DBMS Project

Tour & Travel Info Services

Submitted by

Pratham Rana	453058	8021074706
Dhvanish Pitroda	453055	8021077647
Pari Patel	453050	8021075429
Pratiksha Jani	453020	8021076474
Ayushi Sharma	453068	8021077806

INDEX

➢ Motive :

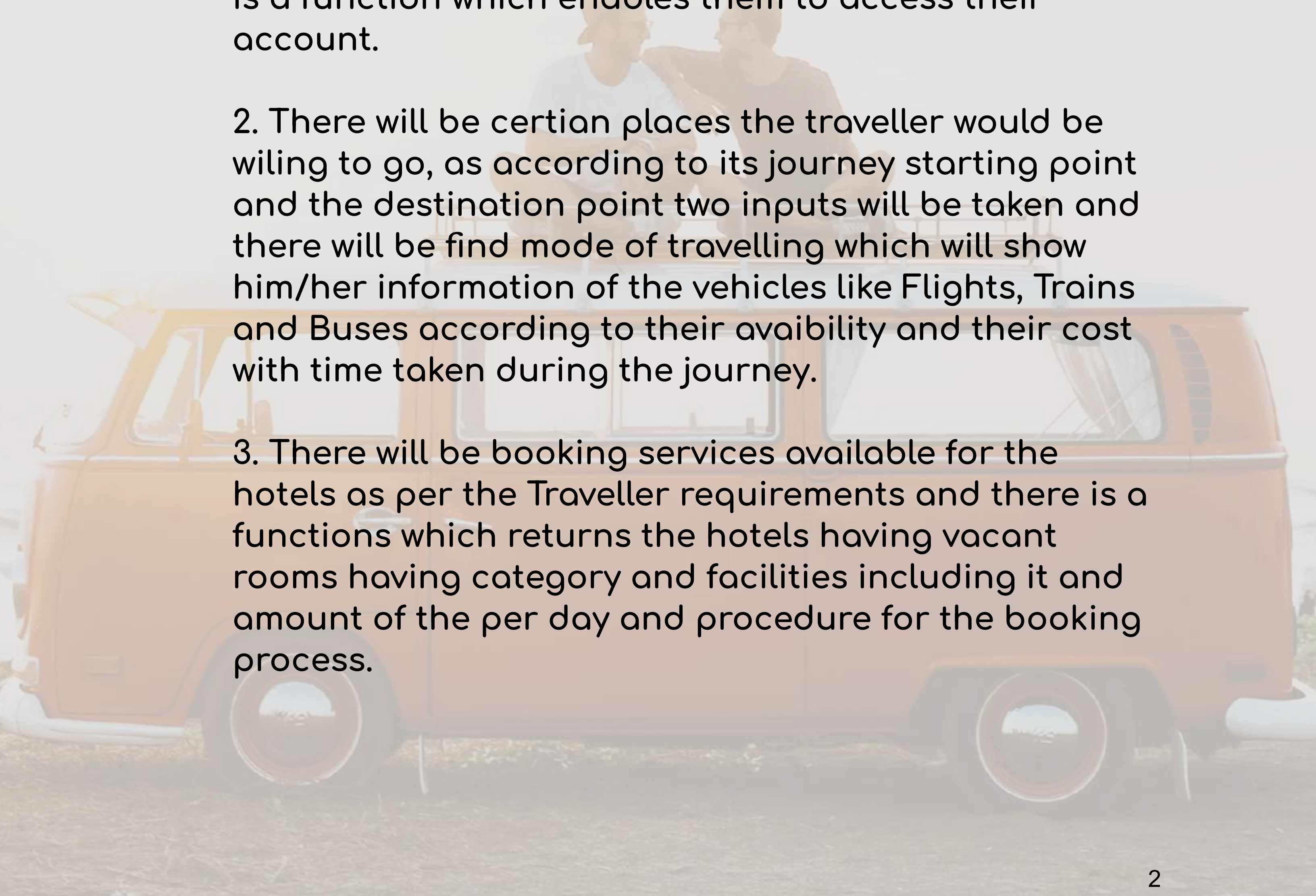
This database manages the tour and travel related information and includes booking of the rooms for the user during its travelling period. Also it includes information of the transportation means available for the location for travelling on the basis of the input provided by the users.

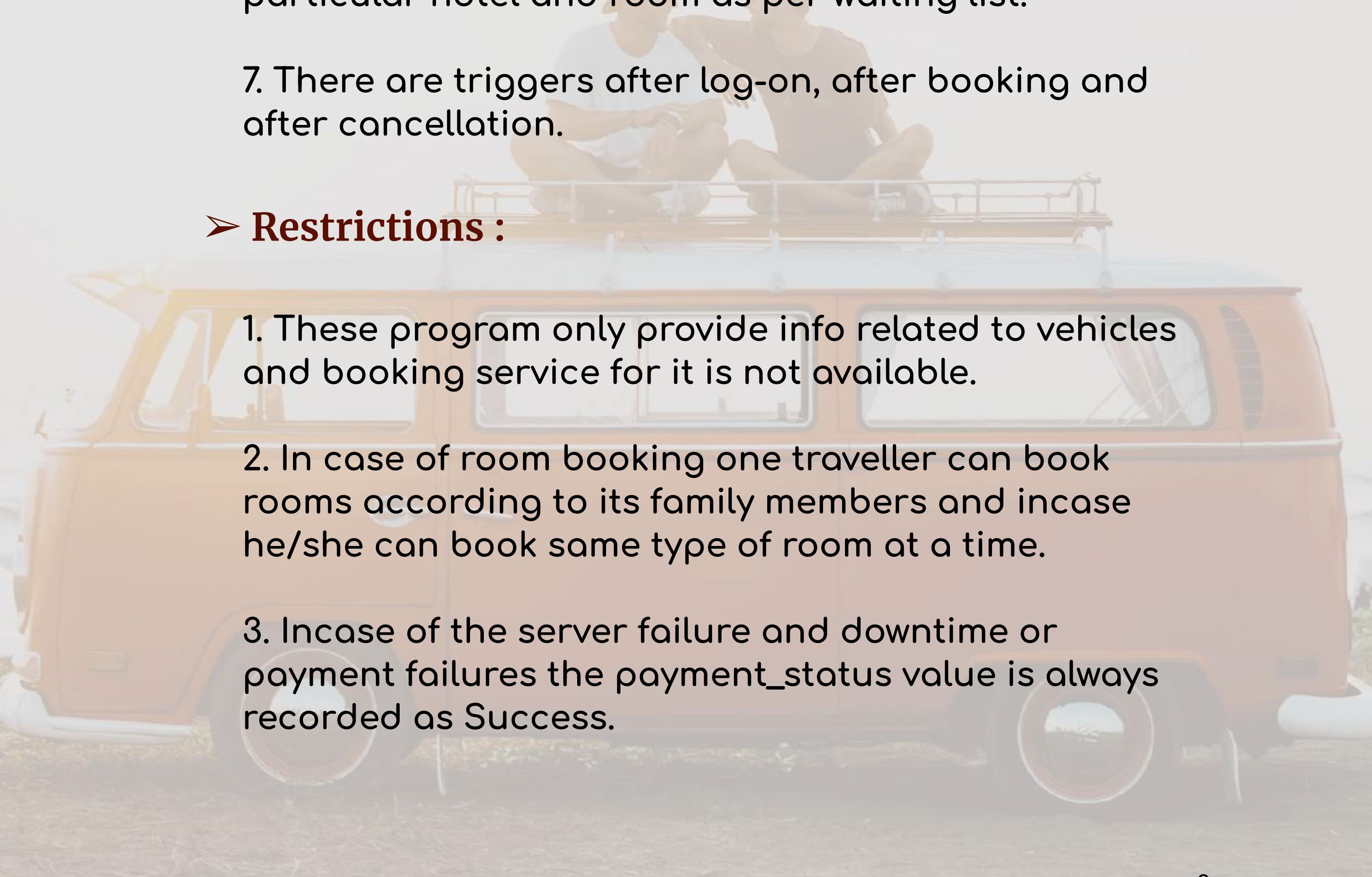
➢ Description :

1. First of all the user need to Sign_in in his/her account and fill the info related to him/her as per required it will be done by sign-in procedure and their is a function which enables them to access their account.

2. There will be certian places the traveller would be wiling to go, as according to its journey starting point and the destination point two inputs will be taken and there will be find mode of travelling which will show him/her information of the vehicles like Flights, Trains and Buses according to their avaibility and their cost with time taken during the journey.

3. There will be booking services available for the hotels as per the Traveller requirements and there is a functions which returns the hotels having vacant rooms having category and facilities including it and amount of the per day and procedure for the booking process.





4. There is a cancellation services which provides the cancellation for the booked rooms of the hotels in case of an cancelletion is required by the travellers and refund criteria will be there for the cancellation.

5. There is a Booking details procedure will stores the data related to the all the entities from booked from and booked until
and the waiting info also for incase if any cancellation occurs the priority will be set by the waiting number.

6. There are procedures that insert, update and delete the data accordingly to the process of after booking and after cancellation
which handles the data of the room booking status and after cancellation also dynamically allocates the particular hotel and room as per waiting list.

7. There are triggers after log-on, after booking and after cancellation.

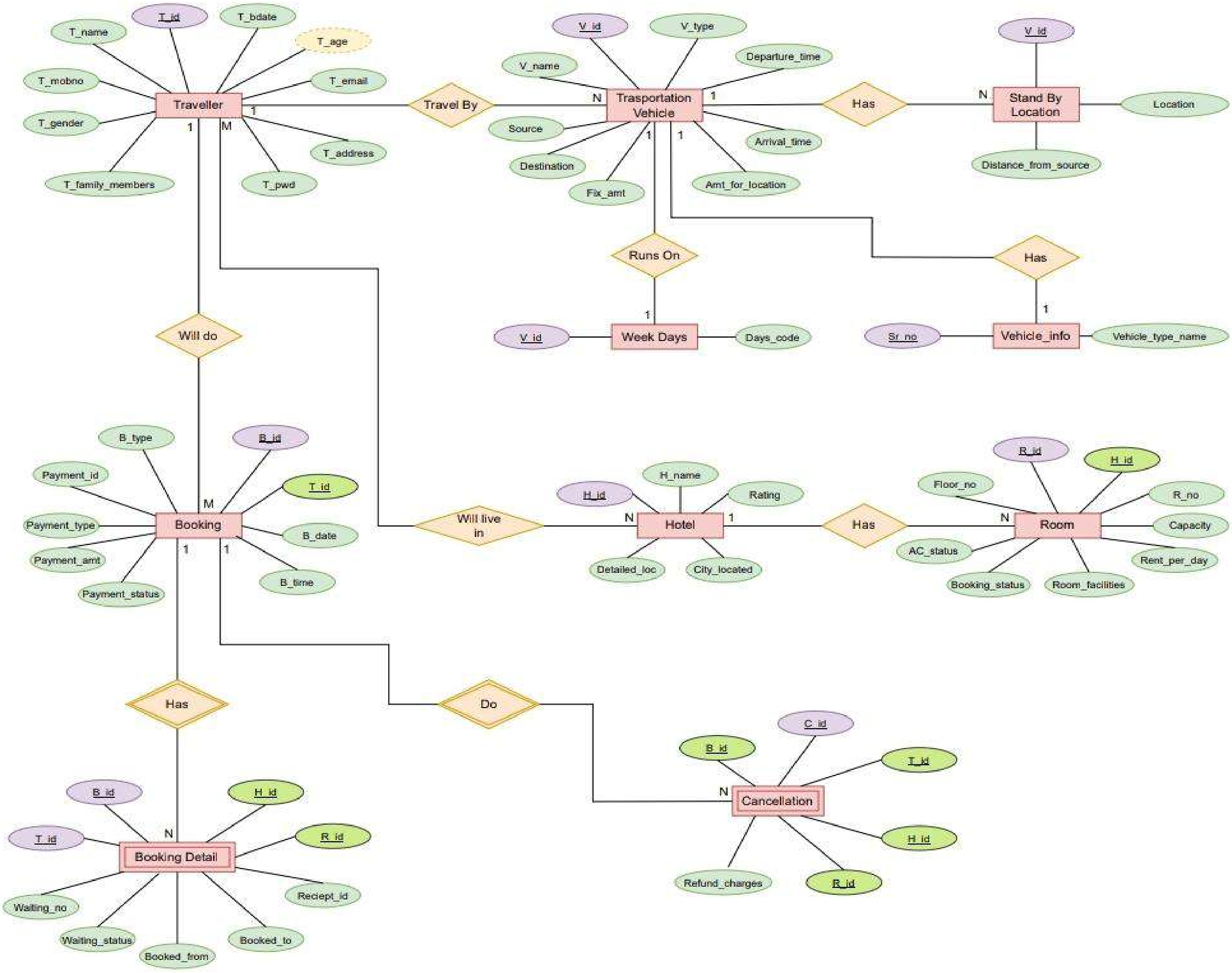
➢ Restrictions :

