Checkin\_Parking\_Space: RED, UBX987, 1-JAN-2022, 14

declare

x varchar(20);

begin

x := 14;

Checkin\_Parking\_Space('RED', 'UBX987', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--B) Checkin\_Parking\_Space: WHITE, ZZX954, 1-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('WHITE', 'ZZX954', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--C) Checkin\_Parking\_Space: YELLOW, UEX982, 1-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('YELLOW', 'UEX982', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--D) Checkin\_Parking\_Space: BLUE, AAX983, 2-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('BLUE', 'AAX983', '2-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--E) Checkin\_Parking\_Space: RED, BBX987, 3-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('RED', 'BBX987', '3-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

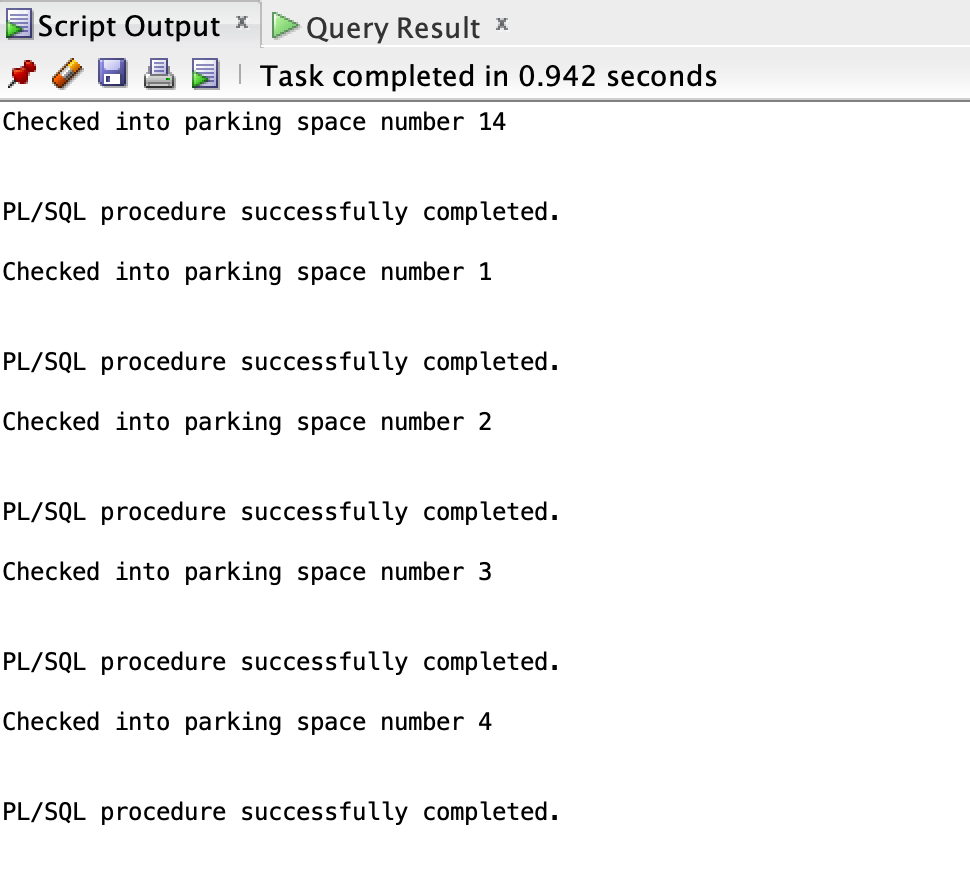
else

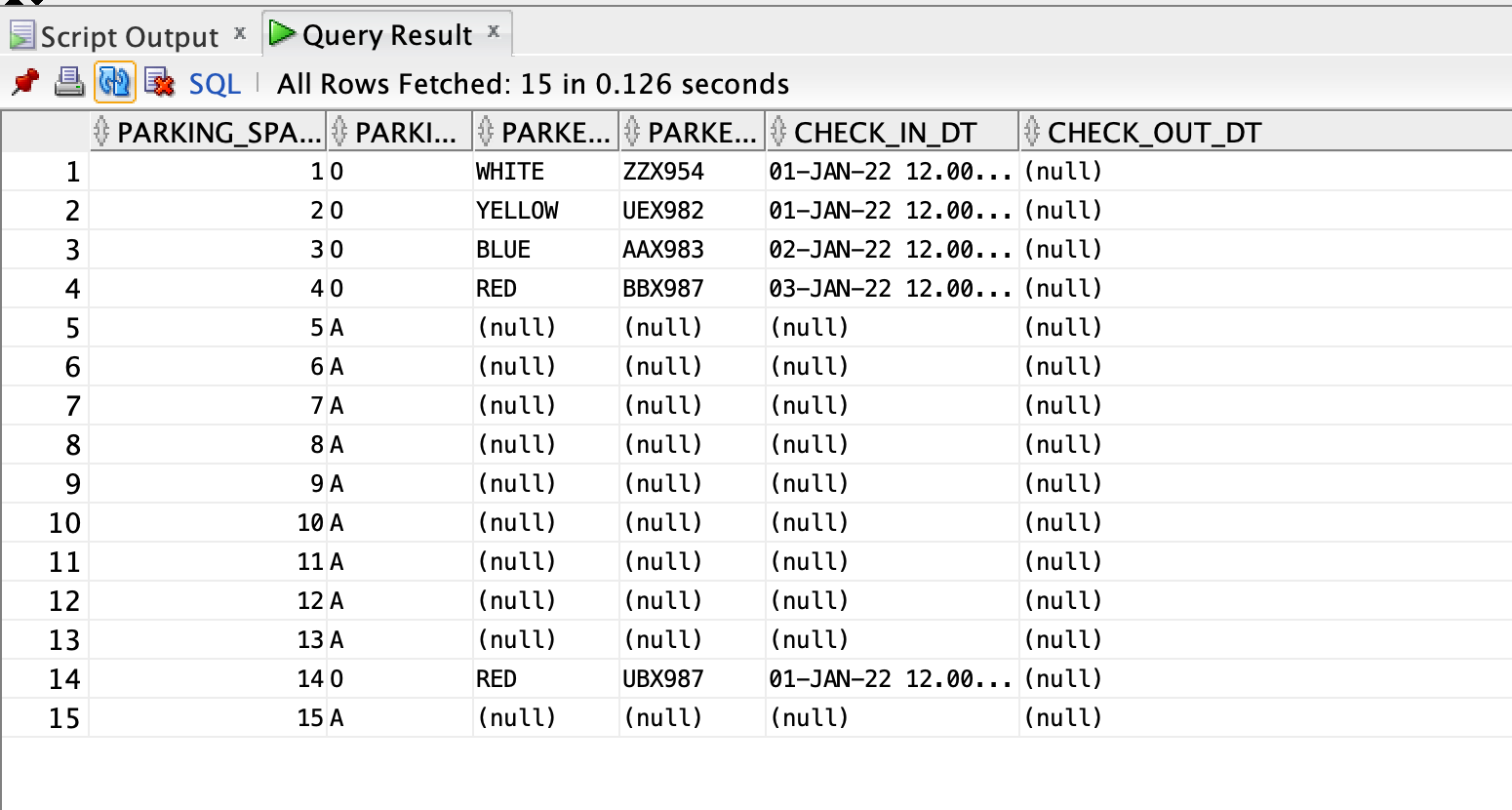
dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/





--F) Checkin\_Parking\_Space: RED, CCX585, 4-JAN-2022, 14 (Should throw error as already occupied)

declare

x varchar(20);

begin

x := 14;

Checkin\_Parking\_Space('RED', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--G) Checkin\_Parking\_Space: WHITE, CCX585, 4-JAN-2022, 13

declare

x varchar(20);

begin

x := 13;

Checkin\_Parking\_Space('WHITE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

