

## 1. LIST FEATURES IMPLEMENTED

All business requirements were successfully implemented as below:

### Requirement 1 a and b

Fully implemented using the following:

#### PROCEDURES

1. **Get\_customer\_details** Retrieves customer details given customer\_id,surname,email or phone.
2. **New\_customer** - Creates new customer

#### FUNCTIONS

1. **Check\_customer\_by\_id**- checks existence of a customer based on id provided.
2. **Check\_customer\_by\_surname**- checks existence of a customer based on surname.
3. **Check\_customer\_by\_email**- checks existence of a customer based on email.
4. **Check\_customer\_by\_phone**- checks existence of a customer based on phone number.

### Requirement 1 c and d

Fully implemented using the following:

#### PROCEDURES

1. **Add\_booking\_by\_id** Makes new booking based on customer\_id entered.
2. **Add\_booking\_by\_surname** Makes new booking based on customer surname if id is forgotten.
3. **Add\_booking\_by\_email** Makes new booking based on customer\_email if id is forgotten.
4. **Add\_booking\_by\_phone** Makes new booking based on customer\_phone number if id is forgotten.
5. **Check\_start\_date** Checks if startdate is equal to current date.

### Requirement 2

Fully implemented using the following:

#### PROCEDURES

1. **Cancel\_booking** Cancels open/confirmed bookings

### Requirement 3

Fully implemented using the following:

#### PROCEDURES

1. **Add\_new\_vehicle** Creates a new vehicle into the database by adding vehicle details in vehicle, manufacturer and model tables.

### Requirement 4

Fully implemented using the following:

#### TRIGGER

1. **Secure\_booking\_statuscode** Trigger enforces business rule only open/confirmed bookings can be cancelled .

### Requirement 5

Fully implemented using the following:

#### FUNCTIONS

1. **daily\_bookings\_report**- prints report on bookings as per date provided.

#### VIEWS

1. **Available\_vehicles view**-Selects list of vehicles from vehicles and booking tables.
2. **Booking\_view**- shows booking list- used as report source

### Requirement 6

Fully implemented. Appropriate error handling incorporated in the all the code.

## 2. IMPLEMENTATION

### a. Overall Design

CAR HIRE Flowchart

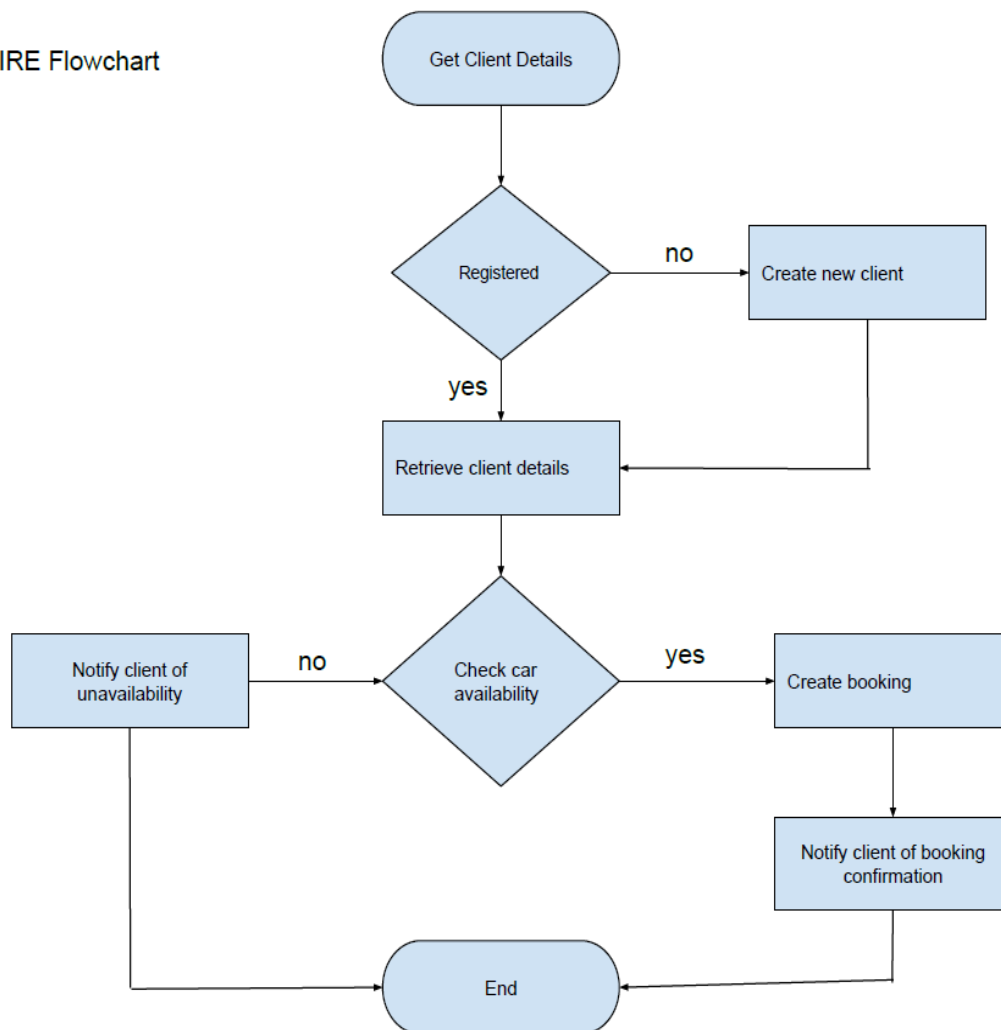


Figure 1

## Structure

The implantation structure used was to have one CAR\_HIRE package that holds procedures and functions required to run the application. Views were used to consolidate data for report generation and triggers to compliment functionality such as enforce business rule of cancelling only open/confirmed bookings and inserting a record when it does not exist in customers table.

