

```
create or replace PACKAGE body cars4you
AS
/*this function is to register a new customer*/
FUNCTION register_customer(
    i_forename customer.CUST_FORENAME%type,
    i_surname customer.CUST_surname%type,
    i_gender customer.gender%type,
    i_email customer.EMAIL_ADDRESS%type,
    i_phone customer.phone_number%type,
    i_ADDRESS_LINE1 customer.ADDRESS_LINE1%type,
    i_ADDRESS_LINE2 customer.ADDRESS_LINE2%type,
    i_ADDRESS_LINE3 customer.ADDRESS_LINE3%type,
    i_TOWN customer.TOWN%type,
    i_post_code customer.post_code%type,
    i_Country customer.country%type )
    RETURN NUMBER
IS
    o_cust_id NUMBER;
BEGIN
    IF(i_forename IS NULL OR i_surname IS NULL OR i_email IS NULL) THEN
        dbms_output.put_line('forename ,surname and email are mandatory fields');
    ELSE
        SELECT cust_id_seq.nextval INTO o_cust_id FROM dual;
        INSERT
        INTO customer VALUES
            (
                o_cust_id,
                i_forename,
                i_surname,
                i_gender,
                i_email,
                i_phone,
                i_ADDRESS_LINE1,
                i_ADDRESS_LINE2,
                i_ADDRESS_LINE3,
                i_TOWN,
                i_post_code,
                i_Country
            );
        commit;
    END IF;
    IF o_cust_id IS NOT NULL THEN
        dbms_output.put_line('Customer registration is successful and your customer id is: '||
            o_cust_id);
    ELSE
        dbms_output.put_line('Customer registration is unsuccessful');
    END IF;
    RETURN o_cust_id;
EXCEPTION
WHEN OTHERS THEN
    dbms_output.put_line('something wrong,Insertion failed');
    RETURN NULL;
END register_customer;
/*this function is to register a customer by accepting only mandatory fields*/

FUNCTION register_customer
(
```

```

        i_forename customer.CUST_FORENAME%type,
        i_surname customer.CUST_surname%type,
        i_email customer.EMAIL_ADDRESS%type
    )
    RETURN NUMBER
IS
    o_cust_id NUMBER;
BEGIN
    IF(i_forename IS NULL OR i_surname IS NULL OR i_email IS NULL) THEN
        dbms_output.put_line('forename , surname and email are mandatory fields');
    ELSE
        SELECT cust_id_seq.nextval INTO o_cust_id FROM dual;
        INSERT
        INTO customer
        (
            customer_id,
            cust_forename,
            cust_surname,
            email_address
        )
        VALUES
        (
            o_cust_id,
            i_forename,
            i_surname,
            i_email
        );
        commit;
    END IF;
    IF o_cust_id IS NOT NULL THEN
        dbms_output.put_line('Customer registration is successful and your customer id is: '||
            o_cust_id);
    ELSE
        dbms_output.put_line('Customer registration is unsuccessful');
    END IF;
    RETURN o_cust_id;
EXCEPTION
WHEN OTHERS THEN
    dbms_output.put_line('something wrong, Insertion failed');
    RETURN NULL;
END register_customer;

/*code for booking a vehicle*/
FUNCTION book_a_car
(
    cust_id NUMBER,
    i_forename customer.CUST_FORENAME%type,
    i_surname customer.CUST_surname%type,
    i_email customer.EMAIL_ADDRESS%type,
    manufacturer_code VARCHAR2,
    model_code VARCHAR2,
    vehicle_category_code VARCHAR2,
    from_date DATE,
    to_date DATE
)
RETURN BOOLEAN

```

IS

```

is_status                                BOOLEAN;
l_cust_id_count                          NUMBER:=0;
l_cust_id_email_count                    NUMBER;
l_is_available                           BOOLEAN;
l_reg_number                             VARCHAR2(20);
l_booking_id                             NUMBER;
l_cust_id                                NUMBER;
l_cust_id_frm_email number;
l_to_email varchar2(50);
l_email_subject varchar2(100):='Booking Confirmation';
l_msg_body varchar2(300);
is_sent boolean;
    invalid_date_combi_excep              EXCEPTION;
booking_period_term_exception EXCEPTION;
advance_booking_exception                EXCEPTION;
invalid_cust_id                          EXCEPTION;
customer_creation_error                  EXCEPTION;
past_date_exception Exception;