1. These program only provide info related to vehicles and booking service for it is not available.

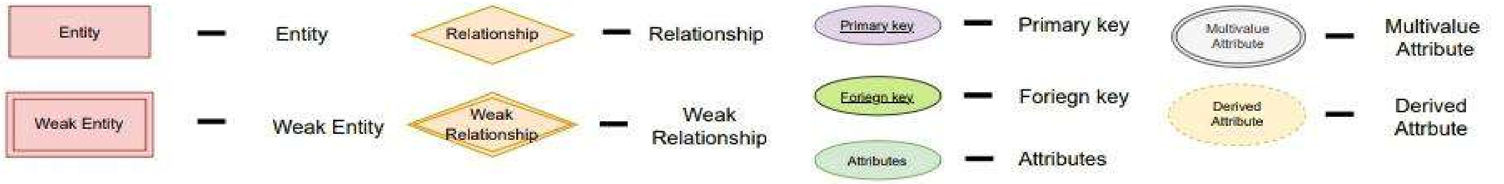
2. In case of room booking one traveller can book rooms according to its family members and incase he/she can book same type of room at a time.

3. Incase of the server failure and downtime or payment failures the payment_status value is always recorded as Success.

ER Diagram of Tour and Travel Info

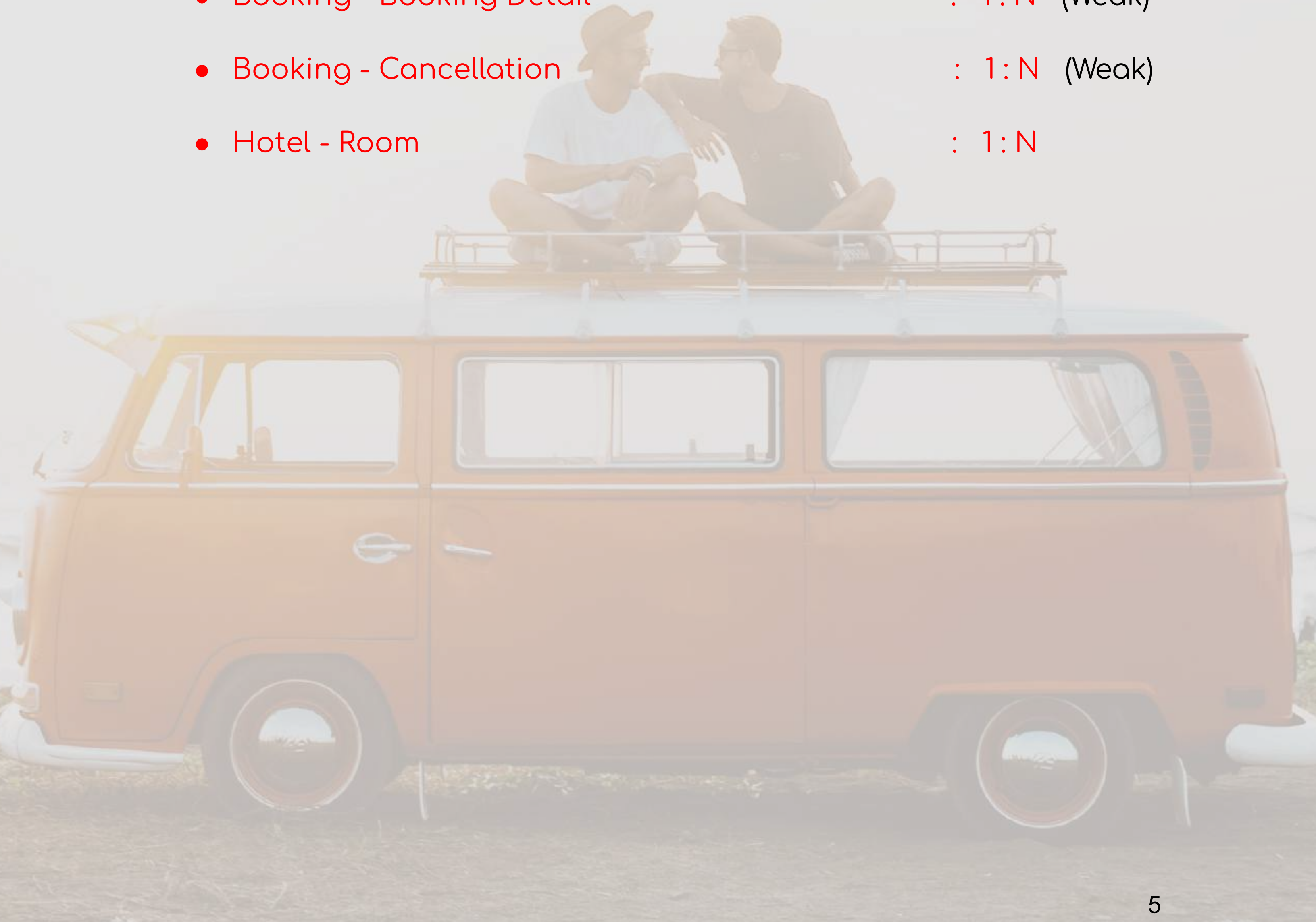


*Notations:



❖ Cardinality :

● Traveller - Transportation Vehicle	:	1 : N	
● Traveller - Booking	:	1 : N	
● Traveller - Hotel	:	M : N	
● Transportation Vehicle - Week Days	:	1 : 1	
● Transportation Vehicle - Vehicle info	:	1 : 1	
● Transportation Vehicle - Stand by Location	:	1 : N	
● Booking - Booking Detail	:	1 : N	(Weak)
● Booking - Cancellation	:	1 : N	(Weak)
● Hotel - Room	:	1 : N	

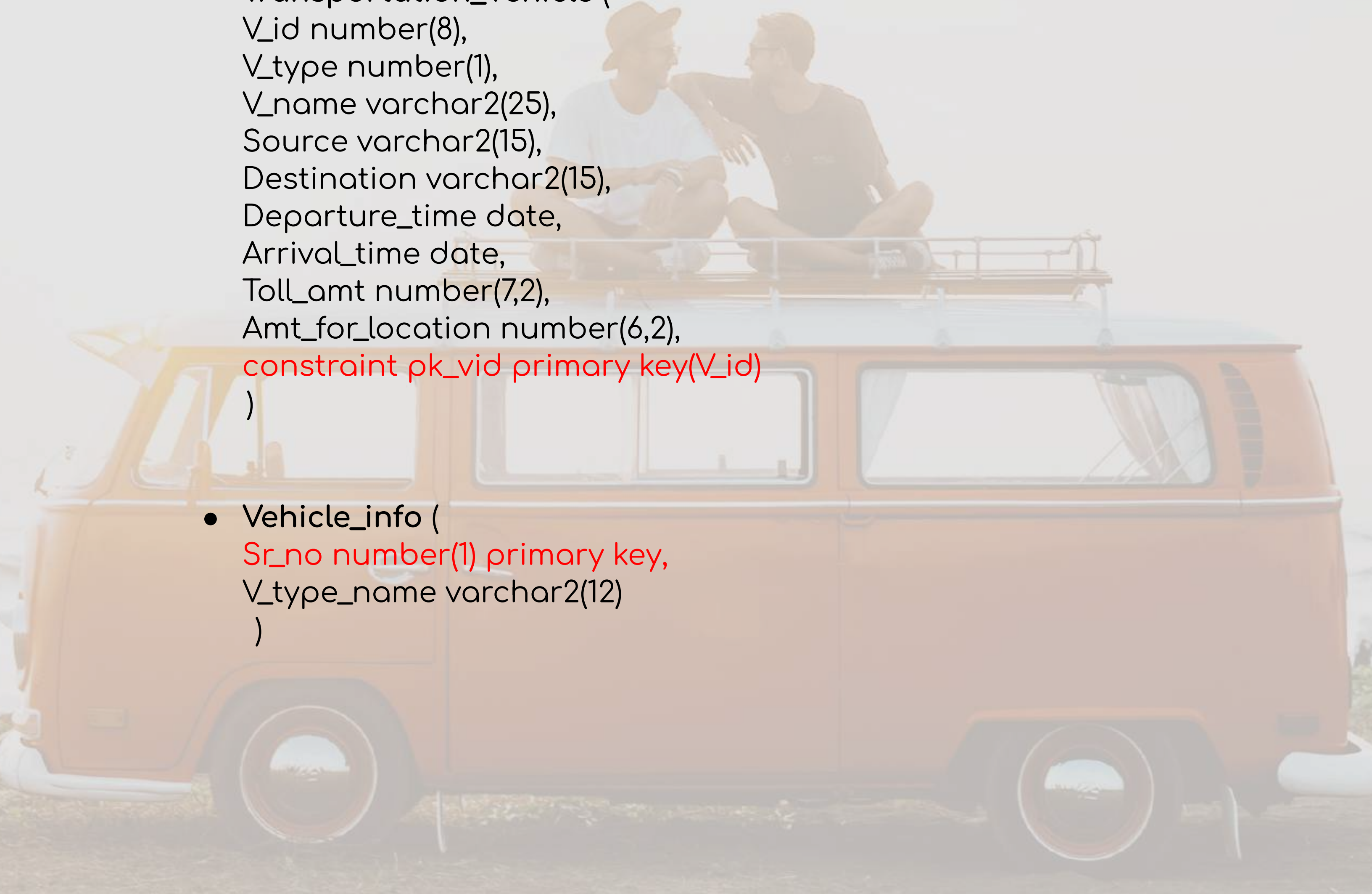


❖ Tables :

- **Traveller** (
  T_id varchar2(12),
  T_name varchar2(25) not null,
  T_age number(3),
  T_mobno number(10),
  T_gender varchar2(1),
  T_address varchar2(40),
  T_email varchar2(40),
  T_family_members number(2),
  T_pwd varchar2(15),
  constraint pk_tid primary key(T_id)
)

