After table creation: select \* from nyc\_parking\_lot;

Table

Description automatically generated

exec Get\_Empty\_Parking\_Space\_Number;

Graphical user interface, text, application, email

Description automatically generated

CHECK IN

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

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

1. Checkin\_Parking\_Space: BLUE, CCX585, 4-JAN-2022, 11

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

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

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Case: Invalid parking space number

Graphical user interface, text, application, email

Description automatically generated

CHECK OUT

exec Get\_Empty\_Parking\_Space\_Number;

Graphical user interface, text, application, email

Description automatically generated

Case: Check out white cars

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

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

exec Check\_Out\_Parking\_Lot('CCX585', '5-JAN-2022');

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Case: Checking out car again after its already checked out:

exec Check\_Out\_Parking\_Lot('CCX585', '5-JAN-2022');

Graphical user interface, text, application, email

Description automatically generated

Case: Car registration number field null: exec Check\_Out\_Parking\_Lot('', sysdate);

Graphical user interface, text, application

Description automatically generated

Case: Invalid car registration number: exec Check\_Out\_Parking\_Lot('CCX985', '5-JAN-2022');

Graphical user interface, text, application, email

Description automatically generated

Case:Lower case car registration number and checked out on sysdate

exec Check\_Out\_Parking\_Lot('uex982', sysdate);

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

exec Get\_Empty\_Parking\_Space\_Number;

Graphical user interface, text, application, email

Description automatically generated

Code:

set serveroutput on;

--drop table NYC\_Parking\_lot;

--Parking Space Table creation

declare

v\_cnt number;

begin

select count(\*) into v\_cnt from user\_tables where table\_name = upper('NYC\_Parking\_lot');

if v\_cnt = 0 then

execute immediate('

create table NYC\_Parking\_lot (

parking\_space\_id number primary key,

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

constraint check\_status check (parking\_space\_status=''A'' OR parking\_space\_status=''O''),

parked\_car\_color varchar(50),

parked\_car\_reg\_number varchar(50) constraint reg\_unique unique,

checkin\_dt date,

checkout\_dt date)');

dbms\_output.put\_line('NYC\_Parking\_lot table created');

else

execute immediate('Delete from NYC\_Parking\_lot');

commit;

dbms\_output.put\_line('NYC\_Parking\_lot table already exists; deleted existing data and inserting again');

end if;

end;

/

begin

for j in 1..15

loop

insert into nyc\_parking\_lot (parking\_space\_id) values (j);

end loop;

commit;

end;

/

--Get empty parking spaces procedure

create or replace procedure Get\_Empty\_Parking\_Space\_Number is

begin

dbms\_output.put\_line('Available parking spaces are: ');

for i in (select parking\_space\_id, parking\_space\_status from nyc\_parking\_lot where parking\_space\_status = 'A')

loop

dbms\_output.put\_line(i.parking\_space\_id);

end loop;

end;

/

exec Get\_Empty\_Parking\_Space\_Number;

-- Check In Procedure

create or replace procedure Checkin\_Parking\_Space (car\_color varchar, car\_reg\_number varchar, checkin\_date varchar, parking\_space\_num in out number) is

v\_status varchar(50);

v\_parking\_space\_num number;

v\_checkin\_date date;

e\_notavailable exception;

e\_invalidparkingspace exception;

e\_carregnumberReqd exception;

begin

if parking\_space\_num not in (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) then

raise e\_invalidparkingspace;

end if;

if length(car\_reg\_number) is null then

raise e\_carregnumberReqd;

end if;

if length(checkin\_date) is null then

v\_checkin\_date := sysdate ;

else

v\_checkin\_date := to\_date(checkin\_date, 'DD-MON-YYYY');

end if;

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

if v\_status != 'A' then

raise e\_notavailable;

end if;

update nyc\_parking\_lot set parking\_space\_status = 'O', parked\_car\_color=car\_color, parked\_car\_reg\_number=upper(car\_reg\_number),

checkin\_dt=v\_checkin\_date, checkout\_dt = null where parking\_space\_id=parking\_space\_num;

dbms\_output.put\_line('You have been checked into parking space number ' || parking\_space\_num || ' successfully.');

commit;

exception

when e\_invalidparkingspace then dbms\_output.put\_line('Please enter a valid parking space number between 1 and 15.');

when e\_carregnumberReqd then dbms\_output.put\_line('Car registration number is a required field.');

when e\_notavailable then dbms\_output.put\_line('Sorry, parking space number ' || parking\_space\_num || ' is not available.');

when dup\_val\_on\_index then dbms\_output.put\_line('Sorry, car registration number ' || car\_reg\_number || ' has already been checked in.');