```

BEGIN

```

l_cust_id :=cust_id;
if(from_date<trunc(sysdate))
then
raise past_date_exception;
end if;

IF(from_date>to_date) THEN
    raise invalid_date_combi_excep;
END IF;
IF(to_date-from_date>14) THEN
    raise booking_period_term_exception;
END IF;
IF l_cust_id IS NOT NULL THEN
    SELECT COUNT(*)
    INTO l_cust_id_count
    FROM customer
    WHERE customer_id =l_cust_id;
    IF l_cust_id_count <>1 THEN
        raise invalid_cust_id;
    END IF;
ELSIF i_email IS NOT NULL THEN
    SELECT COUNT(*),max(customer_id)
    INTO l_cust_id_email_count,l_cust_id_frm_email
    FROM customer
    WHERE email_address =i_email ;
END IF;
l_cust_id:=l_cust_id_frm_email;

dbms_output.put_line('Cust id from search is: '||l_cust_id_frm_email);

IF (l_cust_id_count <> 1 AND l_cust_id_email_count<1) THEN
    l_cust_id :=register_customer(i_forename,i_surname,i_email);
    IF l_cust_id IS NULL THEN
        raise customer_creation_error;
    END IF;
END IF;

```

```
select email_address into l_to_email from customer where customer_id=l_cust_id;
```

```
l_is_available      := get_availability( manufacturer_code,model_code,
vehicle_category_code,from_date,to_date,l_reg_number);
```

```
dbms_output.put_line('vehicle availability is : '||case l_is_available when true then
'true' else 'false' end);
```

```
IF l_is_available      =true THEN
```

```
  IF(from_date-sysdate>7) THEN
```

```
    raise advance_booking_exception;
```

```
  END IF;
```

```
  SELECT book_id_seq.nextval INTO l_booking_id FROM dual;
```

```
  INSERT
```

```
  INTO booking VALUES
```

```
  (
    l_booking_id,
    'OPEN',
    l_cust_id,
    l_reg_number,
    from_date,
    to_date,
    'N',
    'N'
  );
```

```
dbms_output.put_line('your booking is successful and your booking id: '||l_booking_id
|| ' Customer id:'||l_cust_id||' and your return date: '||to_date);
```

```
l_msg_body:='Dear Customer,'||chr(13) ||'Your booking is successful and your booking
id: '||l_booking_id|| ' Customer id:'||l_cust_id||' and your return date: '||to_date;
```

```
/* Commenting the email confirmation sending code as there are issues with SMTP mail
server setup and setting the is_sent flag as true even mail didnt send
```

```
is_sent:=
```

```
mail_util.send_mail('no-reply@cars4you.com',l_to_email,l_email_subject,l_msg_body);
```

```
*/
```

```
is_sent:=true;
```

```
if(is_sent =true)
```

```
then
```

```
  dbms_output.put_line('Email sent successfully.....');
```

```
update booking
```

```
set confirm_letter_sent_flag='Y',
```

```
BOOKING_STATUS_CODE='CONFIRMED'
```

```
where booking_id=l_booking_id;
```

```
end if;
```

```
commit;
```

```
RETURN true;
```

```
END IF;
```

```
RETURN is_status;
```

```
EXCEPTION
```

```
when past_date_exception then
```

```
dbms_output.put_line('Booking start date should not be in the past....booking unsuccessful');
```

```
return false;
```

```

WHEN invalid_cust_id THEN
dbms_output.put_line('You entered invalid Customer id, if you forget your customer_id, just
pass your email.....booking unsuccessful');
RETURN false;
WHEN advance_booking_exception THEN
dbms_output.put_line('You should not book a car more than 7 days in advance..booking
unsuccessful');
return false;
WHEN invalid_date_combi_excep THEN
dbms_output.put_line('booking start date should not be greater than the booking end
date..booking unsuccessful');
RETURN false;
WHEN booking_period_term_exception THEN
dbms_output.put_line('You shouldnt be allowed to book more than 14 days in single booking
, you need to do a subsequent bookings in order to book for more than 14 days..booking
unsuccessful');
RETURN false;
WHEN customer_creation_error THEN
dbms_output.put_line('Unable to create new customer.....booking unsuccessful');
RETURN false;
WHEN OTHERS THEN
dbms_output.put_line('exception occured'||SUBSTR(SQLERRM, 1, 200));
dbms_output.put_line('booking unsuccessful');
RETURN false;
END book_a_car;
FUNCTION get_availability
(
    manufacturer_code    VARCHAR2,
    model_code           VARCHAR2,
    vehicle_category_code VARCHAR2,
    start_date           DATE,
    end_date             DATE,
    o_reg_number OUT VARCHAR2
)
RETURN BOOLEAN
IS
    l_reg_number VARCHAR2(50);
BEGIN

select vehicle.reg_number    INTO l_reg_number from vehicle
WHERE vehicle.manufacturer_code =manufacturer_code
AND vehicle.model_code        =model_code
AND vehicle.vehicle_category_code=vehicle_category_code
and not exists
(
select 1 from booking where vehicle.reg_number = booking.reg_number
AND
((start_date BETWEEN booking.date_from AND booking.date_to)
OR (end_date BETWEEN booking.date_from AND booking.date_to)))
and rownum=1;
IF l_reg_number IS NOT NULL THEN
    o_reg_number :=l_reg_number;
    RETURN true;
ELSE
    RETURN false;
END IF;
EXCEPTION