### Package structure

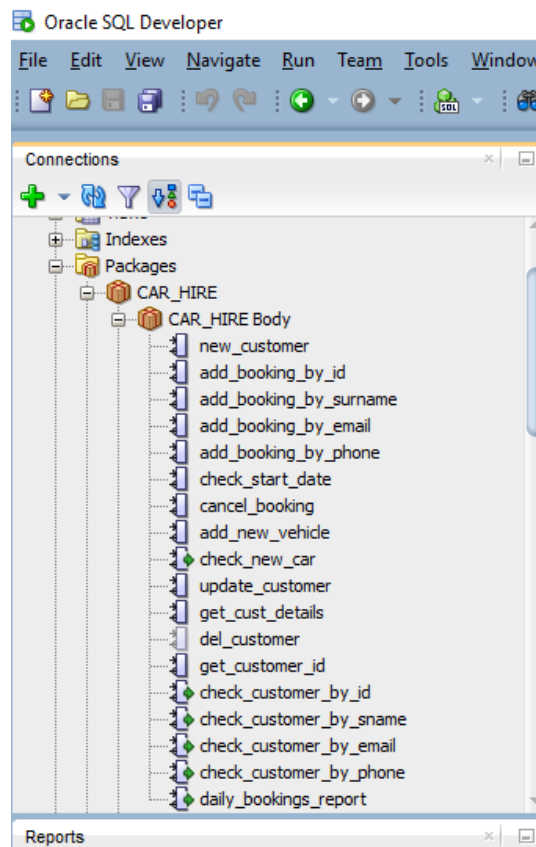


Figure 2

### Views

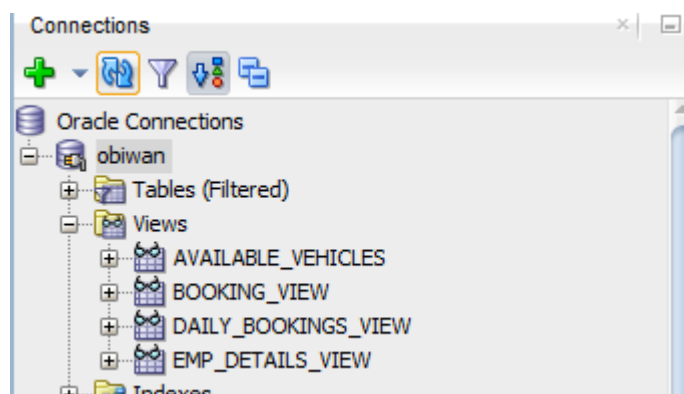


Figure 3

## Triggers

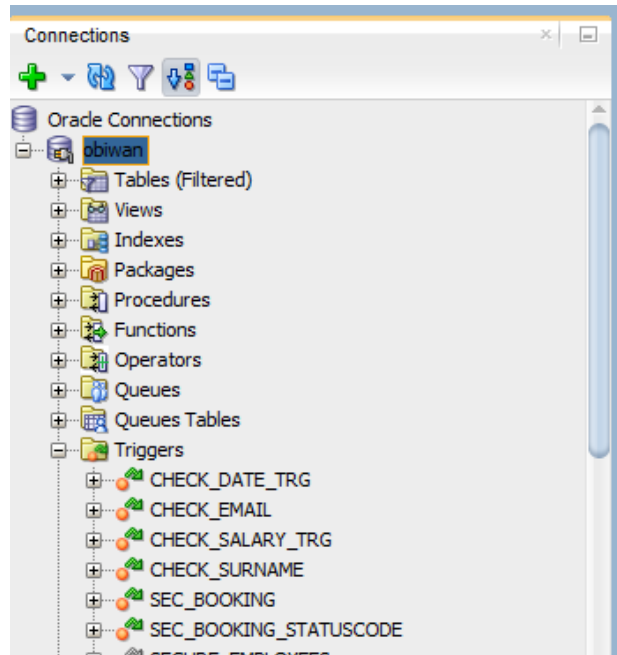


Figure 4

### b. Detailed Development

1. The system should be able to recognise if the customer is new or existing based on the information provided by the customer.
  - a. If a new customer is submitting a car hire request, they must be registered by the system as a part of the booking process.

checking for existing customer

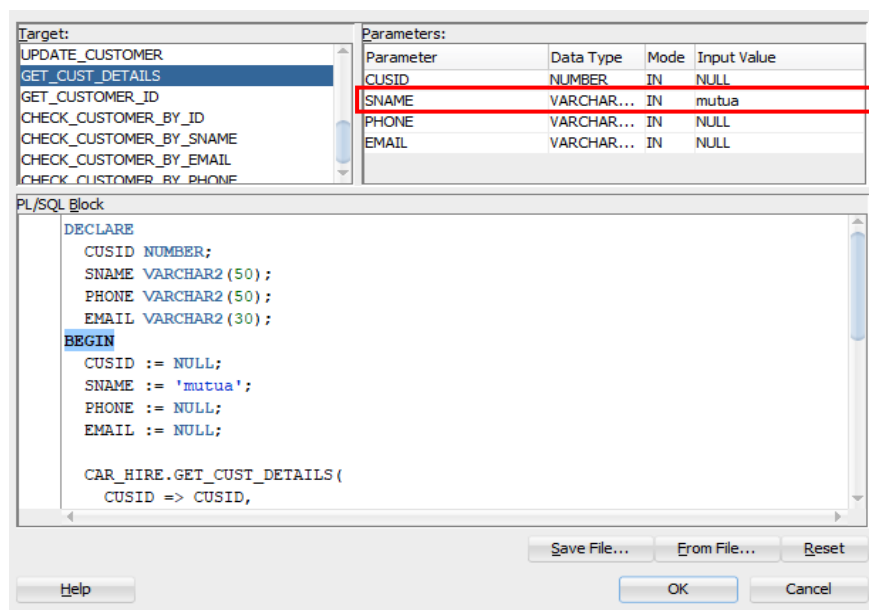


Figure 5

results

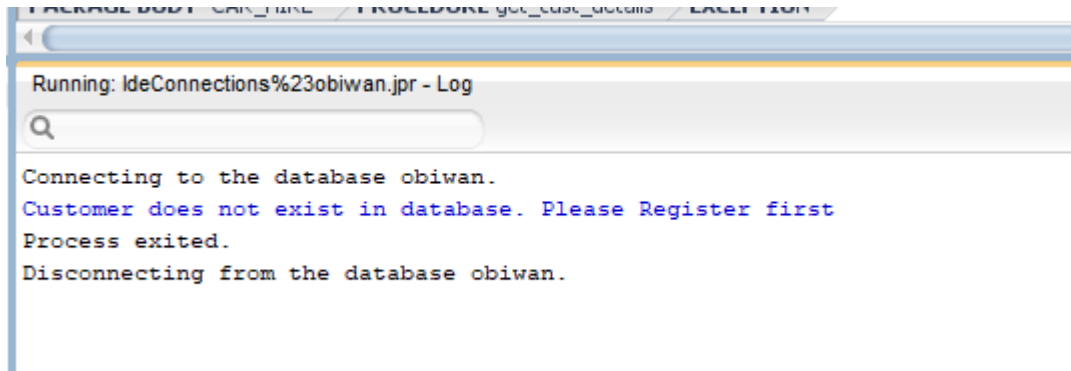


Figure 6

b. If an existing customer is submitting a car hire request their details should be taken from the customer table.