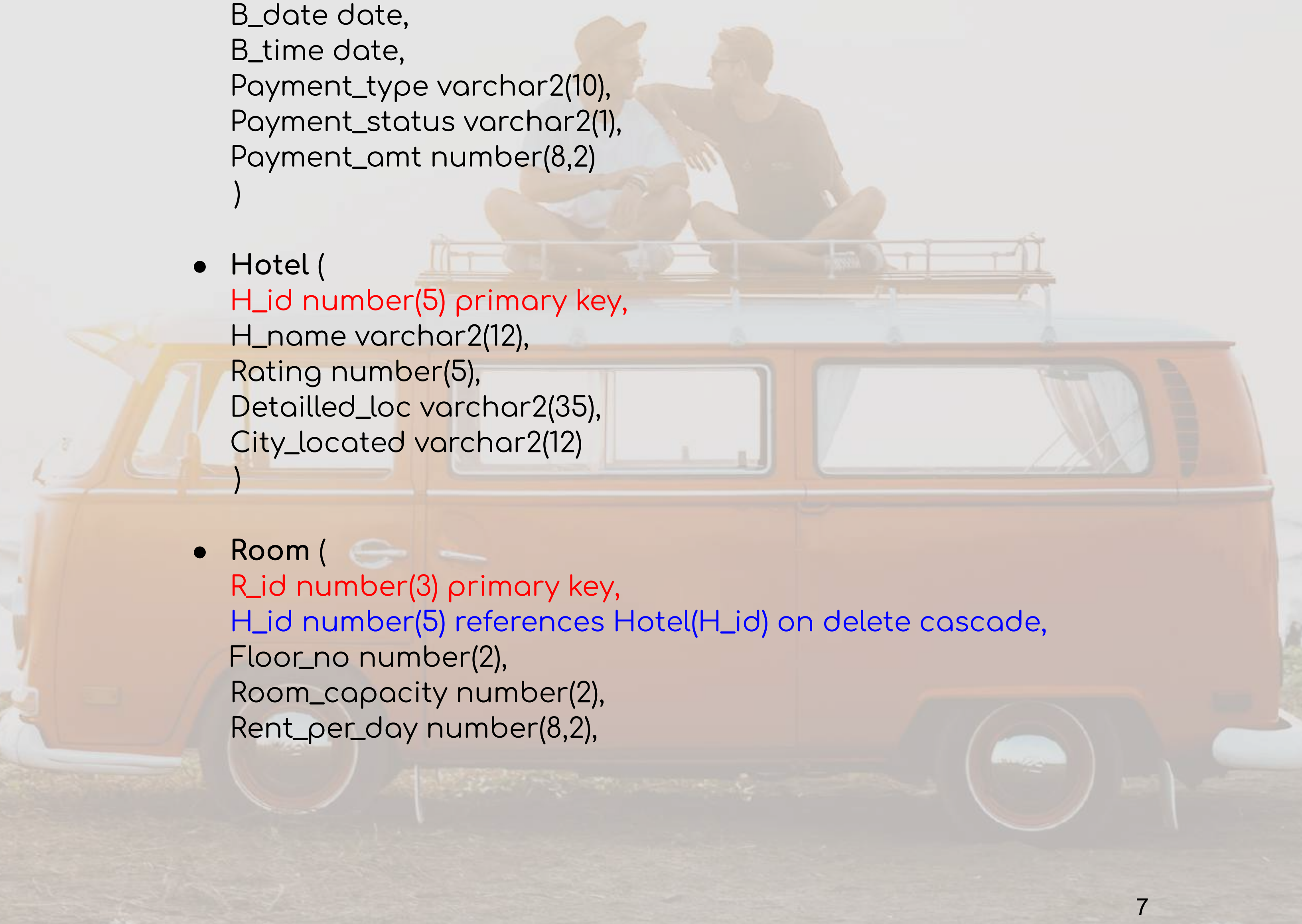
- **Transportation_Vehicle** (
  V_id number(8),
  V_type number(1),
  V_name varchar2(25),
  Source varchar2(15),
  Destination varchar2(15),
  Departure_time date,
  Arrival_time date,
  Toll_amt number(7,2),
  Amt_for_location number(6,2),
  constraint pk_vid primary key(V_id)
)

- **Vehicle_info** (
  Sr_no number(1) primary key,
  V_type_name varchar2(12)
)



- Week_days (
  V_id number(8) references Transportation_Vehicle(V_id) on delete cascade,
  Days_code varchar2(7)
)

- Stand_by_location (
  V_id number(8) references Transportation_Vehicle(V_id) on delete cascade,
  Location varchar2(10),
  Distance_from_source number(4)
)



- Booking (
  B_id number(7) primary key,
  T_id varchar2(12) references Traveller(T_id) on delete cascade,
  B_type varchar2(10),
  B_date date,
  B_time date,
  Payment_type varchar2(10),
  Payment_status varchar2(1),
  Payment_amt number(8,2)
)

- Hotel (
  H_id number(5) primary key,
  H_name varchar2(12),
  Rating number(5),
  Detailled_loc varchar2(35),
  City_located varchar2(12)
)

- Room (
  R_id number(3) primary key,
  H_id number(5) references Hotel(H_id) on delete cascade,
  Floor_no number(2),
  Room_capacity number(2),
  Rent_per_day number(8,2),

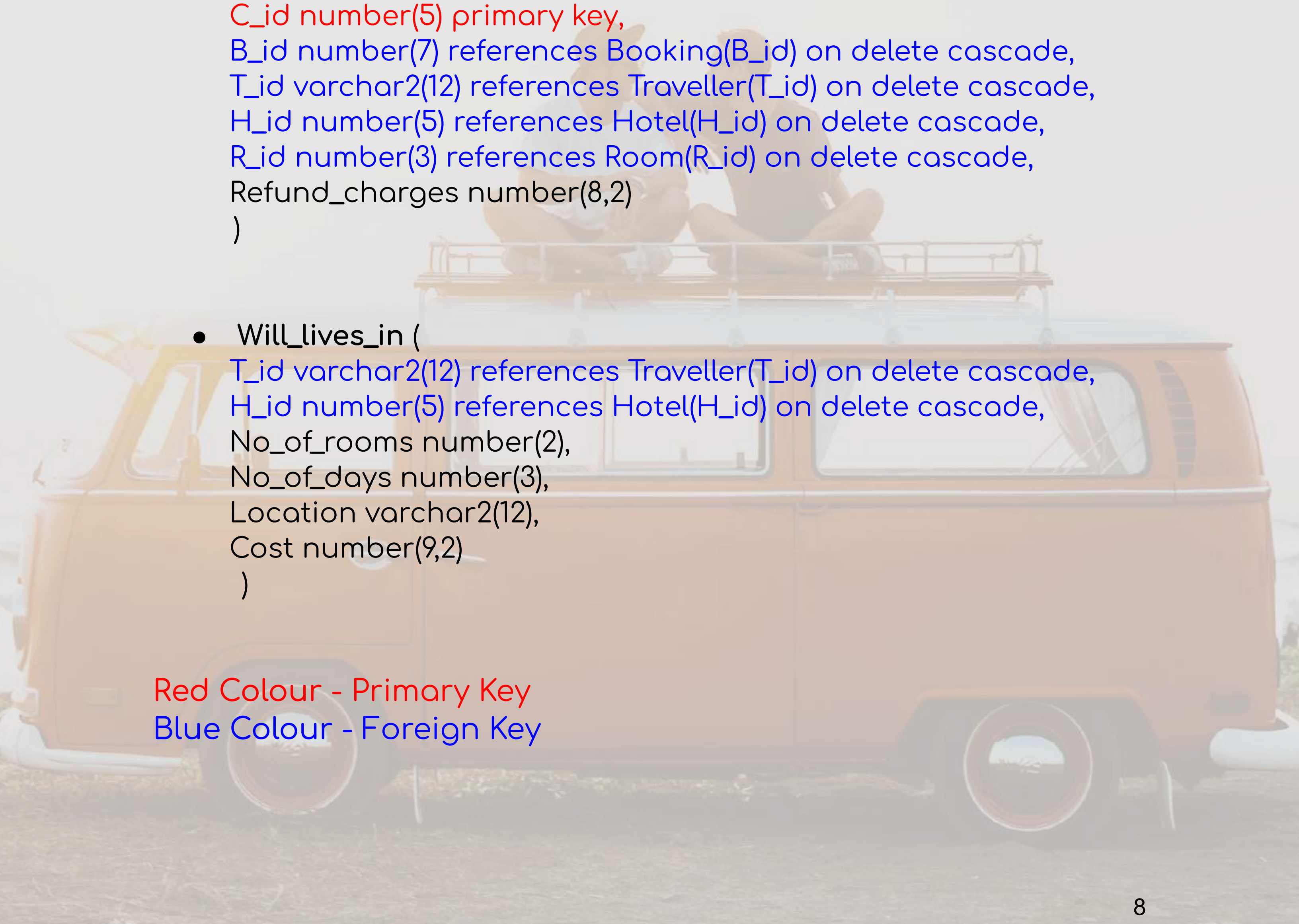
AC_status varchar2(1),
Booking_status varchar2(1),
R_no number(3),
Room_facilites varchar(40)
)

- **Booking_details** (
  H_id number(5) references Hotel(H_id) on delete cascade,
  R_id number(3) references Room(R_id) on delete cascade,
  B_id number(7) references Booking(B_id) on delete cascade,
  T_id varchar2(12) references Traveller(T_id) on delete cascade,
  Booked_from date,
  Booked_till date,
  Amount number(8,2)
)

- **Cancellation** (
  C_id number(5) primary key,
  B_id number(7) references Booking(B_id) on delete cascade,
  T_id varchar2(12) references Traveller(T_id) on delete cascade,
  H_id number(5) references Hotel(H_id) on delete cascade,
  R_id number(3) references Room(R_id) on delete cascade,
  Refund_charges number(8,2)
)

- **Will_lives_in** (
  T_id varchar2(12) references Traveller(T_id) on delete cascade,
  H_id number(5) references Hotel(H_id) on delete cascade,
  No_of_rooms number(2),
  No_of_days number(3),
  Location varchar2(12),
  Cost number(9,2)
)

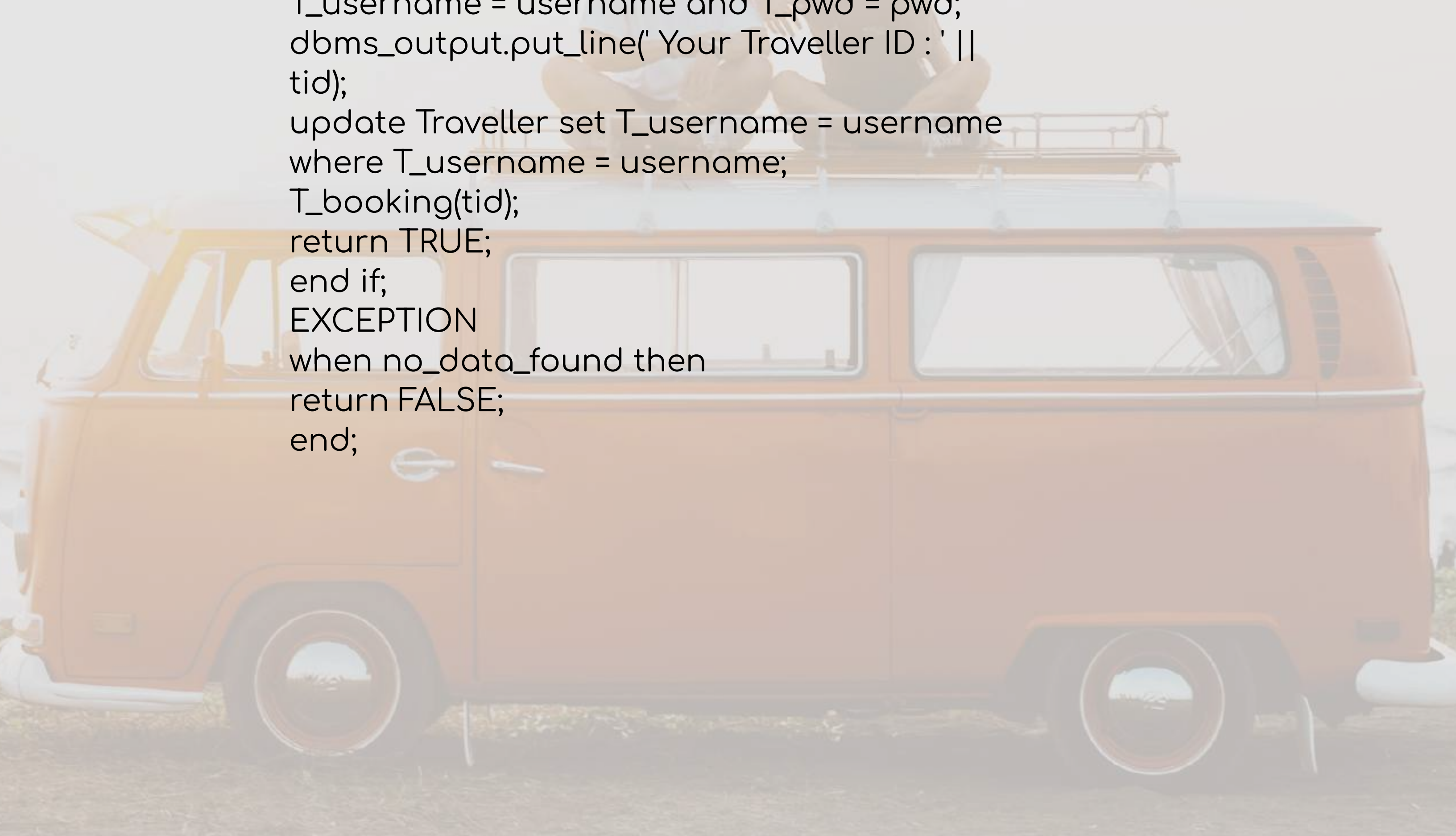
Red Colour - Primary Key
Blue Colour - Foreign Key