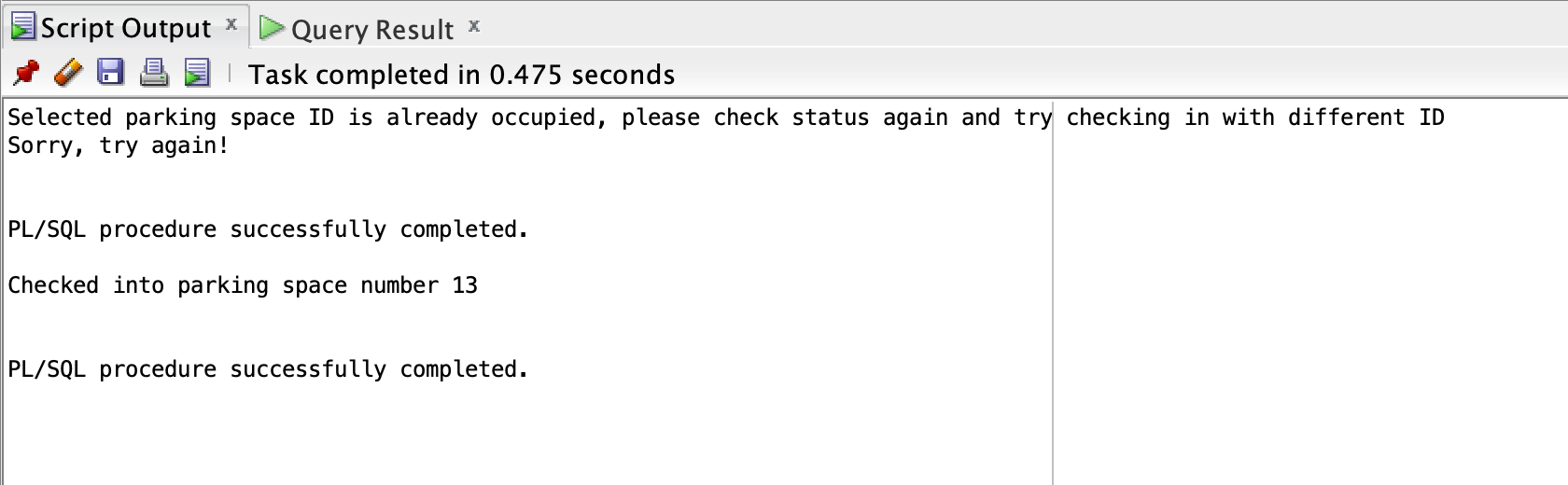
else

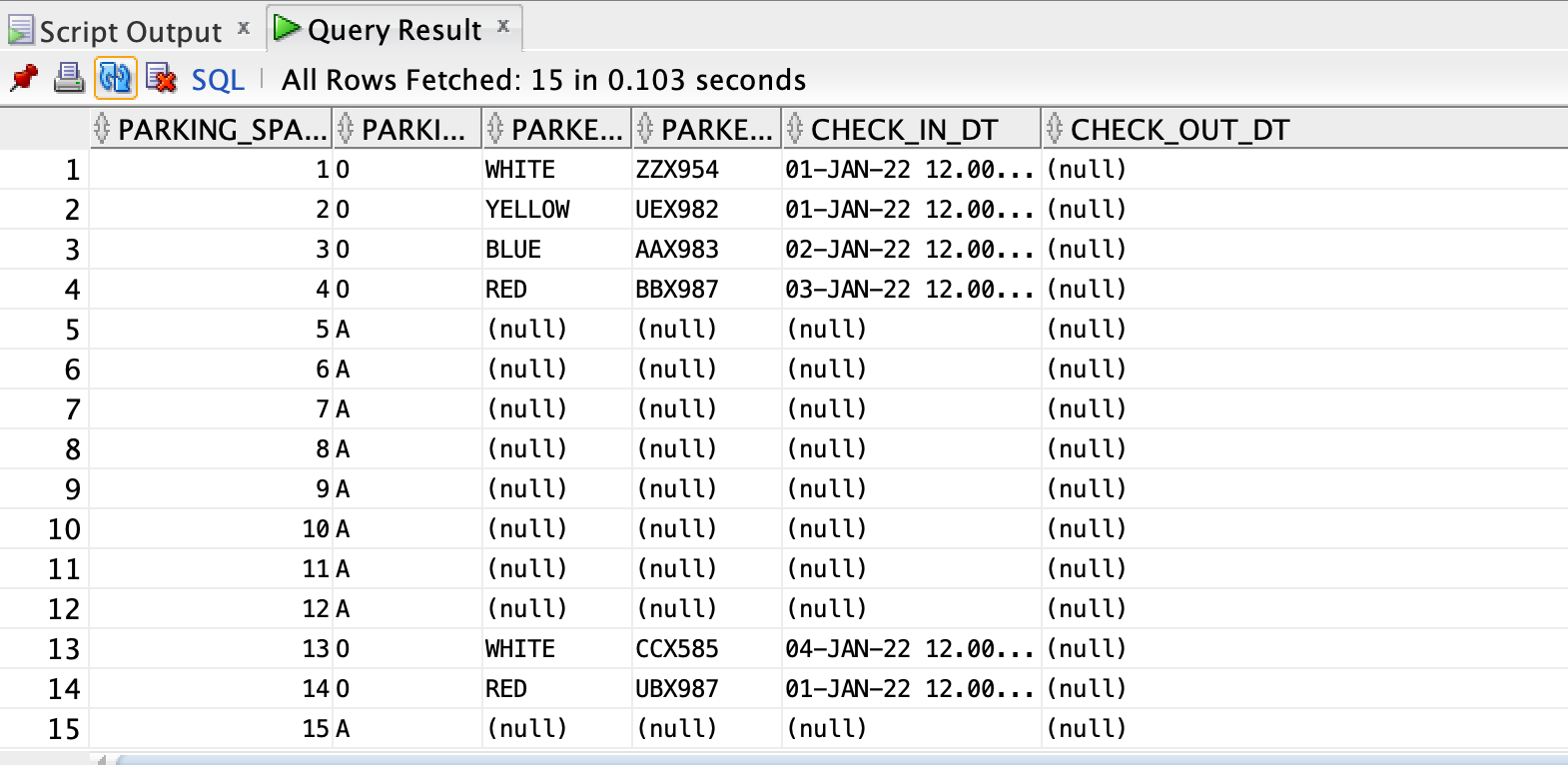
dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/