Checking for existing customer

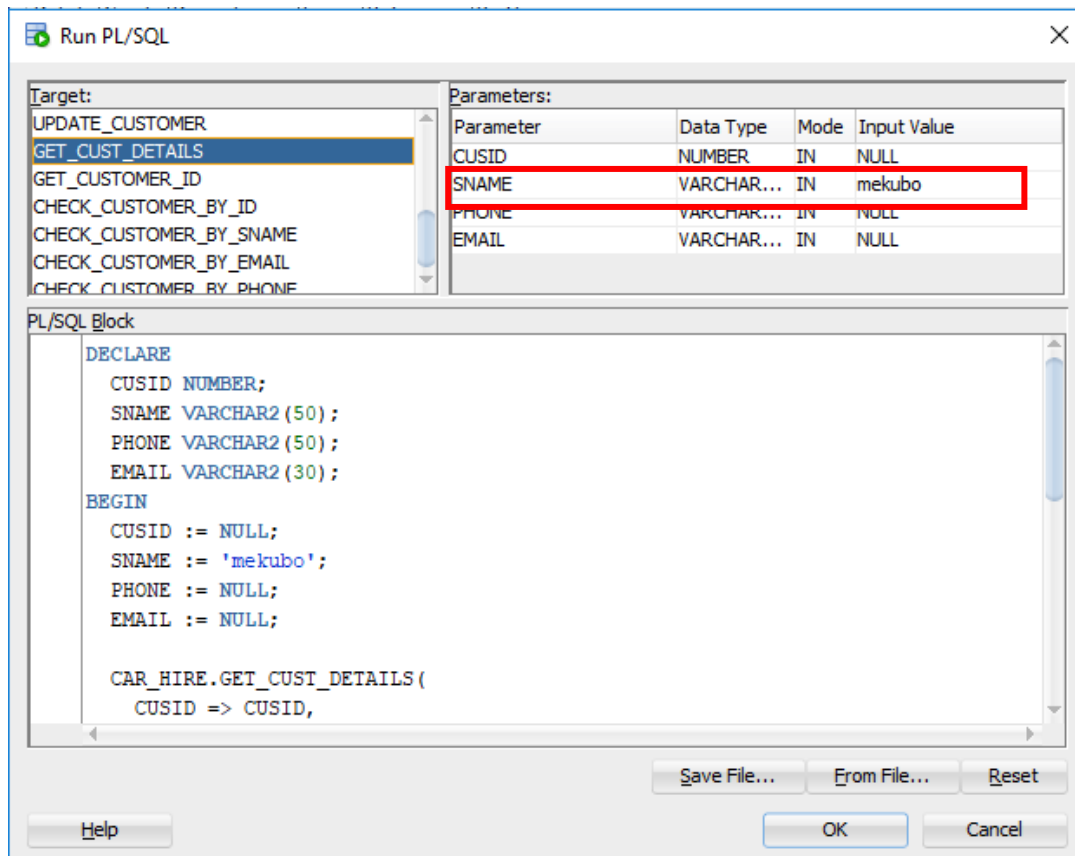


Figure 7

results

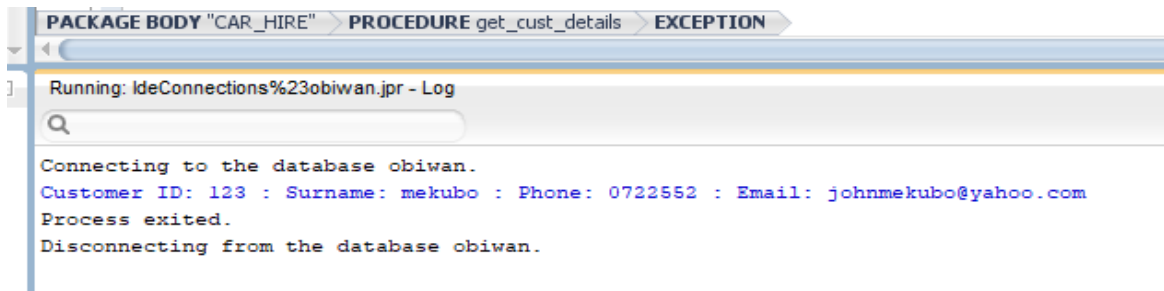


Figure 8

The process should use the existing sequence `book_id_seq` to generate a new booking id automatically. After the booking has been made it should also print on the screen an appropriate confirmation message including the customer id, the booking id as well as car hiring details (simulation of sending booking confirmation to the customer via email).

On receiving a car hire enquiry, the company's process is required to check the availability of requested cars and vans for dates specified:

c. If there are vehicles of a requested category available, the customer's details are recorded (if not stored already) and a new booking is made and one of the available cars (using registration number) is assigned to the booking. An appropriate confirmation message is displayed on the screen. Customers are not required to pay at this stage and will pay for the vehicle at the time of pick up.