❖ FUNCTIONS :

1) User Login

```
CREATE OR REPLACE FUNCTION User_login
return boolean
is
username Traveller.T_username%type;
pwd Traveller.T_pwd%type;
u_name Traveller.T_username%type;
u_pwd Traveller.T_pwd%type;
tid Traveller.T_id%type;
begin
-- username := '&username';
-- pwd := '&pwd';
select T_username, T_pwd into u_name, u_pwd from
Traveller where T_username = username and T_pwd = pwd;
if (username = u_name and pwd = u_pwd) then
select T_id into tid from Traveller where
T_username = username and T_pwd = pwd;
dbms_output.put_line(' Your Traveller ID : ' ||
tid);
update Traveller set T_username = username
where T_username = username;
T_booking(tid);
return TRUE;
end if;
EXCEPTION
when no_data_found then
return FALSE;
end;
```

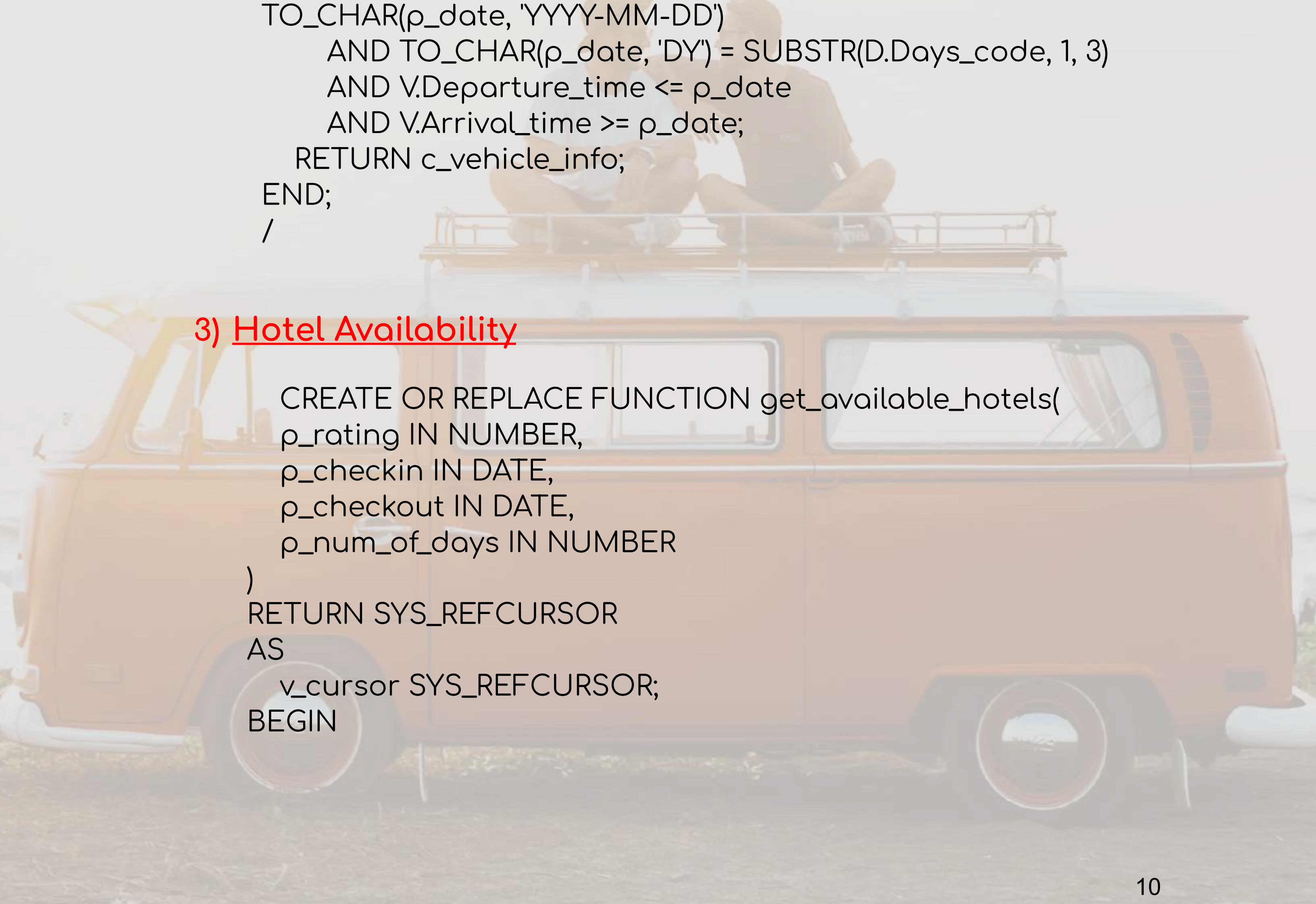


2) Transportation

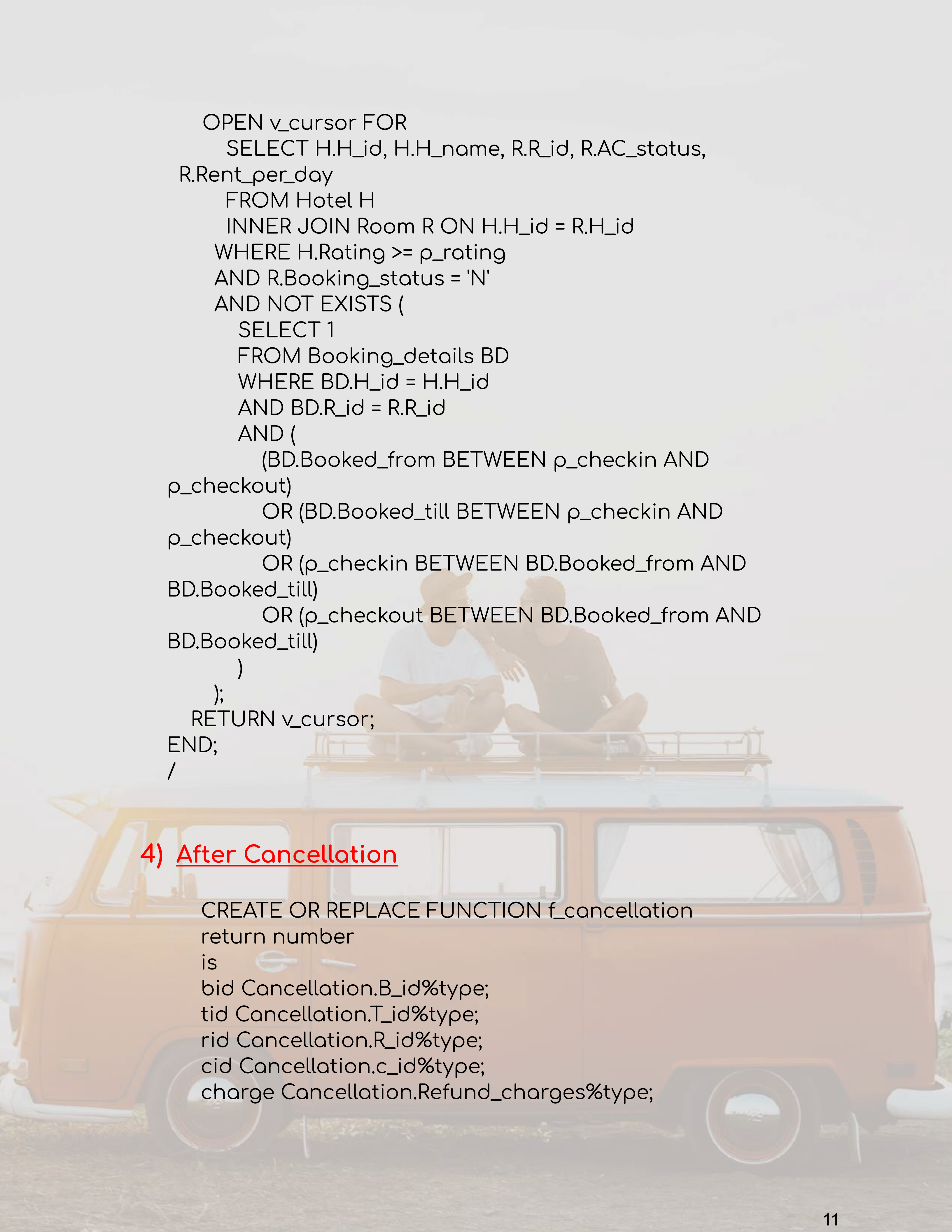
```
CREATE OR REPLACE FUNCTION get_vehicle_info (
p_source IN VARCHAR2,
p_destination IN VARCHAR2,
p_date IN DATE
)
RETURN SYS_REFCURSOR
IS
   c_vehicle_info SYS_REFCURSOR;
BEGIN
   OPEN c_vehicle_info FOR
      SELECT V.V_id, V.Departure_time, V.Arrival_time
      FROM Transportation_Vehicle V
      JOIN Week_days D ON V.V_id = D.V_id
      WHERE V.Source = p_source
      AND V.Destination = p_destination
      AND TO_CHAR(V.Departure_time, 'YYYY-MM-DD') =
TO_CHAR(p_date, 'YYYY-MM-DD')
      AND TO_CHAR(p_date, 'DY') = SUBSTR(D.Days_code, 1, 3)
      AND V.Departure_time <= p_date
      AND V.Arrival_time >= p_date;
   RETURN c_vehicle_info;
END;
/
```

3) Hotel Availability

```
   CREATE OR REPLACE FUNCTION get_available_hotels(
   p_rating IN NUMBER,
   p_checkin IN DATE,
   p_checkout IN DATE,
   p_num_of_days IN NUMBER
)
RETURN SYS_REFCURSOR
AS
   v_cursor SYS_REFCURSOR;
BEGIN
```