--H) Checkin\_Parking\_Space: YELLOW, CCX585, 4-JAN-2022, 15

declare

x varchar(20);

begin

x := 15;

Checkin\_Parking\_Space('YELLOW', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--I) Checkin\_Parking\_Space: BLUE, CCX585, 4-JAN-2022, 11

declare

x varchar(20);

begin

x := 11;

Checkin\_Parking\_Space('BLUE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--J) Checkin\_Parking\_Space: WHITE, CCX585, 4-JAN-2022, 12

declare

x varchar(20);

begin

x := 12;

Checkin\_Parking\_Space('WHITE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

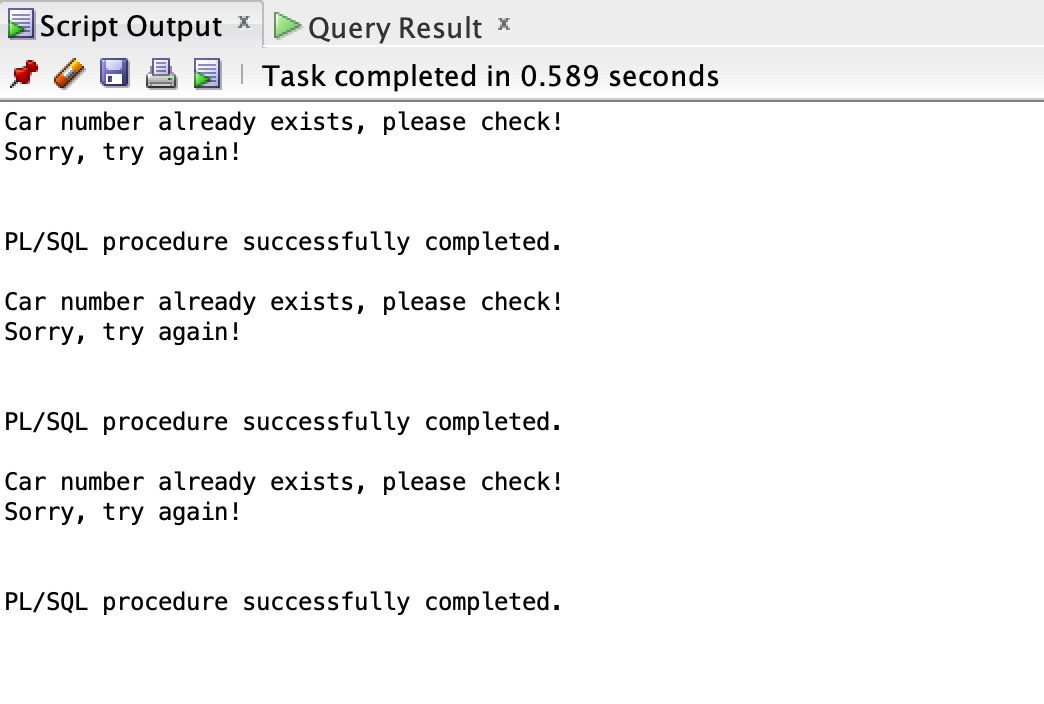
else

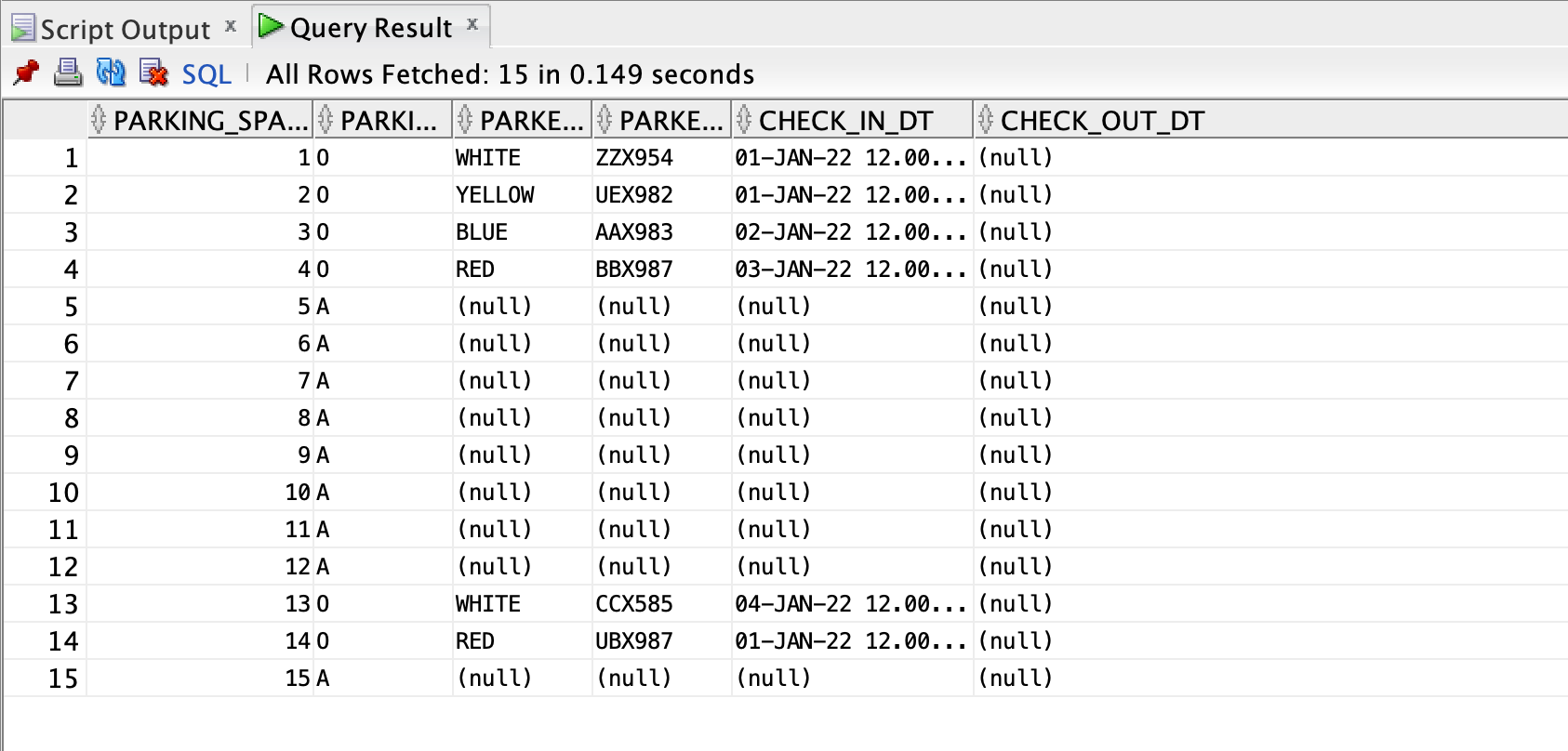
dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