### Booking by customer id

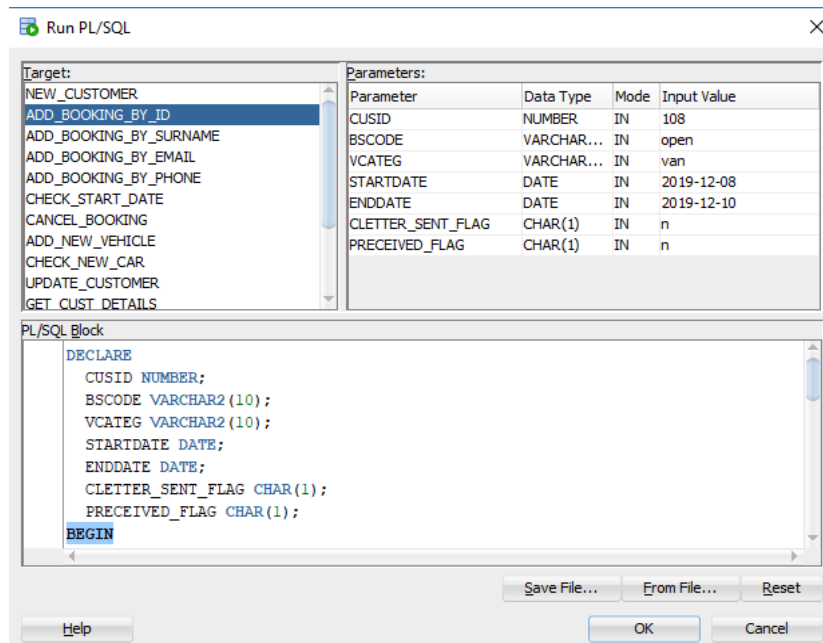


Figure 9

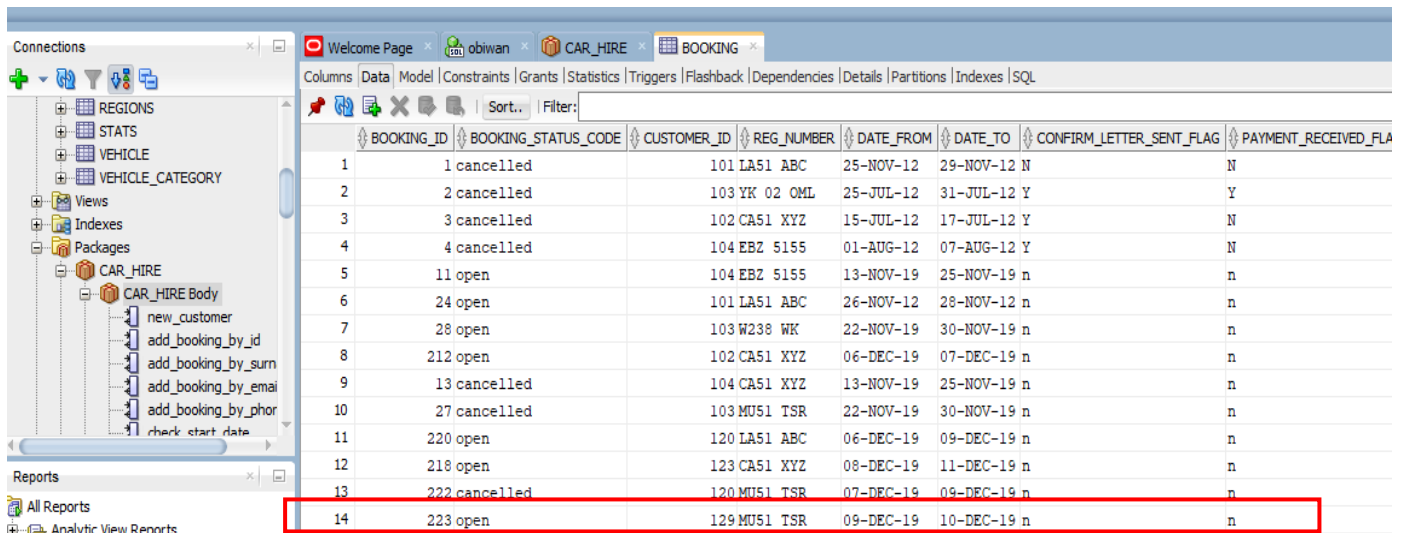
## Results

```
Car Registration Number W238 WK is available
Please note that booking is made 1 day from booking date(Today)
Car Registration Number W238 WK has been booked from 09-DEC-19 To: 10-DEC-19
```

Figure 10

3. A customer should be able to cancel an existing booking. Cancelling the highlighted record

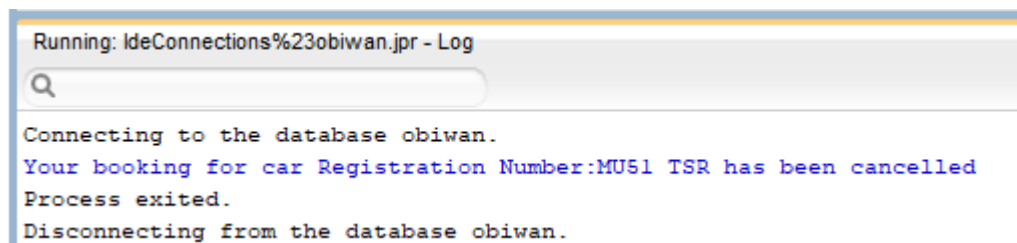
## Record before cancellation



BOOKING_ID	BOOKING_STATUS_CODE	CUSTOMER_ID	REG_NUMBER	DATE_FROM	DATE_TO	CONFIRM_LETTER_SENT_FLAG	PAYMENT_RECEIVED_FLAG
1	1 cancelled	101 LA51	ABC	25-NOV-12	29-NOV-12	N	N
2	2 cancelled	103 YK 02	OML	25-JUL-12	31-JUL-12	Y	Y
3	3 cancelled	102 CA51	XYZ	15-JUL-12	17-JUL-12	Y	N
4	4 cancelled	104 EBZ	5155	01-AUG-12	07-AUG-12	Y	N
5	11 open	104 EBZ	5155	13-NOV-19	25-NOV-19	n	n
6	24 open	101 LA51	ABC	26-NOV-12	28-NOV-12	n	n
7	28 open	103 W238	WK	22-NOV-19	30-NOV-19	n	n
8	212 open	102 CA51	XYZ	06-DEC-19	07-DEC-19	n	n
9	13 cancelled	104 CA51	XYZ	13-NOV-19	25-NOV-19	n	n
10	27 cancelled	103 MU51	TSR	22-NOV-19	30-NOV-19	n	n
11	220 open	120 LA51	ABC	06-DEC-19	09-DEC-19	n	n
12	218 open	123 CA51	XYZ	08-DEC-19	11-DEC-19	n	n
13	222 cancelled	120 MU51	TSR	07-DEC-19	09-DEC-19	n	n
14	223 open	129 MU51	TSR	09-DEC-19	10-DEC-19	n	n

Figure 11

## results



```
Running: IdeConnections%23obiwan.jpr - Log

Connecting to the database obiwan.
Your booking for car Registration Number:MU51 TSR has been cancelled
Process exited.
Disconnecting from the database obiwan.
```

Figure 12



Record after cancellation

BOOKING_ID	BOOKING_STATUS_CODE	CUSTOMER_ID	REG_NUMBER	DATE_FROM	DATE_TO	CONFIRM_LETTER_SENT_FLAG	P
1	1 cancelled	101	LA51 ABC	25-NOV-12	29-NOV-12	N	N
2	2 cancelled	103	YK 02 OML	25-JUL-12	31-JUL-12	Y	Y
3	3 cancelled	102	CA51 XYZ	15-JUL-12	17-JUL-12	Y	N
4	4 cancelled	104	EBZ 5155	01-AUG-12	07-AUG-12	Y	N
5	11 open	104	EBZ 5155	13-NOV-19	25-NOV-19	n	n
6	24 open	101	LA51 ABC	26-NOV-12	28-NOV-12	n	n
7	28 open	103	W238 WK	22-NOV-19	30-NOV-19	n	n
8	212 open	102	CA51 XYZ	06-DEC-19	07-DEC-19	n	n
9	13 cancelled	104	CA51 XYZ	13-NOV-19	25-NOV-19	n	n
10	27 cancelled	103	MU51 TSR	22-NOV-19	30-NOV-19	n	n
11	220 open	120	LA51 ABC	06-DEC-19	09-DEC-19	n	n
12	218 open	123	CA51 XYZ	08-DEC-19	11-DEC-19	n	n
13	222 cancelled	120	MU51 TSR	07-DEC-19	09-DEC-19	n	n
14	223 cancelled	129	MU51 TSR	09-DEC-19	10-DEC-19	n	n
15	224 open	108	W238 WK	09-DEC-19	10-DEC-19	n	n