```

```

WHEN no_data_found THEN
    dbms_output.put_line('No vehicles available for booking with this time frames');
    RETURN false;
END get_availability;

FUNCTION get_customer_id(
    forename VARCHAR2,
    surname   VARCHAR2,
    email     VARCHAR2 )
    RETURN NUMBER
IS
    l_cust_id number;

BEGIN
    SELECT customer_id
    INTO l_cust_id
    FROM customer
    WHERE email_address =email and cust_forename=forename and cust_surname=surname ;
    if(l_cust_id is not null) then
        return l_cust_id;
    end if;
    exception
    when no_data_found then
        return null;

END get_customer_id;

PROCEDURE print_booking_report
IS
    TYPE report_record IS RECORD (
        booking_id number,
        booking_status_code varchar2(100),
        customer_id number,
        booking_dates varchar2(100),
        vehicle_category_code varchar2(100),
        manufacturer_code varchar2(100),
        manufacturer_name varchar2(100) ,
        model_code varchar2(100),
        model_name varchar2(100),
        reg_number varchar2(100)
    );
    l_report_record report_record;
    cursor booking_cur is
SELECT booking_id,
        booking_status_code,
        customer_id,
        date_from
        ||' to '
        ||date_to as booking_dates,
        vehicle.vehicle_category_code,
        manufacturer.manufacturer_code,
        manufacturer.manufacturer_name ,
        model.model_code,
        model.model_name,
        vehicle.reg_number
FROM booking_hist,
    vehicle,

```

```

    manufacturer,
    model
WHERE booking_hist.reg_number =vehicle.reg_number
AND vehicle.manufacturer_code = manufacturer.manufacturer_code
AND model.model_code =vehicle.model_code
AND booking_hist.booking_date =TRUNC(sysdate) ;
BEGIN
dbms_output.put_line('Printing todays booking details.....');

dbms_output.put_line('Booking-Id   Booking-Status   Customer-Id   Booking-Dates
vehicle-type   manufacturer-Name   vehicle-Model   Registration#');
dbms_output.put_line(
'-----'
'-----');

OPEN booking_cur;
LOOP
    FETCH booking_cur INTO l_report_record;

    EXIT WHEN booking_cur%notfound;
    dbms_output.put_line(rpad(to_char(l_report_record.booking_id),11,'#')||' ' ||rpad(
l_report_record.booking_status_code,length('Booking-Status'),'#')||' '||
rpad(to_char(l_report_record.customer_id),length('customer_id'),'#')||' '||
rpad(l_report_record.booking_dates,25,'#')||' '||
rpad(l_report_record.vehicle_category_code,length('vehicle-type'),'#')||' '||
rpad(l_report_record.manufacturer_name,length('manufacturer-Name'),'#')||' '||
rpad(l_report_record.model_name,length('vehicle-Model'),'#')||' '||
rpad(l_report_record.reg_number,length('Registration#'),'#'));

END LOOP;
CLOSE booking_cur;

END print_booking_report;

PROCEDURE add_new_vehicle_to_fleet(
    REG_NUMBER          VARCHAR2,
    MANUFACTURER_CODE   VARCHAR2,
    MODEL_CODE           VARCHAR2,
    VEHICLE_CATEGORY_CODE VARCHAR2,
    CURRENT_MILEAGE      NUMBER,
    DAILY_HIRE_RATE      NUMBER,
    DATE_MOT_DUE         DATE )
IS
BEGIN
    INSERT
    INTO VEHICLE VALUES
    (
        REG_NUMBER ,
        MANUFACTURER_CODE,
        MODEL_CODE,
        VEHICLE_CATEGORY_CODE ,
        CURRENT_MILEAGE,
        DAILY_HIRE_RATE,
        DATE_MOT_DUE
    );
    commit;