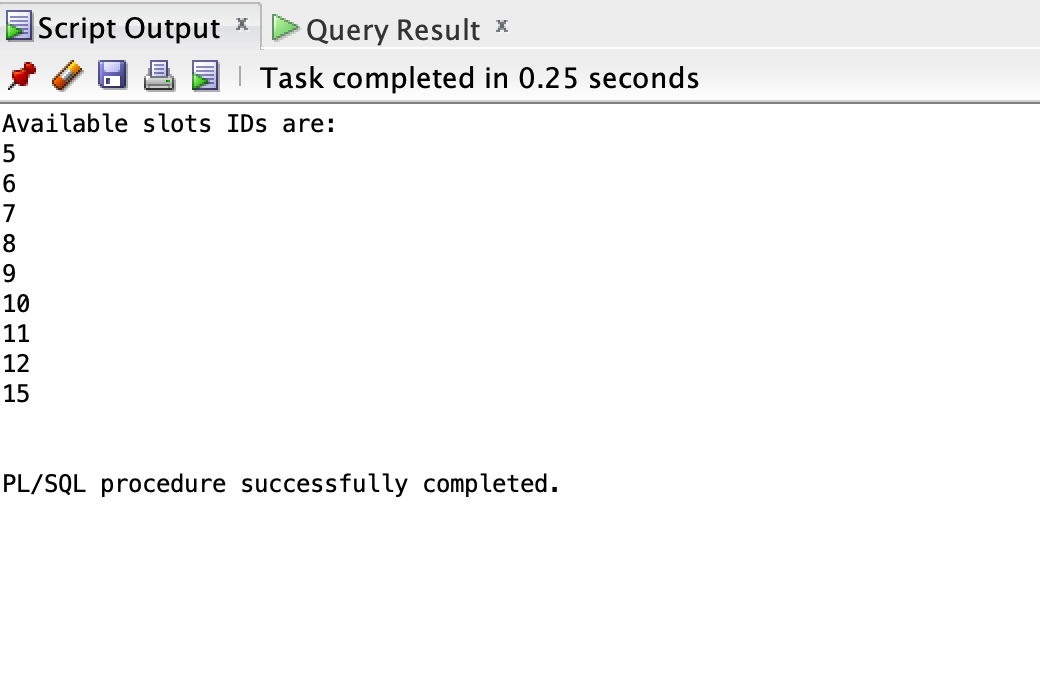
/





-- check available spots

exec get\_empty\_parking\_space\_number();



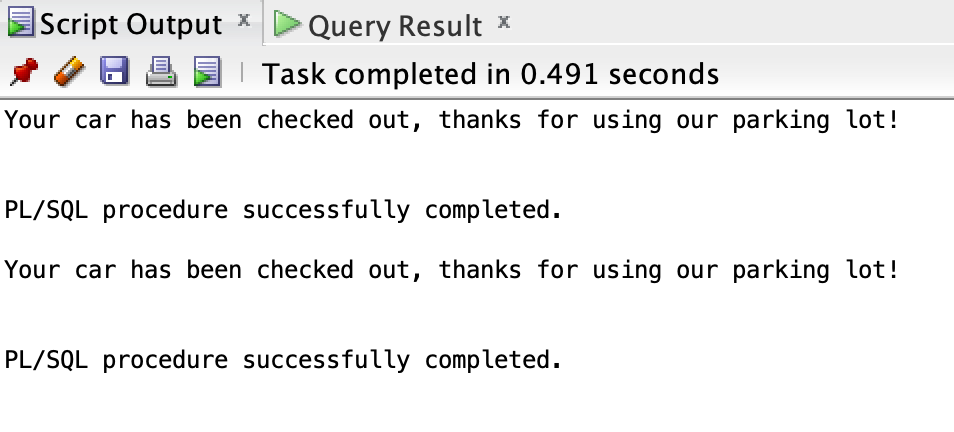
**-- Test cases for check out**

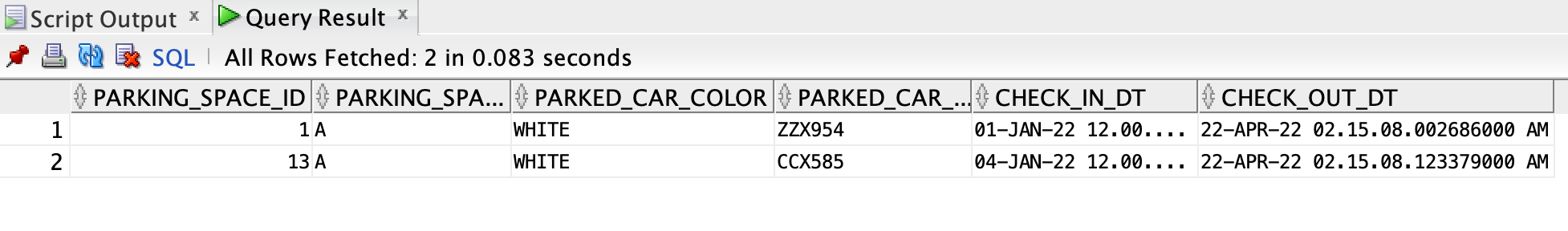
select \* from nyc\_parking\_lot where parked\_car\_color = 'WHITE';

exec Check\_Out\_Parking\_Lot('ZZX954', '');

exec Check\_Out\_Parking\_Lot('CCX585', '');

select \* from nyc\_parking\_lot where parked\_car\_color = 'WHITE';





**-- Entire Code**

set serveroutput on;

--drop table NYC\_Parking\_Lot;

-- Create table if exists

begin

execute immediate (

'create table NYC\_Parking\_Lot(

parking\_space\_id number primary key,

parking\_space\_status varchar(20) default ''A'',

parked\_car\_color varchar(50),

parked\_car\_reg\_number varchar(50) constraint nyc\_car\_uk unique,

check\_in\_dt timestamp,

check\_out\_dt timestamp)');

dbms\_output.put\_line('Table created');

exception

when others then

if sqlcode = -955 then

dbms\_output.put\_line('Table already exists');

else

raise;

end if;

end;