Figure 13

3. The system should be able to record all necessary information about new cars added to the company fleet.

When a new vehicle is created, apart from the vehicle, model and manufacturer tables are updated with necessary details of the same vehicle.

capture vehicle details screen

Target:

- NEW\_CUSTOMER
- ADD\_BOOKING\_BY\_ID
- ADD\_BOOKING\_BY\_SURNAME
- ADD\_BOOKING\_BY\_EMAIL
- ADD\_BOOKING\_BY\_PHONE
- CHECK\_START\_DATE
- CANCEL\_BOOKING
- ADD\_NEW\_VEHICLE
- CHECK\_NEW\_CAR
- UPDATE\_CUSTOMER
- GET\_CUST\_DETAILS
- GET\_CUSTOMER\_ID
- CHECK\_CUSTOMER\_BY\_ID
- CHECK\_CUSTOMER\_BY\_SNAME

Parameters:

Parameter	Data Type	Mode	Input Value
REG	VARCHAR2(10)	IN	GB 789N
MANUF_CODE	VARCHAR2(3)	IN	BMW
MANUF_NAME	VARCHAR2(50)	IN	MBW German
MANUF_DET	VARCHAR2(2000)	IN	Luxury Vehicle Manufacturer
MODEL_CODE	VARCHAR2(5)	IN	5XL
MODEL_NAME	VARCHAR2(100)	IN	5 series
CAT_CODE	VARCHAR2(10)	IN	small
CURRMILE	NUMBER	IN	2500
DAILYRATE	NUMBER	IN	45
MOTDATE	DATE	IN	2019-12-31

PL/SQL Block

```
DECLARE
  REG VARCHAR2(10);
  MANUF_CODE VARCHAR2(3);
  MANUF_NAME VARCHAR2(50);
  MANUF_DET VARCHAR2(2000);
  MODEL_CODE VARCHAR2(5);
```

Buttons: Save File..., From File..., Reset, OK, Cancel, Help

Figure 14

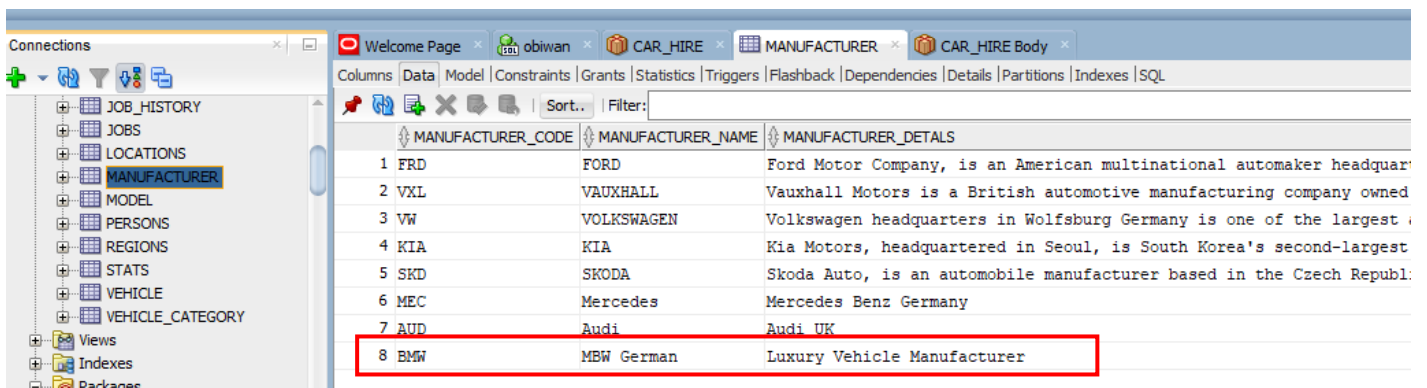
## Results

```
Running: IdeConnections%23obiwan.jpr - Log

Connecting to the database obiwan.
Vehicle Registration Number GB 789N details added into manufacturer table
Vehicle Registration Number GB 789N details inserted into model table
Vehicle Registration Number GB 789N details inserted into vehicle table
Process exited.
Disconnecting from the database obiwan.
```

