VILLA RENTAL SYSTEM

Database Application Development [DBT2113C]

UNIVERSITY of TECHNOLOGY, MAURITIUS

Name: BEEHARRY Mohammad Taajuddin (2003_19375)

Cohort: BSE20AFT

Year 2 Semester 1

BSC (HONS) SOFTWARE ENGINEERING

Lecturer: Mr. Hansraj Seegobin

Table of Contents

Job Description

An owner of a villa rental assists individuals who need to rent a villa for a short amount of time. Different types of villa are available in the country. For example: a luxurious villa, a villa with one bed and a garden, etc. So, a database system is needed to keep track of all information.
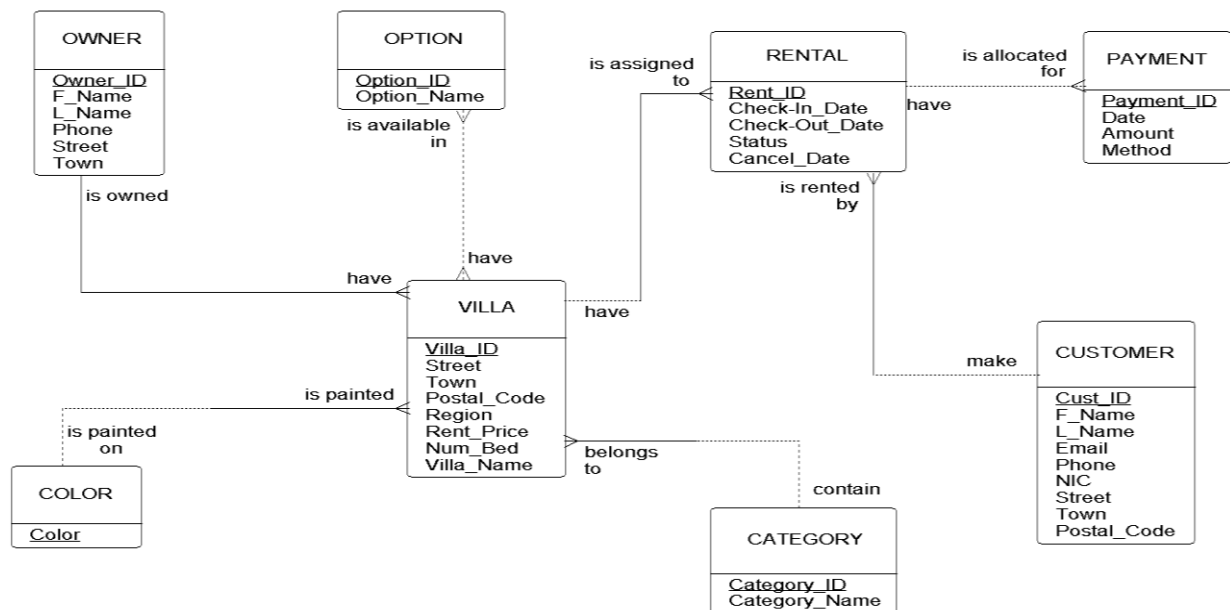
Business Rules