/

-- Insert values for parking space ID

declare

v\_cnt number;

begin

select count(\*) into v\_cnt from NYC\_Parking\_Lot;

if v\_cnt = 0 then

for i in 1..15 loop

insert into NYC\_Parking\_Lot(parking\_space\_id) values(i);

end loop;

commit;

dbms\_output.put\_line('Values inserted');

else

dbms\_output.put\_line('Cannot insert values, clear table and insert again. (Count Exceeded 15');

end if;

end;

/

-- create store procedure to display empty parking lots

create or replace procedure get\_empty\_parking\_space\_number

is

v\_checkCount number;

v\_parkingId nyc\_parking\_lot.parking\_space\_id%TYPE;

v\_parking\_space\_status nyc\_parking\_lot.parking\_space\_status%TYPE;

cursor c\_checkLot is select parking\_space\_id, parking\_space\_status from NYC\_Parking\_Lot;

begin

v\_checkCount := 0;

dbms\_output.put\_line('Available slots IDs are: ');

open c\_checkLot;

loop

fetch c\_checkLot into v\_parkingId, v\_parking\_space\_status;

exit when c\_checkLot%notfound;

if v\_parking\_space\_status = 'A' then

dbms\_output.put\_line(v\_parkingId);

v\_checkCount := 1;

end if;

end loop;

close c\_checkLot;

if v\_checkCount = 0 then

dbms\_output.put\_line('Sorry, no parking slots available for you!');

end if;

end;

/

--exec get\_empty\_parking\_space\_number();

-- create stored procedure to check-in to parking lot

create or replace procedure Checkin\_Parking\_Space(in\_parked\_car\_color in nyc\_parking\_lot.parked\_car\_color%TYPE,

in\_parked\_car\_reg\_number in nyc\_parking\_lot.parked\_car\_reg\_number%TYPE,

in\_check\_in\_dt in char,

in\_parking\_space\_id in out varchar)

is

v\_lotIDError number;

v\_orig\_parking\_status nyc\_parking\_lot.parking\_space\_status%TYPE;

v\_parkingID number;

v\_carExists number;

v\_check\_in\_dt char(50);

e\_carExists exception;

e\_reg\_number exception;

e\_orig\_parking\_status exception;

e\_lotIDError exception;

begin

if length(in\_parked\_car\_reg\_number) is null then raise e\_reg\_number; end if;

select count(\*) into v\_carExists from nyc\_parking\_lot where parked\_car\_reg\_number = in\_parked\_car\_reg\_number;

if v\_carExists > 0 then

raise e\_carExists;

end if;

if length(to\_char(in\_parking\_space\_id)) is null then

select min(parking\_space\_id) into v\_parkingID from nyc\_parking\_lot where parking\_space\_status = 'A';

else

v\_parkingID := in\_parking\_space\_id;

end if;

select count(\*) into v\_lotIDError from nyc\_parking\_lot where parking\_space\_id = v\_parkingID;

if v\_lotIDError = 0 then

raise e\_lotIDError;

end if;

select parking\_space\_status into v\_orig\_parking\_status from nyc\_parking\_lot where parking\_space\_id = v\_parkingID;

if v\_orig\_parking\_status != 'A' then

raise e\_orig\_parking\_status;

end if;

if length(in\_check\_in\_dt) is null then

v\_check\_in\_dt := to\_char(systimestamp, 'DD-Mon-YYYY HH12:MI:SS');

else

v\_check\_in\_dt := in\_check\_in\_dt;

end if;

update nyc\_parking\_lot set parked\_car\_color = upper(in\_parked\_car\_color),

parked\_car\_reg\_number = upper(in\_parked\_car\_reg\_number),

check\_in\_dt = to\_timestamp(v\_check\_in\_dt, 'DD-Mon-YYYY HH24:MI:SS.FF'),

check\_out\_dt = null,

parking\_space\_status = 'O'

where parking\_space\_id = v\_parkingID;

commit;

in\_parking\_space\_id := v\_parkingID;

exception

when e\_carExists then dbms\_output.put\_line('Car number already exists, please check!');

in\_parking\_space\_id := 200;

when e\_reg\_number then dbms\_output.put\_line('Please do not leave car registration number blank!');

in\_parking\_space\_id := 200;

when no\_data\_found then dbms\_output.put\_line('Please do not have null data input');

in\_parking\_space\_id := 200;

when e\_lotIDError then dbms\_output.put\_line('Either invalid ID entered, or no spots available for you at the moment');

in\_parking\_space\_id := 200;

when e\_orig\_parking\_status then dbms\_output.put\_line('Selected parking space ID is already occupied, please check status again and try checking in with different ID');

in\_parking\_space\_id := 200;

when others then raise; in\_parking\_space\_id := 200;

rollback;

end;