Figure 15

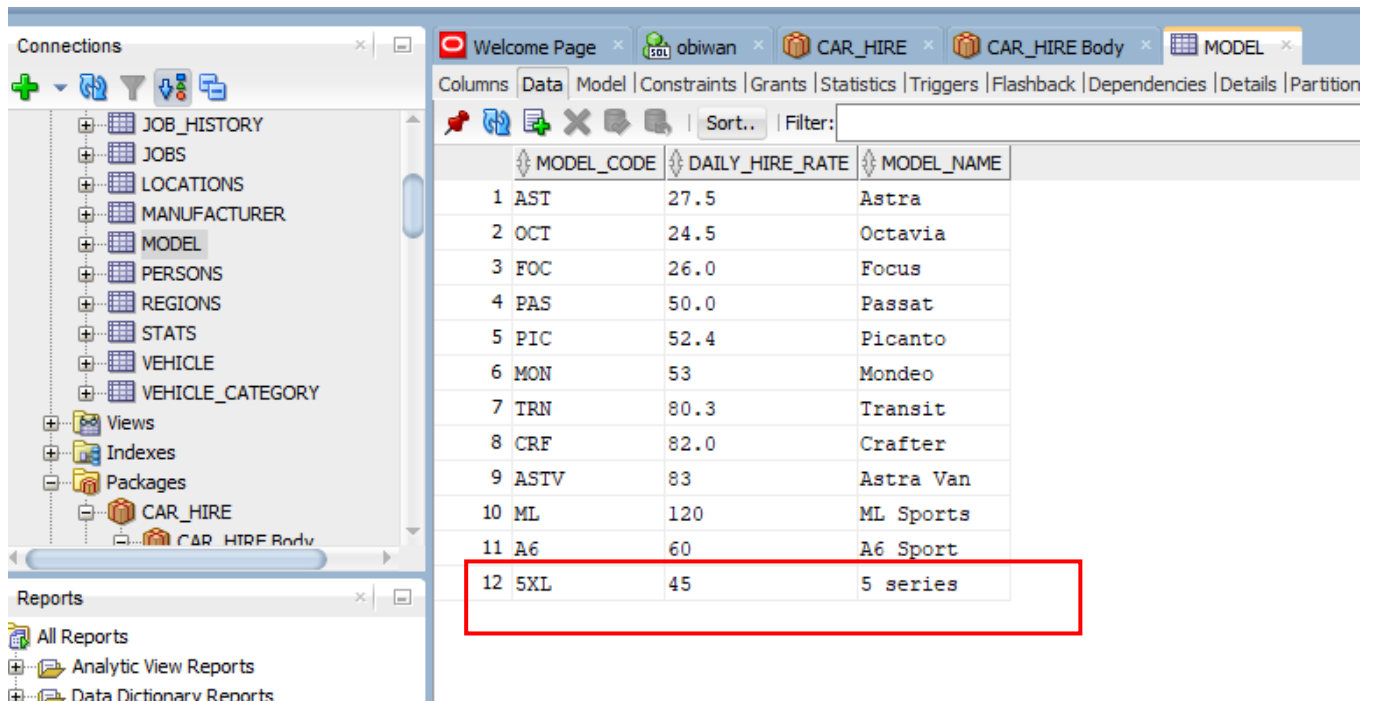
## Updated manufacturer table



MANUFACTURER_CODE	MANUFACTURER_NAME	MANUFACTURER_DETAILS
1 FRD	FORD	Ford Motor Company, is an American multinational automaker headquar
2 VXL	VAUXHALL	Vauxhall Motors is a British automotive manufacturing company owned
3 VW	VOLKSWAGEN	Volkswagen headquarters in Wolfsburg Germany is one of the largest
4 KIA	KIA	Kia Motors, headquartered in Seoul, is South Korea's second-largest
5 SKD	SKODA	Skoda Auto, is an automobile manufacturer based in the Czech Republ
6 MEC	Mercedes	Mercedes Benz Germany
7 AUD	Audi	Audi UK
8 BMW	MBW German	Luxury Vehicle Manufacturer

Figure 16

## Updated model table



MODEL_CODE	DAILY_HIRE_RATE	MODEL_NAME
1 AST	27.5	Astra
2 OCT	24.5	Octavia
3 FOC	26.0	Focus
4 PAS	50.0	Passat
5 PIC	52.4	Picanto
6 MON	53	Mondeo
7 TRN	80.3	Transit
8 CRF	82.0	Crafter
9 ASTV	83	Astra Van
10 ML	120	ML Sports
11 A6	60	A6 Sport
12 5XL	45	5 series

Figure 17

4. The company has a business rule: ‘If the status of the booking is ‘open’, or ‘confirmed’ it can be changed, but the bookings with the ‘cancelled’ cannot be changed’. You must design and implement the best way to enforce this business rule.

This section was implemented using a trigger to check status of sec\_booking\_statuscode column in booking table. If the status is “open” or “confirmed”, the record may be edited otherwise record is locked for edit status is” cancelled”.