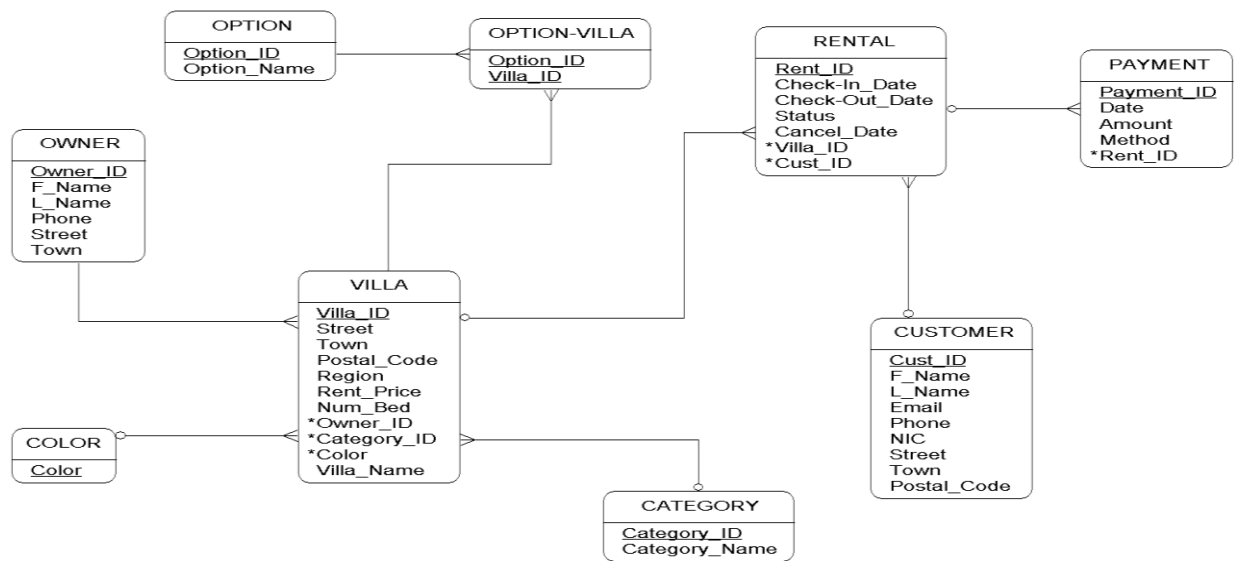**Below are the business rules for a Villa Rental:**

1. A customer may make one or more rentals. The identifier for customer is **Cust_ID** and other attributes **F_Name**, **L_Name**, **Email**, **Phone**, **NIC**, **Street**, **Town** and **Postal_Code**. A rental must be made by only one customer. The identifier for rental is **Rent_ID** with the other attributes **Check_In_Date**, **Check_Out_Date**, **Status** and **Cancel_Date**.

2. A payment is allocated for one rental. The identifier for entity payment is **Payment_ID**. Other attributes are **Date**, **Amount** and **Method.** A rental may have one or more than one payments.

3. A rental must assign to a villa. The identifier for villa is **villa_ID** and other attributes are **Street**, **Town**, **Postal_Code**, **Region**, **Rent_Price**, **Num_Bed** and **Villa_Name**. A villa may have one or more rentals.

4. A villa must be owned by only one owner. The identifier for owner is **Owner_ID** along with the other attributes **F_Name**, **L_Name**, **Phone**, **Street** and **Town**. An owner can have one or more than one villas.

5. A villa may have one or more options. The identifier for option is **Option_ID** and the other attribute is **Option_Name**. Moreover, an option may also available on many villas.

6. A villa must belong to one and only category and each category may contain one or more villas. The identifier for category is **Category_ID** and other attribute is **Category_Name**.

7. Each villa must be painted with only one color. The identifier for color is **Color** and a color may be painted on one or more villas.