/

-- create check-out stored procedure

create or replace procedure Check\_Out\_Parking\_Lot (in\_parked\_car\_reg\_number nyc\_parking\_lot.parked\_car\_reg\_number%TYPE, in\_check\_out\_dt char)

is

v\_nosuchcar number;

v\_check\_out\_dt char(50);

v\_carAlreadychecked number;

e\_carAlreadychecked exception;

e\_carnumbernull exception;

e\_nosuchcar exception;

begin

if length(to\_char(in\_parked\_car\_reg\_number)) is null then

raise e\_carnumbernull;

end if;

select count(\*) into v\_nosuchcar from nyc\_parking\_lot where parked\_car\_reg\_number = in\_parked\_car\_reg\_number;

if v\_nosuchcar = 0 then

raise e\_nosuchcar;

end if;

select count(\*) into v\_carAlreadychecked from nyc\_parking\_lot where parked\_car\_reg\_number = in\_parked\_car\_reg\_number and parking\_space\_status = 'A';

if v\_carAlreadychecked != 0 then

raise e\_carAlreadychecked;

end if;

if length(in\_check\_out\_dt) is null then

v\_check\_out\_dt := to\_char(systimestamp, 'DD-Mon-YYYY HH24:MI:SS.FF');

else

v\_check\_out\_dt := in\_check\_out\_dt;

end if;

update nyc\_parking\_lot set check\_out\_dt = to\_timestamp(v\_check\_out\_dt, 'DD-Mon-YYYY HH24:MI:SS.FF'),

parking\_space\_status = 'A'

where parked\_car\_reg\_number = in\_parked\_car\_reg\_number;

commit;

dbms\_output.put\_line('Your car has been checked out, thanks for using our parking lot!');

exception

when e\_carAlreadychecked then dbms\_output.put\_line('Sorry you have already checked your car out');

when e\_carnumbernull then dbms\_output.put\_line('Please do not leave car reg number null, it is a req field!');

when e\_nosuchcar then dbms\_output.put\_line('Sorry, no such car exists in our lot!');

when others then dbms\_output.put\_line('Contact sys admin!');

rollback;

end;

/

-- check available spots

exec get\_empty\_parking\_space\_number();

--test cases

---- 1) Insert given data

--A) Checkin\_Parking\_Space: RED, UBX987, 1-JAN-2022, 14

declare

x varchar(20);

begin

x := 14;

Checkin\_Parking\_Space('RED', 'UBX987', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--B) Checkin\_Parking\_Space: WHITE, ZZX954, 1-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('WHITE', 'ZZX954', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--C) Checkin\_Parking\_Space: YELLOW, UEX982, 1-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('YELLOW', 'UEX982', '1-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--D) Checkin\_Parking\_Space: BLUE, AAX983, 2-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('BLUE', 'AAX983', '2-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--E) Checkin\_Parking\_Space: RED, BBX987, 3-JAN-2022, NULL

declare

x varchar(20);

begin

x := null;

Checkin\_Parking\_Space('RED', 'BBX987', '3-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--F) Checkin\_Parking\_Space: RED, CCX585, 4-JAN-2022, 14 (Should throw error as already occupied)

declare

x varchar(20);

begin

x := 14;

Checkin\_Parking\_Space('RED', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--G) Checkin\_Parking\_Space: WHITE, CCX585, 4-JAN-2022, 13

declare

x varchar(20);

begin

x := 13;

Checkin\_Parking\_Space('WHITE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--H) Checkin\_Parking\_Space: YELLOW, CCX585, 4-JAN-2022, 15

declare

x varchar(20);

begin

x := 15;

Checkin\_Parking\_Space('YELLOW', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--I) Checkin\_Parking\_Space: BLUE, CCX585, 4-JAN-2022, 11

declare

x varchar(20);

begin

x := 11;

Checkin\_Parking\_Space('BLUE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

--J) Checkin\_Parking\_Space: WHITE, CCX585, 4-JAN-2022, 12

declare

x varchar(20);

begin

x := 12;

Checkin\_Parking\_Space('WHITE', 'CCX585', '4-JAN-2022', x);

if x = 200 then

dbms\_output.put\_line('Sorry, try again!');

else

dbms\_output.put\_line('Checked into parking space number '|| x);

end if;

end;

/

-- check available spots

exec get\_empty\_parking\_space\_number();

-- Test cases for check out

select \* from nyc\_parking\_lot where parked\_car\_color = 'WHITE';

exec Check\_Out\_Parking\_Lot('ZZX954', '');

exec Check\_Out\_Parking\_Lot('CCX585', '');

select \* from nyc\_parking\_lot where parked\_car\_color = 'WHITE';