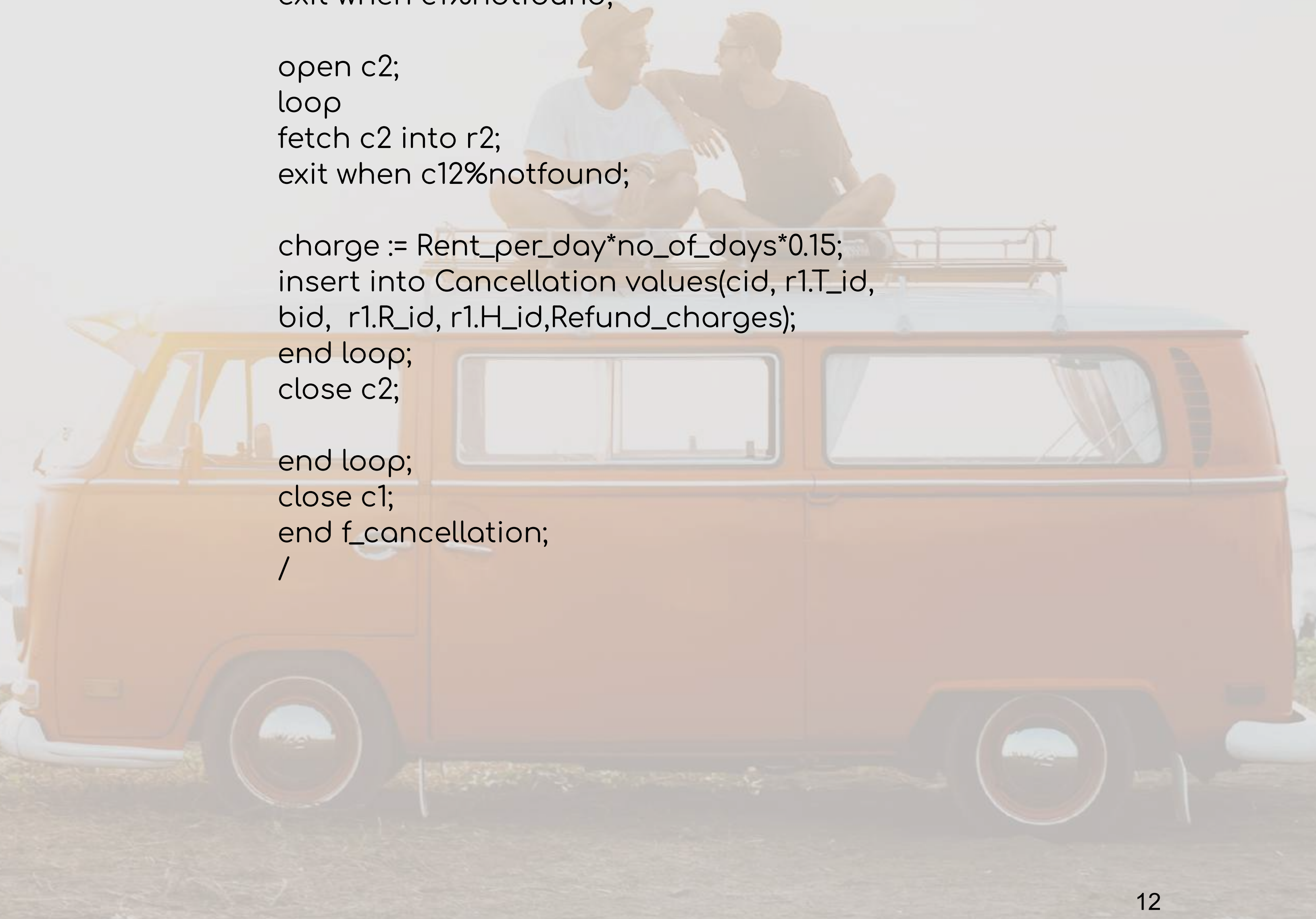


```
    OPEN v_cursor FOR
        SELECT H.H_id, H.H_name, R.R_id, R.AC_status,
  R.Rent_per_day
        FROM Hotel H
        INNER JOIN Room R ON H.H_id = R.H_id
      WHERE H.Rating >= p_rating
      AND R.Booking_status = 'N'
      AND NOT EXISTS (
          SELECT 1
          FROM Booking_details BD
          WHERE BD.H_id = H.H_id
          AND BD.R_id = R.R_id
          AND (
              (BD.Booked_from BETWEEN p_checkin AND
p_checkout)
              OR (BD.Booked_till BETWEEN p_checkin AND
p_checkout)
              OR (p_checkin BETWEEN BD.Booked_from AND
BD.Booked_till)
              OR (p_checkout BETWEEN BD.Booked_from AND
BD.Booked_till)
          )
      );
    RETURN v_cursor;
END;
/
```



4) After Cancellation

```
CREATE OR REPLACE FUNCTION f_cancellation
return number
is
bid Cancellation.B_id%type;
tid Cancellation.T_id%type;
rid Cancellation.R_id%type;
cid Cancellation.c_id%type;
charge Cancellation.Refund_charges%type;
```

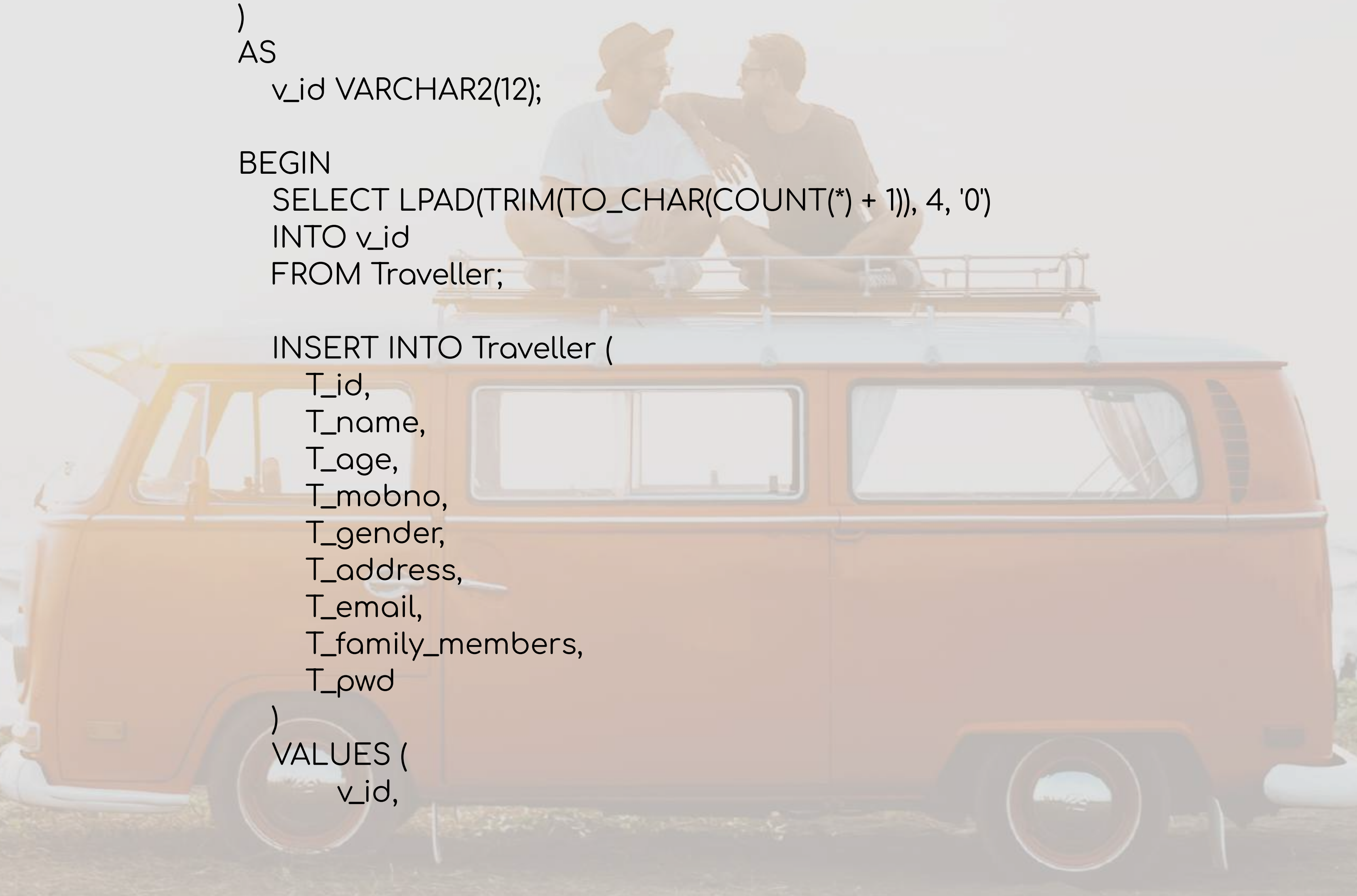


```
cursor c1 is select T_id,H_id,R_id,no_of_days
 from Booking_details where B_id =bid;

cursor c2 is select R_id,H_id,Rent_per_dayfrom Room where
R_id=rid;
r1 c1%rowtype;
r2 c2%rowtype;

begin
-- bid := &bid;
--rid :=&rid;
select count(*) into cid from Cancellation;

open c1;
loop
fetch c1 into r1;
exit when c1%notfound;

open c2;
loop
fetch c2 into r2;
exit when c12%notfound;

charge := Rent_per_day*no_of_days*0.15;
insert into Cancellation values(cid, r1.T_id,
bid,  r1.R_id, r1.H_id,Refund_charges);
end loop;
close c2;

end loop;
close c1;
end f_cancellation;
/
```

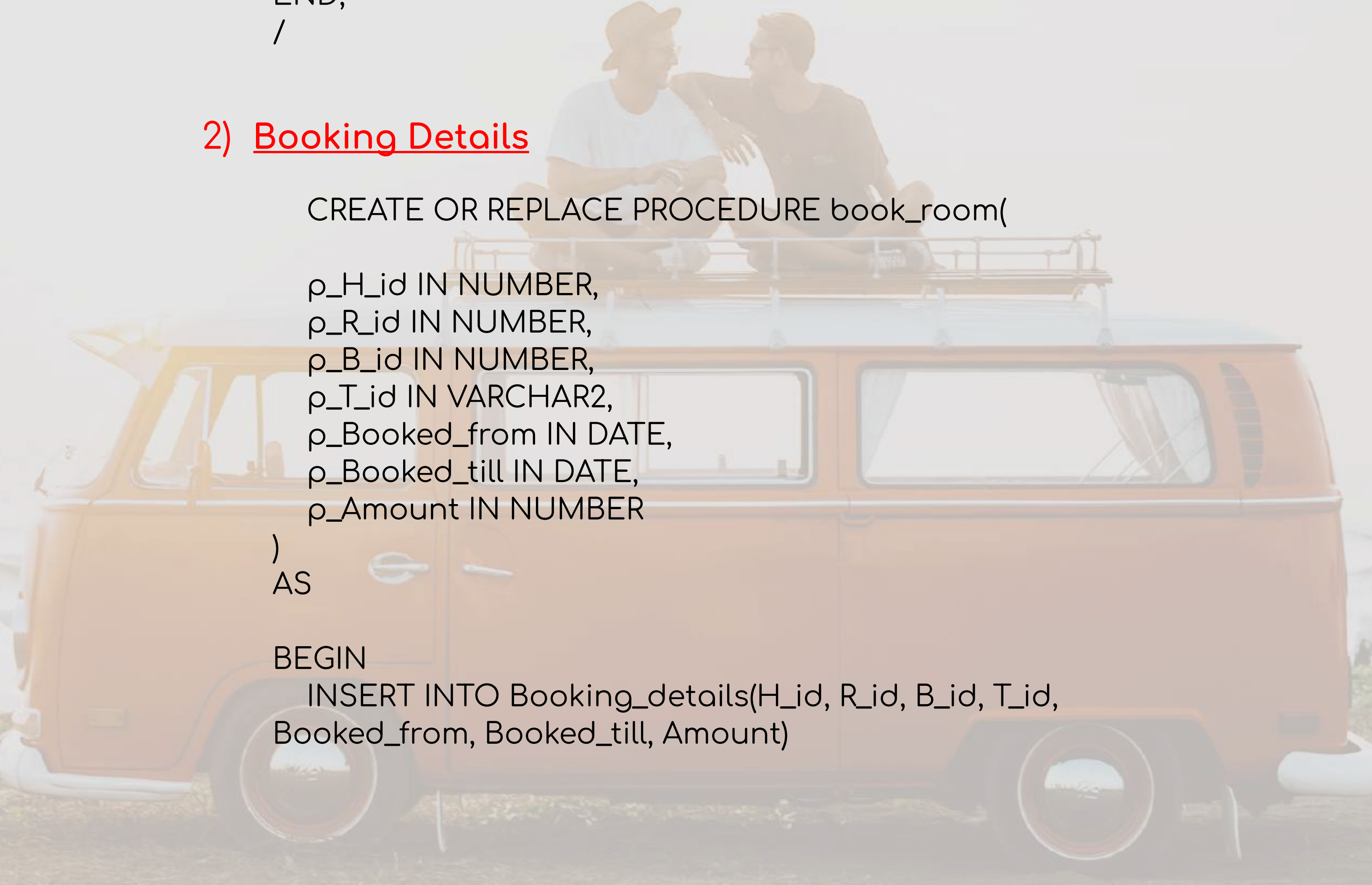
❖ PROCEDURES :

1) User Sign Up

```
    CREATE OR REPLACE PROCEDURE user_signup (
    p_name IN VARCHAR2,
    p_age IN NUMBER,
    p_mobno IN NUMBER,
    p_gender IN VARCHAR2,
    p_address IN VARCHAR2,
    p_email IN VARCHAR2,
    p_family_members IN NUMBER,
    p_pwd IN VARCHAR2
)
AS
    v_id VARCHAR2(12);

BEGIN
    SELECT LPAD(TRIM(TO_CHAR(COUNT(*) + 1)), 4, '0')
    INTO v_id
    FROM Traveller;

    INSERT INTO Traveller (
        T_id,
        T_name,
        T_age,
        T_mobno,
        T_gender,
        T_address,
        T_email,
        T_family_members,
        T_pwd
    )
    VALUES (
            v_id,
```