#### sec\_booking\_statuscode trigger

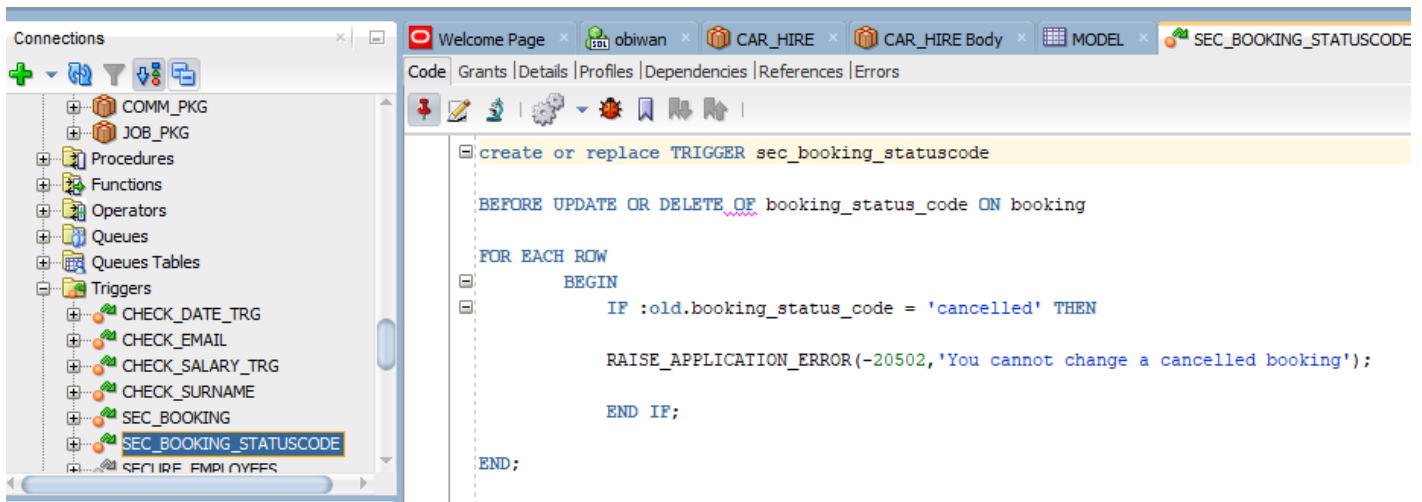


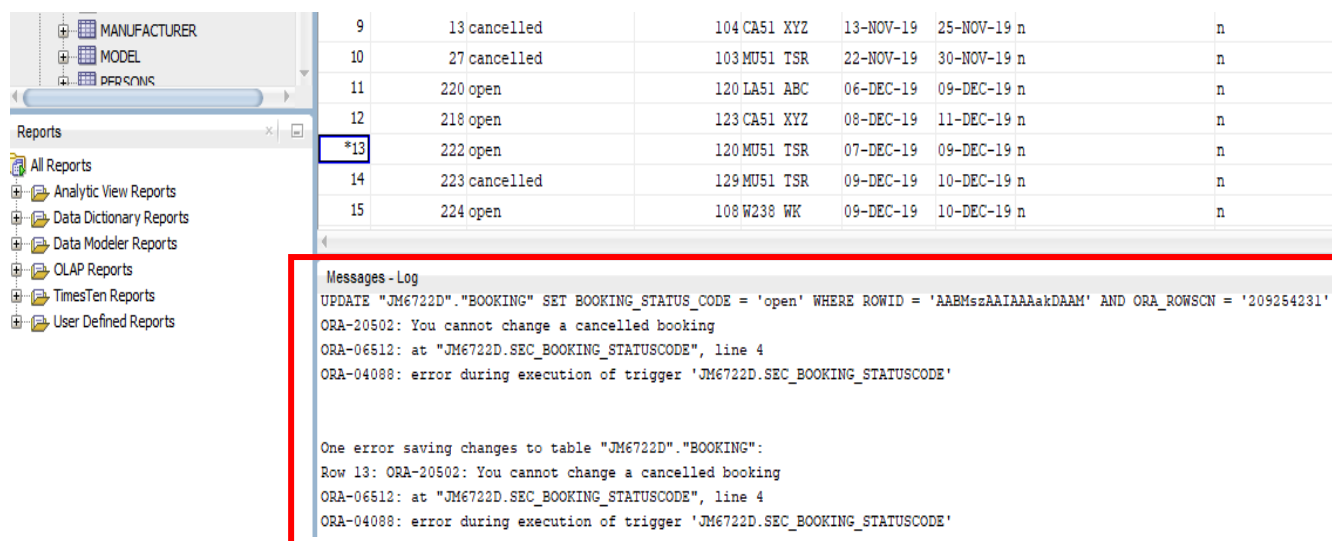
Figure 18

changing highlighted record status to open

BOOKING_ID	BOOKING_STATUS_CODE	CUSTOMER_ID	REG_NUMBER	DATE_FROM	DATE_TO	CONFIRM_LETTER_SENT_FLAG	PAYMENT_
1	cancelled	101 LA51	ABC	25-NOV-12	29-NOV-12	N	N
2	cancelled	103 YK 02	OML	25-JUL-12	31-JUL-12	Y	Y
3	cancelled	102 CA51	XYZ	15-JUL-12	17-JUL-12	Y	N
4	cancelled	104 EBZ 5155		01-AUG-12	07-AUG-12	Y	N
5	open	104 EBZ 5155		13-NOV-19	25-NOV-19	n	n
6	open	101 LA51	ABC	26-NOV-12	28-NOV-12	n	n
7	open	103 W238	WK	22-NOV-19	30-NOV-19	n	n
8	open	102 CA51	XYZ	06-DEC-19	07-DEC-19	n	n
9	cancelled	104 CA51	XYZ	13-NOV-19	25-NOV-19	n	n
10	cancelled	103 MU51	TSR	22-NOV-19	30-NOV-19	n	n
11	open	120 LA51	ABC	06-DEC-19	09-DEC-19	n	n
12	open	123 CA51	XYZ	08-DEC-19	11-DEC-19	n	n
13	cancelled	120 MU51	TSR	07-DEC-19	09-DEC-19	n	n
14	cancelled	129 MU51	TSR	09-DEC-19	10-DEC-19	n	n
15	open	108 W238	WK	09-DEC-19	10-DEC-19	n	n

Figure 19

## Error thrown after trying to edit record



The screenshot shows a database report interface. On the left, there is a sidebar with a tree view containing 'MANUFACTURER', 'MODEL', and 'PERSONS'. Below this is a 'Reports' section with a list of report types: 'All Reports', 'Analytic View Reports', 'Data Dictionary Reports', 'Data Modeler Reports', 'OLAP Reports', 'TimesTen Reports', and 'User Defined Reports'. The main area displays a table with columns for row number, status, and various booking details. Row 13 is highlighted with a blue selection bar. Below the table, a 'Messages - Log' window is open, showing the following error messages:

```

UPDATE "JM6722D"."BOOKING" SET BOOKING_STATUS_CODE = 'open' WHERE ROWID = 'AABMszAAIAAAakDAAM' AND ORA_ROWSCN = '209254231'
ORA-20502: You cannot change a cancelled booking
ORA-06512: at "JM6722D.SEC_BOOKING_STATUSCODE", line 4
ORA-04088: error during execution of trigger 'JM6722D.SEC_BOOKING_STATUSCODE'

One error saving changes to table "JM6722D"."BOOKING":
Row 13: ORA-20502: You cannot change a cancelled booking
ORA-06512: at "JM6722D.SEC_BOOKING_STATUSCODE", line 4
ORA-04088: error during execution of trigger 'JM6722D.SEC_BOOKING_STATUSCODE'
  
```

Figure 20

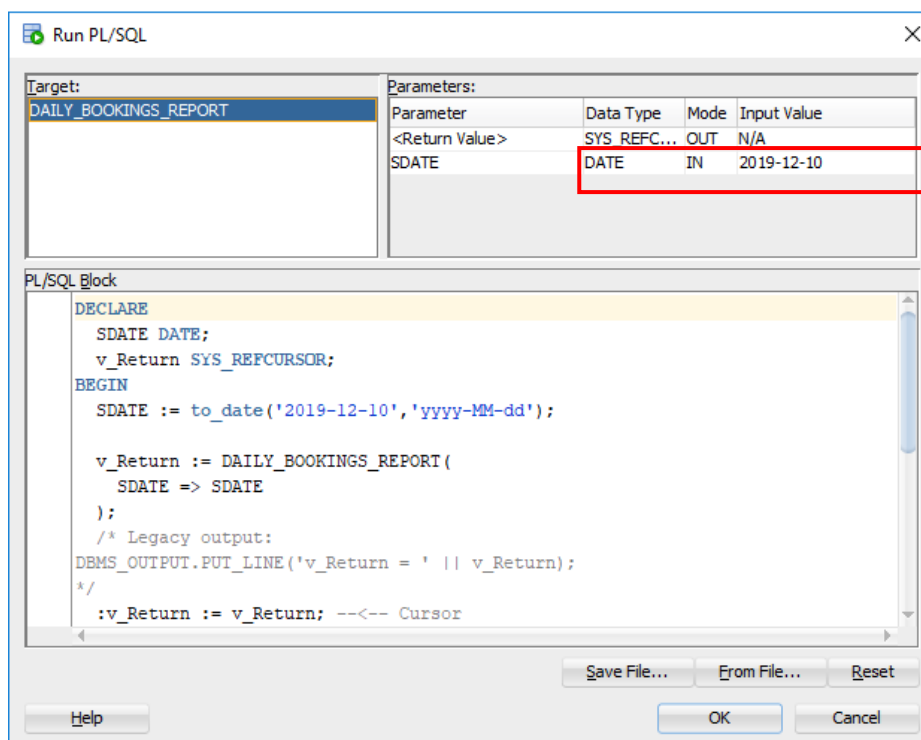
5. A company's clerk should be able to print a report showing all bookings for any day.

Sample report generated.

This requirement was implemented using function **daily\_bookings\_report**. When executed, it displays a prompt for user to enter date and on clicking apply the report is displayed.

business rule.

## Enter date prompt



The screenshot shows the 'Run PL/SQL' dialog box. The 'Target' field is set to 'DAILY\_BOOKINGS\_REPORT'. The 'Parameters' section shows a table with columns: Parameter, Data Type, Mode, and Input Value. The table contains two rows: '<Return Value>' with Data Type 'SYS\_REFC...', Mode 'OUT', and Input Value 'N/A'; and 'SDATE' with Data Type 'DATE', Mode 'IN', and Input Value '2019-12-10'. The 'SDATE' row is highlighted with a red selection bar. Below the parameters section, the 'PL/SQL Block' contains the following code:

```

DECLARE
  SDATE DATE;
  v_Return SYS_REFCURSOR;
BEGIN
  SDATE := to_date('2019-12-10', 'yyyy-MM-dd');

  v_Return := DAILY_BOOKINGS_REPORT (
    SDATE => SDATE
  );
  /* Legacy output:
  DBMS_OUTPUT.PUT_LINE('v_Return = ' || v_Return);
  */
  :v_Return := v_Return; --<-- Cursor
  
```

At the bottom of the dialog, there are buttons for 'Save File...', 'From File...', 'Reset', 'Help', 'OK', and 'Cancel'.