```

```
DBMS_OUTPUT.PUT_LINE('Vehicle added successfully...');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT_LINE('EXCEPTION WHILE ADDING THE VEHICLE TO THE FLEET DUE TO :'+SUBSTR(
        SQLERRM,1,200));
```

```
END;
```

```
PROCEDURE add_new_booking_status
```

```
(
    BOOKING_STATUS_CODE          VARCHAR2,
    BOOKING_STATUS_DESCRIPTION   VARCHAR2
)
```

```
IS
```

```
BEGIN
```

```
    IF BOOKING_STATUS_CODE IS NULL OR LENGTH(trim(BOOKING_STATUS_CODE))>10 THEN
        raise_application_error(-20001,'BOOKING_STATUS_CODE should not be null and shouldnt be
        more than 10 charectors');
```

```
    END IF;
```

```
    IF BOOKING_STATUS_DESCRIPTION IS NULL THEN
```

```
        raise_application_error(-20002,'BOOKING_STATUS_DESCRIPTION should not be null');
```

```
    END IF;
```

```
    INSERT
```

```
    INTO booking_status VALUES
```

```
(
    upper(BOOKING_STATUS_CODE) ,
    BOOKING_STATUS_DESCRIPTION
);
```

```
    COMMIT;
```

```
    dbms_output.put_line('new Booking status added....');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
    dbms_output.put_line('some thing wrong..operation failed due to '+SUBSTR(SQLERRM,1,200));
```

```
END;
```

```
PROCEDURE add_new_vehicle_category
```

```
(
    CATEGORY_CODE          VARCHAR2 ,
    CATEGORY_DESCRIPTION   VARCHAR2
)
```

```
IS
```

```
BEGIN
```

```
    IF CATEGORY_CODE IS NULL OR LENGTH(trim(CATEGORY_CODE))>10 THEN
        raise_application_error(-20003,'CATEGORY_CODE should not be null and shouldnt be more
        than 10 charectors');
```

```
    END IF;
```

```
    INSERT
```

```
    INTO vehicle_category VALUES
```

```
(
    upper(CATEGORY_CODE) ,
    CATEGORY_DESCRIPTION
);
```

```
    COMMIT;
```

```
    dbms_output.put_line('vehicle category was added....');
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
    dbms_output.put_line('some thing wrong..operation failed');
```

```
END;
```



```

PROCEDURE add_new_manufacturer
(
    MANUFACTURER_CODE    VARCHAR2,
    MANUFACTURER_NAME    VARCHAR2,
    MANUFACTURER_DETALS  VARCHAR2
)
IS
BEGIN
    IF MANUFACTURER_CODE IS NULL OR LENGTH(trim(MANUFACTURER_CODE))>3 THEN
        raise_application_error(-20004,'MANUFACTURER_CODE should not be null and shouldnt be
        more than 3 charectors');
    END IF;
    IF MANUFACTURER_NAME IS NULL THEN
        raise_application_error(-20005,'MANUFACTURER_NAME should not be null ');
    END IF;
    INSERT
    INTO manufacturer VALUES
    (
        upper(MANUFACTURER_CODE),
        MANUFACTURER_NAME,
        MANUFACTURER_DETALS
    );
    COMMIT;

    dbms_output.put_line('Manufacturer added....');
EXCEPTION
WHEN OTHERS THEN
    dbms_output.put_line('some thing wrong..operation failed due to '+SUBSTR(SQLERRM,1,200));
END;

PROCEDURE add_new_model
(
    MODEL_CODE          VARCHAR2,
    DAILY_HIRE_RATE     VARCHAR2,
    MODEL_NAME          VARCHAR2
)
IS
BEGIN
    IF MODEL_CODE IS NULL OR LENGTH(trim(MODEL_CODE))>5 THEN
        raise_application_error(-20006,'MODEL_CODE should not be null and shouldnt be more than
        5 charectors');
    END IF;
    IF DAILY_HIRE_RATE IS NULL OR MODEL_NAME IS NULL THEN
        raise_application_error(-20007,'DAILY_HIRE_RATE and MODEL_NAME should not be null ');
    END IF;
    INSERT
    INTO model VALUES
    (
        upper(MODEL_CODE),
        DAILY_HIRE_RATE,
        MODEL_NAME
    );
    COMMIT;
    dbms_output.put_line('new Model details were added....');
EXCEPTION
WHEN OTHERS THEN
    dbms_output.put_line('some thing wrong..operation failed due to '+SUBSTR(SQLERRM,1,200));
END;

```

```
/*this proc will be called at the time of making payment and it will update the payment flag
to 'Y'*/
PROCEDURE update_payment_status
(
    i_booking_id NUMBER
)
IS
BEGIN
    UPDATE booking SET PAYMENT_RECEIVED_FLAG='Y' WHERE booking_id= i_booking_id;
    COMMIT;
END;

/* we should not execute this query once we are in live..this should be premigration task */
END cars4you;
```