Logical Data Model

Physical Data Model

Data Dictionary

TABLE NAME	COLUMN NAME	DATA TYPE	FIELD SIZE	CONSTRAINT	VALUES	CONSTRAINT NAME
OWNER	OWNER_ID	NUMBER	6	PK		OWNER_OWNER_ID_PK
	F_NAME	VARCHAR2	15			
	L_NAME	VARCHAR2	15			
	PHONE	NUMBER	15			
	STREET	VARCHAR2	50			
	TOWN	VARCHAR2	50			
CUSTOMER	CUST_ID	NUMBER	6	PK		CUSTOMER_CUST_ID_PK
	F_NAME	VARCHAR2	15			
	L_NAME	VARCHAR2	15			
	EMAIL	VARCHAR2	50	UK		CUSTOMER_EMAIL_UK
	PHONE	NUMBER	15			
	NIC	VARCHAR2	17			
	STREET	VARCHAR2	50			
	TOWN	VARCHAR2	50			
	POSTAL_CODE	NUMBER	5			
CATEGORY	CATEGORY_ID	NUMBER	6	PK		CATEGORY_CATEGORY_ID_PK
	CATEGORY_NAME	VARCHAR2	50	UK		CATEGORY_CATEGORY_NAME_UK
COLOR	COLOR	VARCHAR2	15	PK		COLOR_COLOR_PK
VILLA	VILLA_ID	NUMBER	6	PK		VILLA_VILLA_ID_PK
	VILLA_NAME	VARCHAR2	50			
	STREET	VARCHAR2	50			
	TOWN	VARCHAR2	50			
	POSTAL_CODE	NUMBER	5			
	REGION	VARCHAR2	15			
	RENT_PRICE	NUMBER	(8,2)		>0	

Table	Column	Data Type	Size	Constraint	References	Constraint Name
	POSTAL_CODE	NUMBER	5			
	REGION	VARCHAR2	15			
	RENT_PRICE	NUMBER	(8,2)		>0	
	NUM_BED	NUMBER	2			
	OWNER_ID	NUMBER	6	FK	REFERENCES OWNER(OWNER_ID)	VILLA_OWNER_ID_FK
	COLOR	VARCHAR2	15	FK	REFERENCES COLOR(COLOR)	VILLA_COLOR_FK
	CATEGORY_ID	NUMBER	6	FK	REFERENCES CATEGORY(CATEGORY_ID)	VILLA_CATEGORY_ID_FK
OPTIONS	OPTION_ID	NUMBER	6	PK		OPTIONS_OPTION_ID_PK
	OPTION_NAME	VARCHAR2	50	UK		OPTIONS_OPTION_NAME_UK
OPTION_VILLA	OPTION_ID	NUMBER	6	COMPOSITE PK & FK	REFERENCES OPTIONS(OPTION_ID)	
	VILLA_ID	NUMBER	6	COMPOSITE PK & FK	REFERENCES VILLA(VILLA_ID)	
RENTAL	RENT_ID	NUMBER	6	PK		RENTAL_RENT_ID_PK
	CHECK_IN_DATE	DATE				
	CHECK_OUT_DATE	DATE				
	STATUS	VARCHAR2	10			
	CANCEL_DATE	DATE				
	VILLA_ID	NUMBER	6	FK	REFERENCES VILLA(VILLA_ID)	RENTAL_VILLA_ID_FK
	CUST_ID	NUMBER	6	FK	REFERENCES CUSTOMER(CUST_ID)	RENTAL_CUST_ID_FK
PAYMENT	PAYMENT_ID	NUMBER	6	PK		PAYMENT_PAYMENT_ID_PK
	PAYMENT_DATE	DATE				
	AMOUNT	NUMBER	(8,2)		>0	
	PAYMENT_METHOD	CHAR	4			
	RENT_ID	NUMBER	6	FK	REFERENCES RENTAL(RENT_ID)	PAYMENT_RENT_ID_FK

Forms

Owner Form

CREATE BUTTON

```sql
SELECT OWNER_ID_SEQ.NEXTVAL
INTO :OWNER_BLOCK.OWNER_ID
FROM DUAL;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST OWNER BUTTON

```sql
GO_ITEM('OWNER_BLOCK.OWNER_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