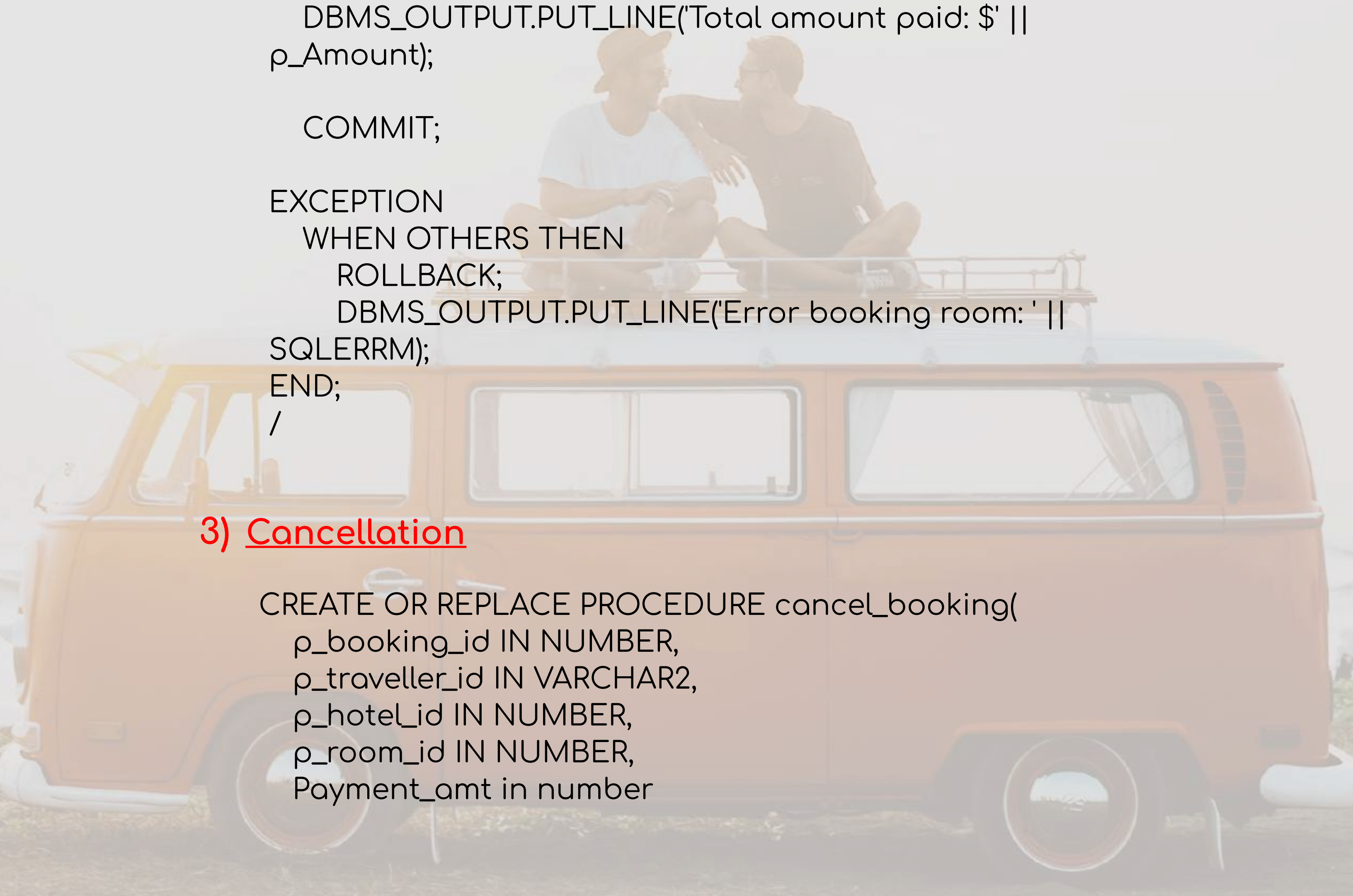
```
        p_name,
        p_age,
        p_mobno,
        p_gender,
        p_address,
        p_email,
        p_family_members,
        p_pwd
    );

    DBMS_OUTPUT.PUT_LINE('User ' || p_name || ' has been
successfully signed up with ID ' || v_id);

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ' ||
SQLERRM);
END;
/
```



2) Booking Details

```
    CREATE OR REPLACE PROCEDURE book_room(

    p_H_id IN NUMBER,
    p_R_id IN NUMBER,
    p_B_id IN NUMBER,
    p_T_id IN VARCHAR2,
    p_Booked_from IN DATE,
    p_Booked_till IN DATE,
    p_Amount IN NUMBER
)
AS

BEGIN
    INSERT INTO Booking_details(H_id, R_id, B_id, T_id,
Booked_from, Booked_till, Amount)
```

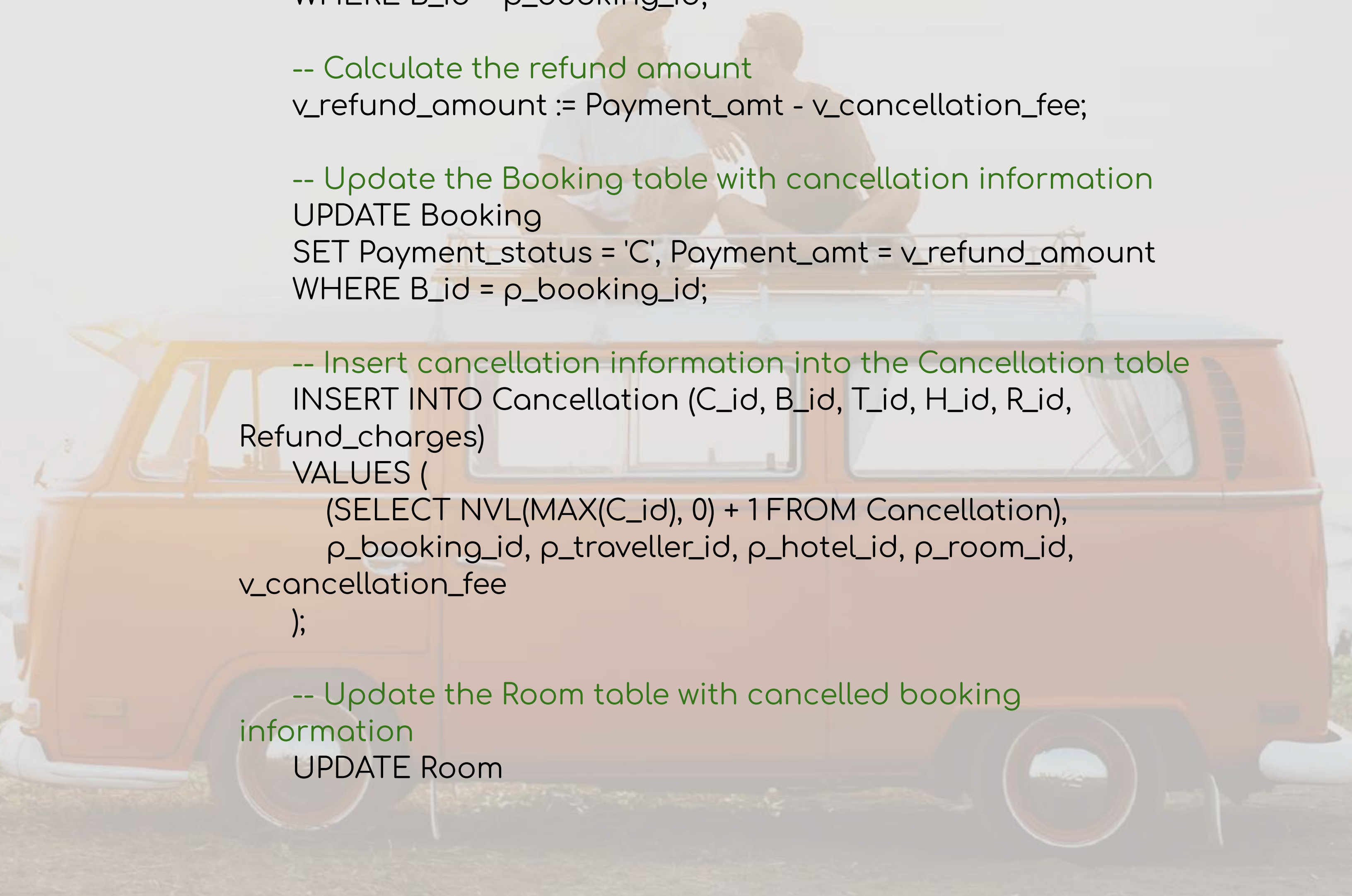
```
    VALUES(p_H_id, p_R_id, p_B_id, p_T_id, p_Booked_from,
p_Booked_till, p_Amount);

    UPDATE Room
    SET Booking_status = 'B'
    WHERE R_id = p_R_id;

    UPDATE Booking
    SET Payment_type = 'Credit Card',
        Payment_status = 'P',
        Payment_amt = p_Amount
    WHERE B_id = p_B_id;

    DBMS_OUTPUT.PUT_LINE('Booking confirmed for room ' ||
p_R_id || ' at hotel ' || p_H_id);
    DBMS_OUTPUT.PUT_LINE('Booked from ' || p_Booked_from ||
' to ' || p_Booked_till);
    DBMS_OUTPUT.PUT_LINE('Total amount paid: $' ||
p_Amount);

    COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error booking room: ' ||
SQLERRM);
END;
/
```