when no\_data\_found then

--When parking space input is null, assigns the first available parking space in the table

select parking\_space\_id into v\_parking\_space\_num from (select parking\_space\_id from nyc\_parking\_lot where parking\_space\_status = 'A') where rownum=1;

update nyc\_parking\_lot set parking\_space\_status = 'O', parked\_car\_color=car\_color, parked\_car\_reg\_number=upper(car\_reg\_number),

checkin\_dt=v\_checkin\_date where parking\_space\_id=v\_parking\_space\_num;

dbms\_output.put\_line('Here is your parking space number: ' || v\_parking\_space\_num);

when others then dbms\_output.put\_line('CONTACT ADMIN');

end;

/

--select \* from nyc\_parking\_lot;

--A

declare

id number;

begin

id := 14;

checkin\_parking\_space('RED', 'UBX987', '1-JAN-2022', id);

end;

/

--B

declare

id number;

begin

id := '';

checkin\_parking\_space('WHITE', 'ZZX954', '1-JAN-2022', id);

end;

/

--C

declare

id number;

begin

id := '';

checkin\_parking\_space('YELLOW', 'UEX982', '1-JAN-2022', id);

end;

/

--D

declare

id number;

begin

id := '';

checkin\_parking\_space('BLUE', 'AAX983', '2-JAN-2022', id);

end;

/

--E

declare

id number;

begin

id := '';

checkin\_parking\_space('RED', 'BBX987', '3-JAN-2022', id);

end;

/

--F

declare

id number;

begin

id := 14;

checkin\_parking\_space('RED', 'CCX585', '4-JAN-2022', id);

end;

/

--G

declare

id number;

begin

id := 13;

checkin\_parking\_space('WHITE', 'CCX585', '4-JAN-2022', id);

end;

/

--H

declare

id number;

begin

id := 15;

checkin\_parking\_space('YELLOW', 'CCX585', '4-JAN-2022', id);

end;

/

--I

declare

id number;

begin

id := 11;

checkin\_parking\_space('BLUE', 'CCX585', '4-JAN-2022', id);

end;

/

--J

declare

id number;

begin

id := 12;

checkin\_parking\_space('WHITE', 'CCX585', '4-JAN-2022', id);

end;

/

--Invalid parking space number

declare

id number;

begin

id := 16;

checkin\_parking\_space('RED', 'UBX987', '1-JAN-2022', id);

end;

/

exec Get\_Empty\_Parking\_Space\_Number;

--Check Out procedure

create or replace procedure Check\_Out\_Parking\_Lot (car\_reg\_number varchar, checkout\_date varchar) is

v\_checkout\_date date;

v\_carreg number;

v\_carcheckedout number;

e\_carregnumberReqd exception;

e\_invalidcarreg exception;

e\_carcheckedout exception;

begin

select count(\*) into v\_carreg from nyc\_parking\_lot where parked\_car\_reg\_number = upper(car\_reg\_number);

if length(car\_reg\_number) is null then

raise e\_carregnumberReqd;

end if;

if v\_carreg = 0 then

raise e\_invalidcarreg;

end if;

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

if v\_carcheckedout != 0 then

raise e\_carcheckedout;

end if;

if length(checkout\_date) is null then

v\_checkout\_date := sysdate ;

else

v\_checkout\_date := to\_date(checkout\_date, 'DD-MON-YYYY');

end if;

update nyc\_parking\_lot set parking\_space\_status = 'A', checkout\_dt=v\_checkout\_date where parked\_car\_reg\_number=upper(car\_reg\_number);

dbms\_output.put\_line('You have been checked out of parking space successfully.');

commit;

exception

when e\_carregnumberReqd then dbms\_output.put\_line('Car registration number is a required field.');

when e\_invalidcarreg then dbms\_output.put\_line('Invalid car registration number.');

when e\_carcheckedout then dbms\_output.put\_line('Car has already been checked out.');

when others then dbms\_output.put\_line('CONTACT ADMIN');

rollback;

end;

/

--select \* from nyc\_parking\_lot;

--Checking out white cars

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

exec Check\_Out\_Parking\_Lot('CCX585', '5-JAN-2022');

--Checking out car again after its already checked out

exec Check\_Out\_Parking\_Lot('CCX585', '5-JAN-2022');

--Car registration number field null

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

--Invalid car registration number

exec Check\_Out\_Parking\_Lot('CCX985', '5-JAN-2022');

--Lower case car registration number and checked out on sysdate

exec Check\_Out\_Parking\_Lot('uex982', sysdate);

exec Get\_Empty\_Parking\_Space\_Number;