Figure 21

**daily\_bookings\_report-** Note that you have to click on output variables tab to see the report.

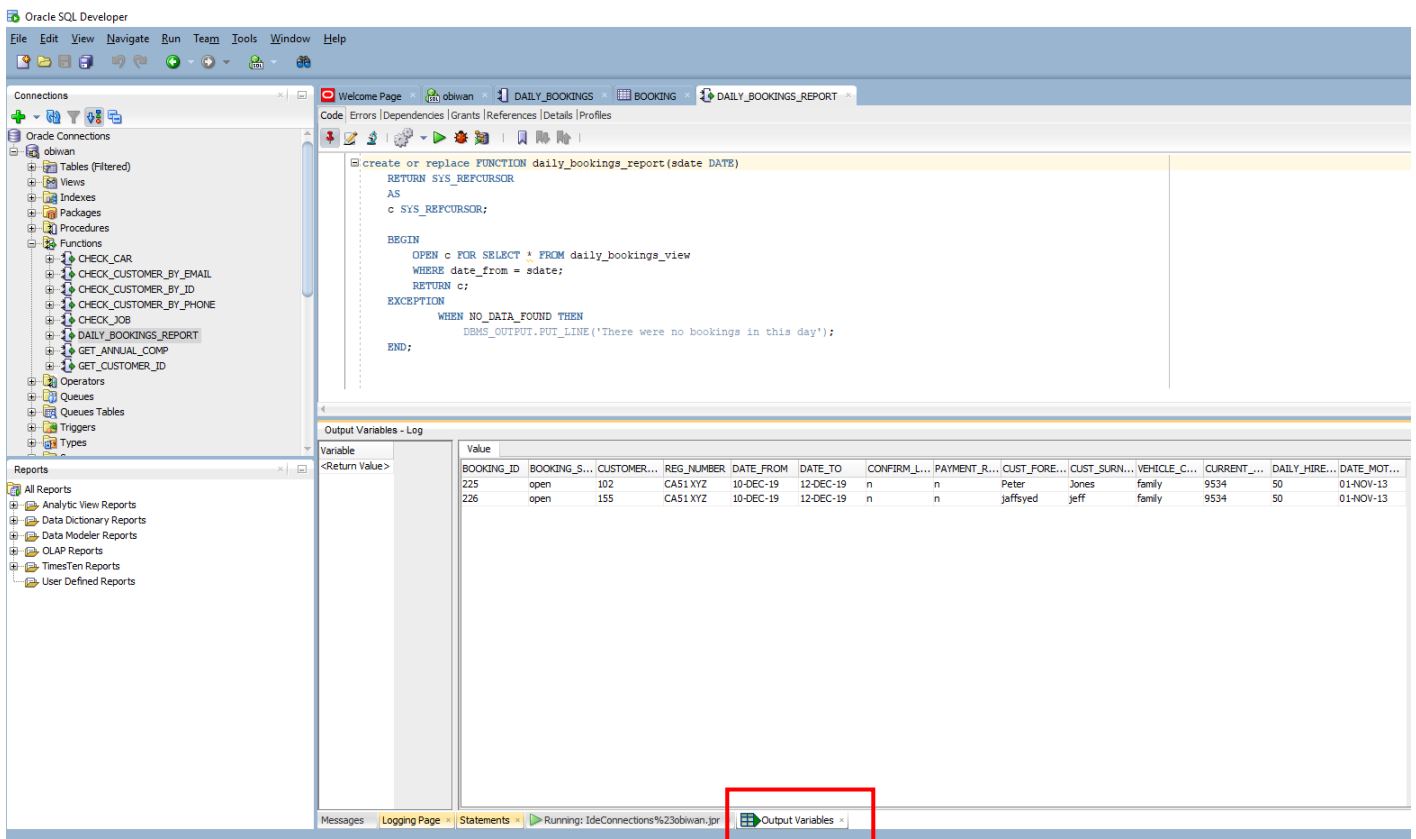


Figure 22

## 6. All your PL/SQL subprograms should include appropriate error handling.

Appropriate error handling incorporated in the code to capture exceptions that may occur.

### Error Handling

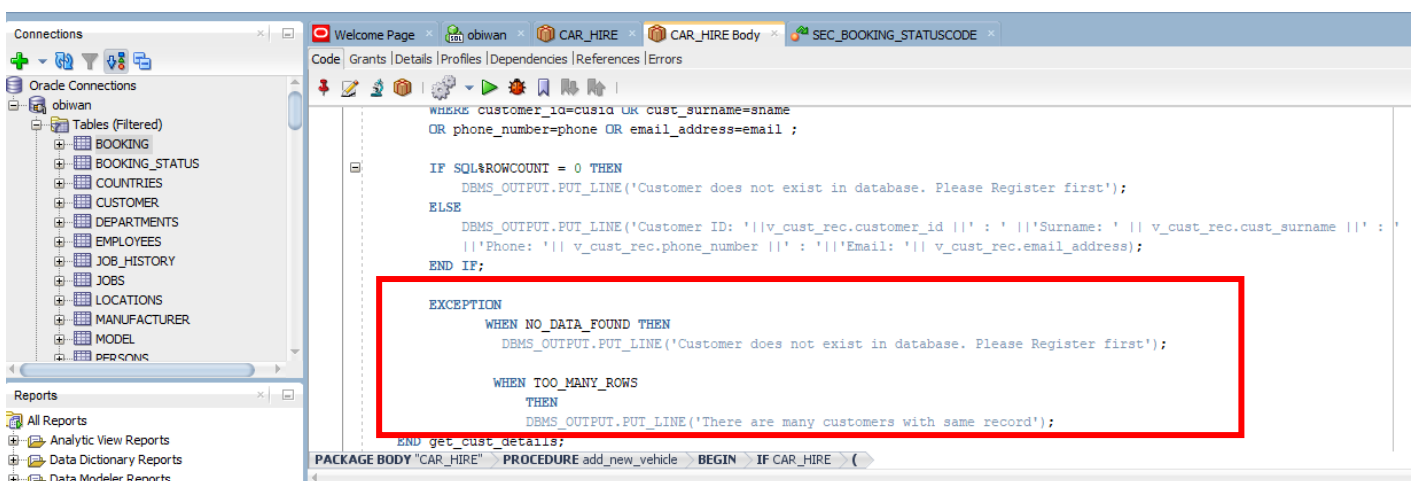


Figure 23