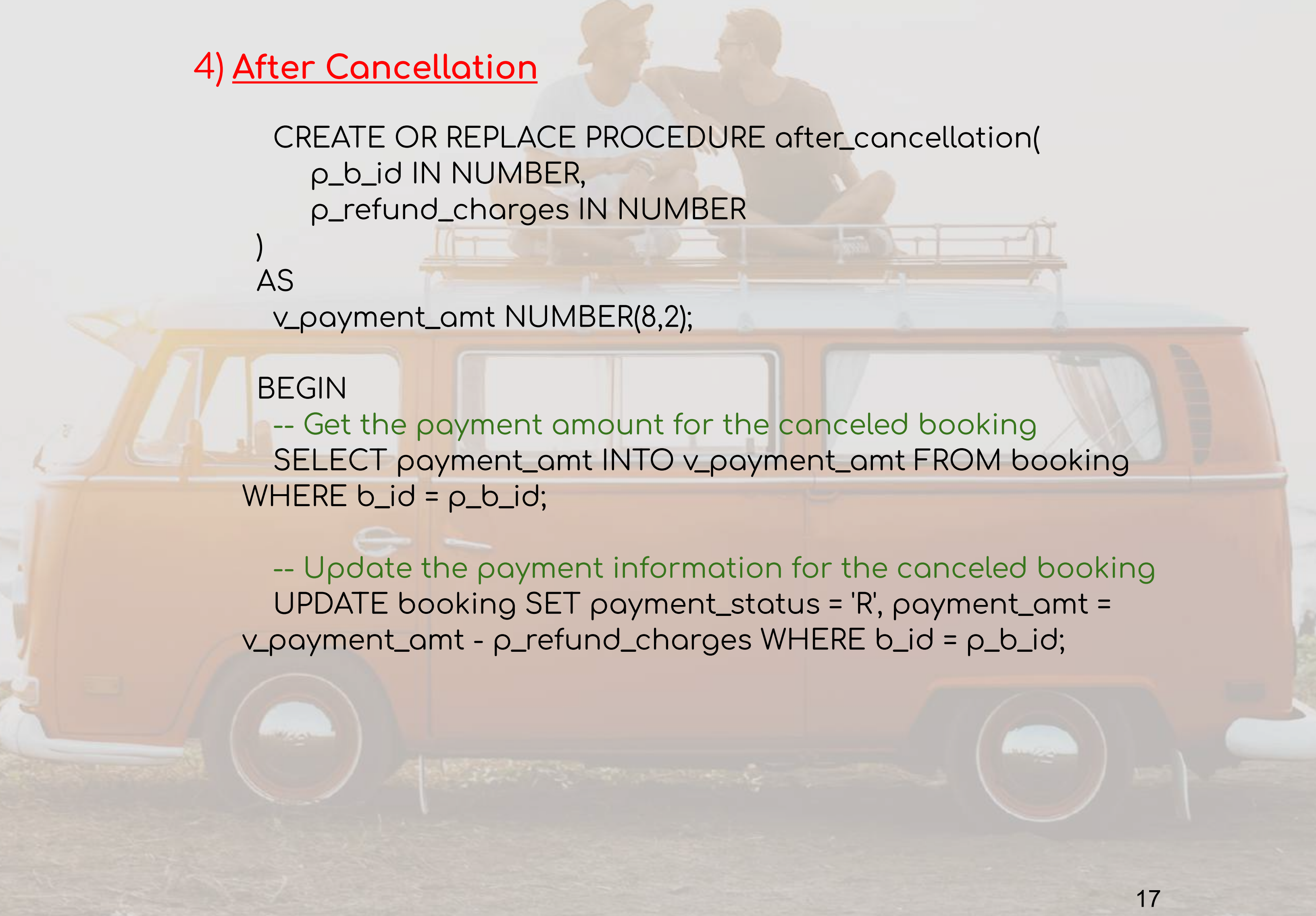
3) Cancellation

```
CREATE OR REPLACE PROCEDURE cancel_booking(
    p_booking_id IN NUMBER,
    p_traveller_id IN VARCHAR2,
    p_hotel_id IN NUMBER,
    p_room_id IN NUMBER,
    Payment_amt in number
```

```
 )
  AS
     v_booking_date DATE;
     v_cancellation_fee NUMBER(8,2);
     v_refund_amount NUMBER(8,2);
 BEGIN

     SELECT B_date, CASE
                    WHEN SYSDATE >= B_date - 1 THEN Payment_amt *
0.5 -- cancellation policy: 50% refund if cancelled within 1 day of
booking date
                    ELSE 0 -- no refund if cancelled before 1 day of
booking date
                 END;

     INTO v_booking_date, v_cancellation_fee
     FROM Booking
     WHERE B_id = p_booking_id;

     -- Calculate the refund amount
     v_refund_amount := Payment_amt - v_cancellation_fee;

     -- Update the Booking table with cancellation information
     UPDATE Booking
     SET Payment_status = 'C', Payment_amt = v_refund_amount
     WHERE B_id = p_booking_id;

     -- Insert cancellation information into the Cancellation table
     INSERT INTO Cancellation (C_id, B_id, T_id, H_id, R_id,
Refund_charges)
     VALUES (
        (SELECT NVL(MAX(C_id), 0) + 1 FROM Cancellation),
        p_booking_id, p_traveller_id, p_hotel_id, p_room_id,
v_cancellation_fee
     );

     -- Update the Room table with cancelled booking
information
     UPDATE Room
```



```
    SET Booking_status = 'A'
    WHERE R_id = p_room_id;

    -- Commit the transaction
    COMMIT;

    -- Display the refund amount to the user
    DBMS_OUTPUT.PUT_LINE('Refund amount: ' ||
v_refund_amount);

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Booking not found.');
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
  END;
  /
```



4) After Cancellation

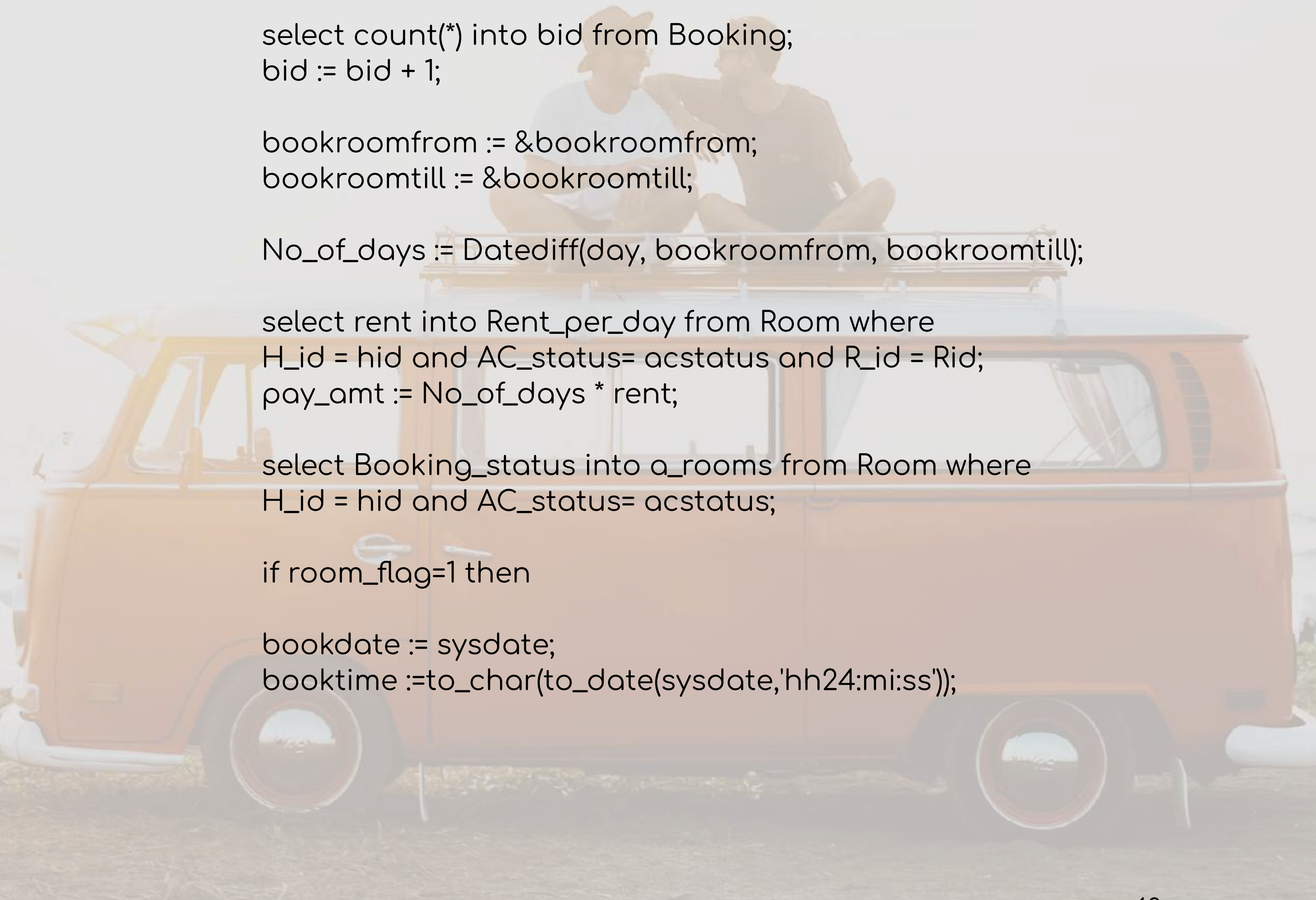
```
  CREATE OR REPLACE PROCEDURE after_cancellation(
      p_b_id IN NUMBER,
      p_refund_charges IN NUMBER
 )
 AS
  v_payment_amt NUMBER(8,2);

 BEGIN
  -- Get the payment amount for the canceled booking
  SELECT payment_amt INTO v_payment_amt FROM booking
WHERE b_id = p_b_id;

  -- Update the payment information for the canceled booking
  UPDATE booking SET payment_status = 'R', payment_amt =
v_payment_amt - p_refund_charges WHERE b_id = p_b_id;
```

```
  -- Update the booking status for the canceled booking's
room
  UPDATE room SET booking_status = 'N' WHERE r_id = (SELECT
r_id FROM booking_details WHERE b_id = p_b_id);

  -- Delete the booking details for the canceled booking
  DELETE FROM booking_details WHERE b_id = p_b_id;

  -- Delete the cancellation record
  DELETE FROM cancellation WHERE b_id = p_b_id;

  DBMS_OUTPUT.PUT_LINE('Booking ' || p_b_id || ' canceled
successfully.');

 EXCEPTION
  WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('Booking ' || p_b_id || ' not found.');
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('Error canceling booking ' ||
p_b_id || ': ' || SQLERRM);
 END;
 /
```

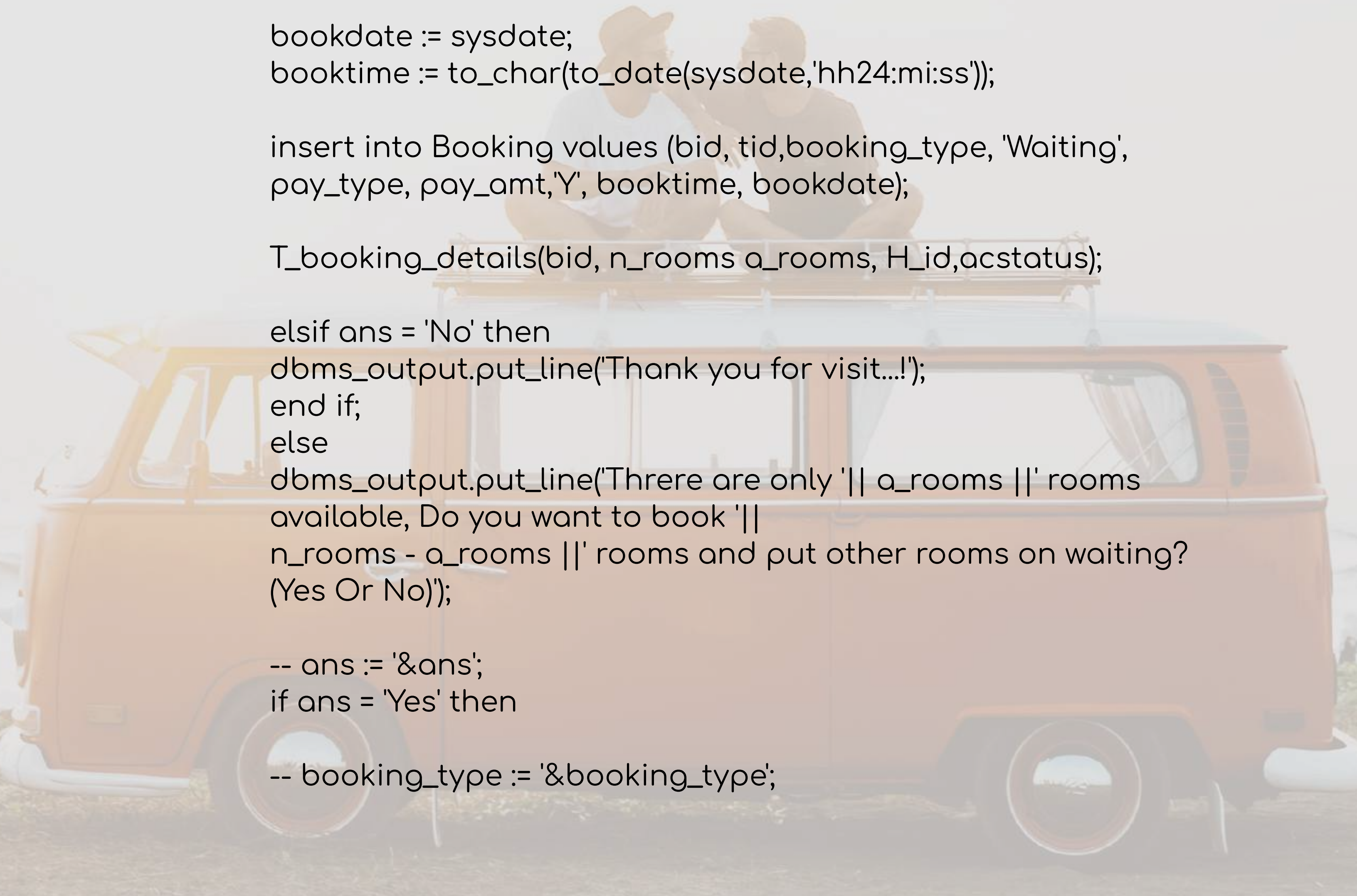
5) Hotel Booking

```
CREATE OR REPLACE PROCEDURE T_Booking
(tid in traveller.T_id%type)
is
hid Hotel.H_id%type;
room_flag number(1);
acstatus room.AC_status%type;
a_rooms number(3);
n_rooms number(3);
No_of_days number(2);
bid Booking.b_id%type;
```

```
booking_type Booking.b_type%type;
pay_type Booking.payment_type%type;
pay_amt Booking.payment_amount%type;
rent Room.Rent_per_day%type;
booktime Booking.B_time%type;
bookdate Booking.B_date%type;
bookroomfrom Booking_details.Booked_from%type;
bookroomtill Booking_details.Booked_till%type;

begin
    H_id := Hotel();
    R_id := &rid; -- Will be selected from the available rooms
    acstatus := &acstatus;


  room_flag := room_status(H_id,R_id); -- Will return if room is
empty or not at an instinct

select count(*) into bid from Booking;
bid := bid + 1;

bookroomfrom := &bookroomfrom;
bookroomtill := &bookroomtill;

No_of_days := Datediff(day, bookroomfrom, bookroomtill);

select rent into Rent_per_day from Room where
H_id = hid and AC_status= acstatus and R_id = Rid;
pay_amt := No_of_days * rent;

select Booking_status into a_rooms from Room where
H_id = hid and AC_status= acstatus;

if room_flag=1 then

bookdate := sysdate;
booktime :=to_char(to_date(sysdate,'hh24:mi:ss'));
```



```
insert into Booking values (bid, tid, booking_type, booktime,
bookdate, 'Success', payid, pay_type, pay_amt,'Success');

T_booking_details(bid,n_rooms,a_rooms,H_id,acstatus);

elsif (seat_flag = -1) then

dbms_output.put_line('There are no rooms available... Still Do
you want to book? (Yes Or No)');

-- ans := '&ans';
if ans = 'Yes' then

dbms_output.put_line('Your Rooms is on waiting...');

-- booking_type := '&booking_type';
-- pay_type := '&pay_type';

bookdate := sysdate;
booktime := to_char(to_date(sysdate,'hh24:mi:ss'));

insert into Booking values (bid, tid,booking_type, 'Waiting',
pay_type, pay_amt,'Y', booktime, bookdate);

T_booking_details(bid, n_rooms a_rooms, H_id,acstatus);

elsif ans = 'No' then
dbms_output.put_line('Thank you for visit...!');
end if;
else
dbms_output.put_line('Threre are only '|| a_rooms ||' rooms
available, Do you want to book '||
n_rooms - a_rooms ||' rooms and put other rooms on waiting?
(Yes Or No)');

-- ans := '&ans';
if ans = 'Yes' then

-- booking_type := '&booking_type';
```



```
  -- pay_type := '&pay_type';

  bookdate := sysdate;
  booktime :=to_char(to_date(sysdate,'hh24:mi:ss'));

  dbms_output.put_line('Your '|| n_rooms - a_rooms || ' rooms
  are confirmed and other rooms on waiting.');

  insert into Booking values (bid, tid,booking_type, 'Waiting',
  payid, pay_type, pay_amt,'Success', booktime, bookdate);
  p_booking_details(bid, a_rooms,H_id,acstatus);
  elsif ans = 'No' then
  dbms_output.put_line('Thank you for visit...!');
      end if;

  end if;

  end T_booking;
/
```

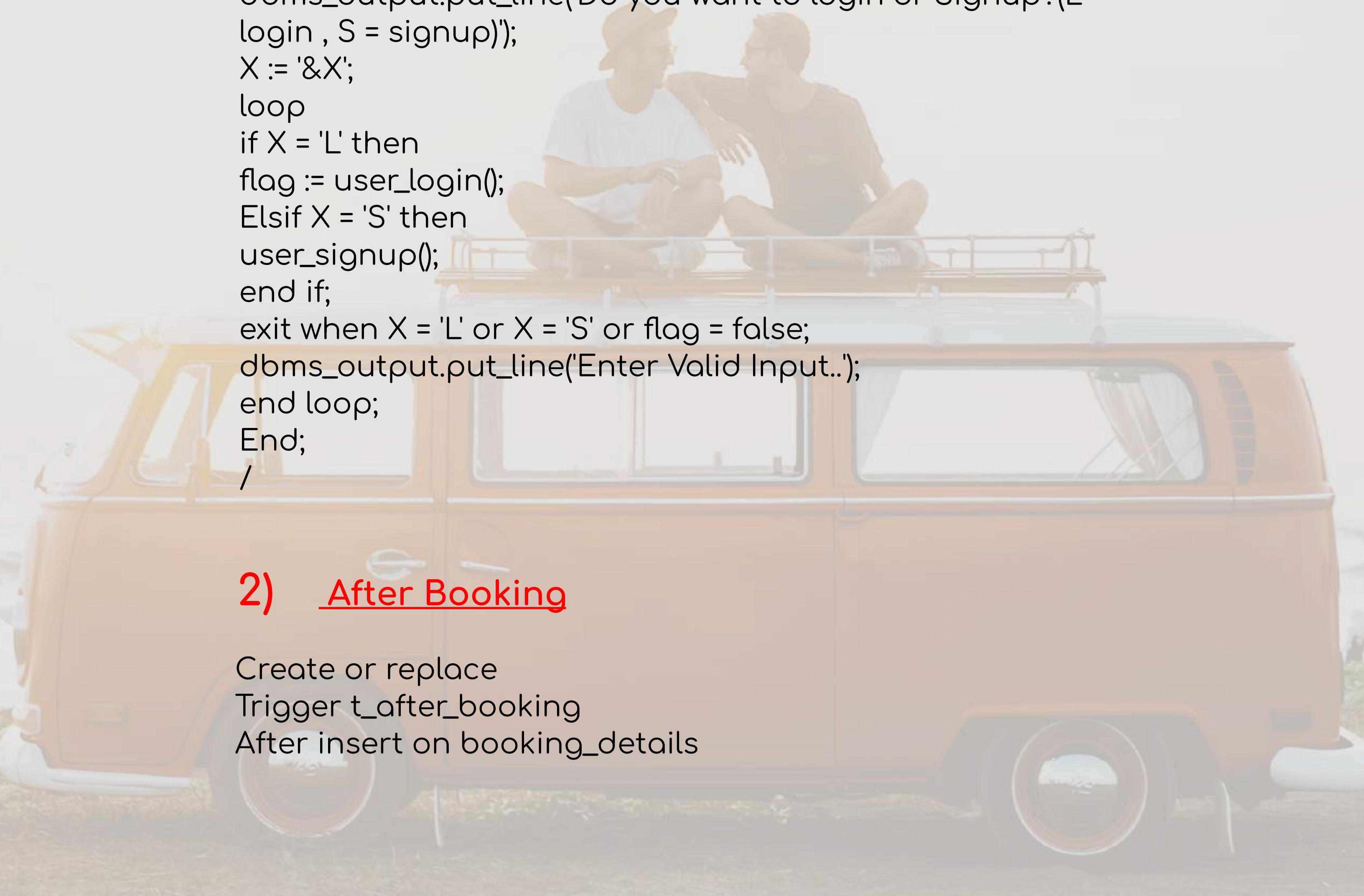
❖ TRIGGERS :

1) After Log On

```
 Create or replace
 Trigger t_after_logon
after logon
On schema
Declare
X varchar2(2);
flag boolean;
Begin
dbms_output.put_line('Do you want to login or Signup?(L =
login , S = signup)');
X := '&X';
loop
if X = 'L' then
flag := user_login();
Elsif X = 'S' then
user_signup();
end if;
exit when X = 'L' or X = 'S' or flag = false;
dbms_output.put_line('Enter Valid Input..');
end loop;
End;
/
```

2) After Booking

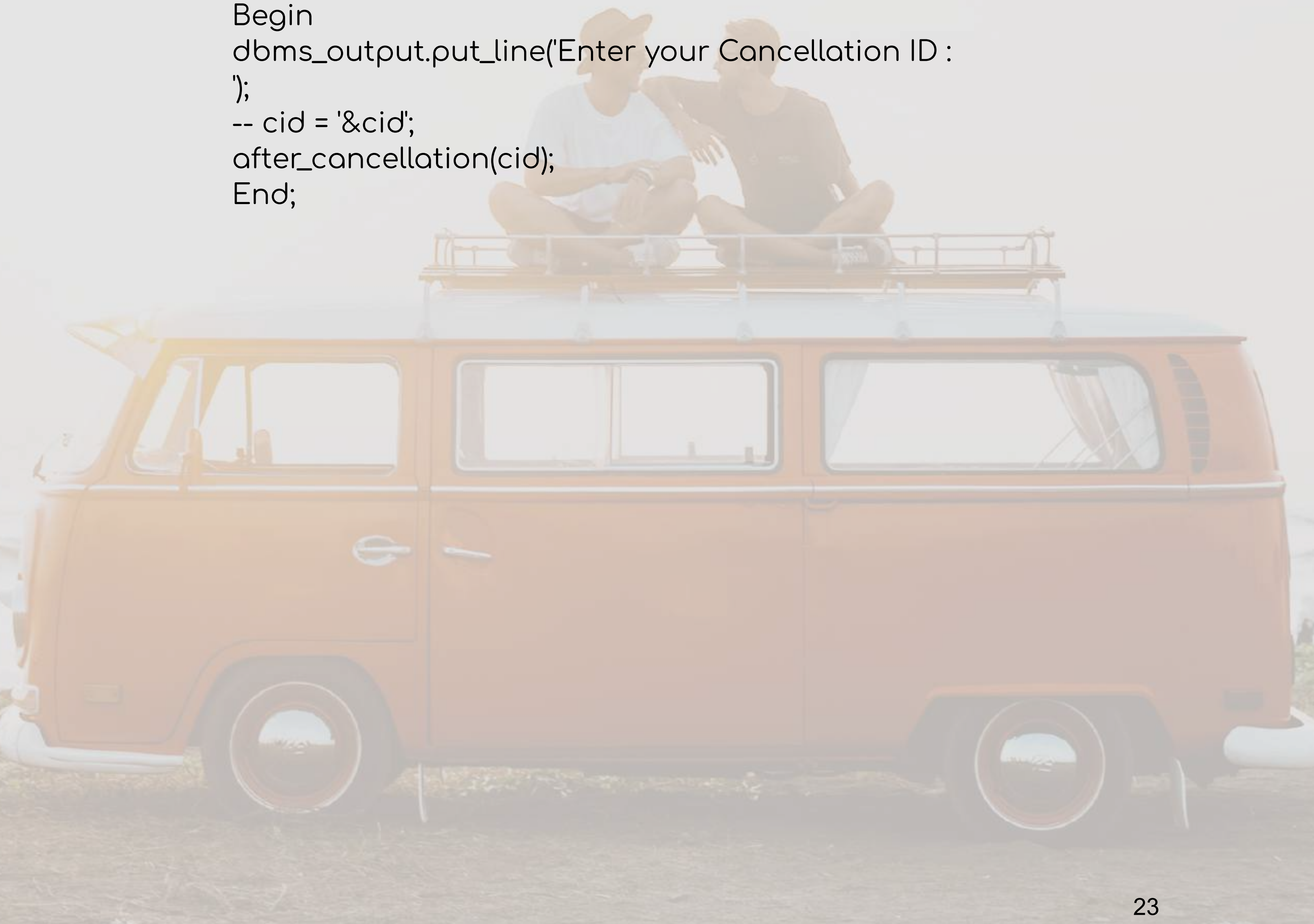
```
Create or replace
Trigger t_after_booking
After insert on booking_details
```



```
 declare
 bid booking_details.b_id%type;
 Begin
 dbms_output.put_line('Enter your Booking ID : ');
-- bid = '&bid';
after_booking(bid);
End;
 /
```

3) After Cancellation

```
Create or replace
Trigger t_after_cancellation
After insert on cancellation
declare
cid Cancellation.c_id%type;
Begin
dbms_output.put_line('Enter your Cancellation ID :
');
-- cid = '&cid';
after_cancellation(cid);
End;
```



Thank You :)

For giving us an opportunity for working on this project and hope to do further.

Submitted to: Mrs. Anjali Jivani
Mrs. Meghna Desai

College :- Faculty of Technology and Engineering,
Maharaja Sayajirao University of Baroda, Vadodara,
Gujarat.

Department :- Department of Computer Science and Engineering.

Year/Semester :- 2nd Year/ 4th Semester

Academic Session :- 2022- 2023