SAVE BUTTON

```sql
INSERT INTO OWNER(OWNER_ID, F_NAME, L_NAME, PHONE, STREET, TOWN)
VALUES(:OWNER_BLOCK.OWNER_ID, :OWNER_BLOCK.F_NAME, :OWNER_BLOCK.L_NAME, :OWNER_BLOCK.PHONE, :OWNER_BLOCK.STREET, :OWNER_BLOCK.TOWN);
COMMIT;
MESSAGE('RECORD IS SAVED SUCCESSFULLY');

GO_BLOCK('OWNER_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_OWNER_FORM;

SET_ITEM_PROPERTY('LIST_OWNER_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```sql
UPDATE OWNER
SET F_NAME = :OWNER_BLOCK.F_NAME,
L_NAME = :OWNER_BLOCK.L_NAME,
PHONE = :OWNER_BLOCK.PHONE,
STREET = :OWNER_BLOCK.STREET,
TOWN = :OWNER_BLOCK.TOWN
WHERE OWNER_ID = :OWNER_BLOCK.OWNER_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('OWNER_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_OWNER_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_OWNER_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_OWNER_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_OWNER_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR OWNER FORM

```
PROCEDURE CLEAR_OWNER_FORM IS
BEGIN
  :OWNER_BLOCK.OWNER_ID:='';
  :OWNER_BLOCK.F_NAME:='';
  :OWNER_BLOCK.L_NAME:='';
  :OWNER_BLOCK.PHONE:='';
  :OWNER_BLOCK.STREET:='';
  :OWNER_BLOCK.TOWN:='';
END;
```

UPDATE ALERT

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

TRIGGER

WHEN-WINDOW-ACTIVATED

```
GO_ITEM('NEW_BUTTON');

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);

GO_BLOCK('OWNER_VIEW_BLOCK');
EXECUTE_QUERY;
```

Customer Form

The picture below refers when the form is being filled.

CREATE BUTTON

```
SELECT CUST_ID_SEQ.NEXTVAL
INTO :CUSTOMER_BLOCK.CUST_ID
FROM DUAL;


SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST CUSTOMER BUTTON

```
GO_ITEM('CUSTOMER_BLOCK.CUST_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

SAVE BUTTON

```
INSERT INTO CUSTOMER(CUST_ID,F_NAME,L_NAME,EMAIL,PHONE,NIC,STREET,TOWN,POSTAL_CODE)
VALUES(:CUSTOMER_BLOCK.CUST_ID, :CUSTOMER_BLOCK.F_NAME, :CUSTOMER_BLOCK.L_NAME, :CUSTOMER_BLOCK.EMAIL, :CUSTOMER_BLOCK.PHONE, :CUSTOMER_BLOCK.NIC, :CUSTOMER_BLOCK.STREET, :CUSTOMER_BLOCK.TOWN, :CUSTOMER_BLOCK.POSTAL_CODE);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('CUSTOMER_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_CUSTOMER_FORM;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE CUSTOMER
SET F_NAME = :CUSTOMER_BLOCK.F_NAME,
L_NAME = :CUSTOMER_BLOCK.L_NAME,
EMAIL =:CUSTOMER_BLOCK.EMAIL,
PHONE = :CUSTOMER_BLOCK.PHONE,
NIC = :CUSTOMER_BLOCK.NIC,
STREET = :CUSTOMER_BLOCK.STREET,
TOWN = :CUSTOMER_BLOCK.TOWN,
POSTAL_CODE = :CUSTOMER_BLOCK.POSTAL_CODE
WHERE CUST_ID = :CUSTOMER_BLOCK.CUST_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('CUSTOMER_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_CUSTOMER_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_CUSTOMER_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_CUSTOMER_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR CUSTOMER FORM

```
PROCEDURE CLEAR_CUSTOMER_FORM IS
BEGIN
  :CUSTOMER_BLOCK.CUST_ID:='';
  :CUSTOMER_BLOCK.F_NAME:='';
  :CUSTOMER_BLOCK.L_NAME:='';
  :CUSTOMER_BLOCK.EMAIL:='';
  :CUSTOMER_BLOCK.PHONE:='';
  :CUSTOMER_BLOCK.NIC:='';
  :CUSTOMER_BLOCK.STREET:='';
  :CUSTOMER_BLOCK.TOWN:='';
  :CUSTOMER_BLOCK.POSTAL_CODE:='';
END;
```

UPDATE ALERT

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

TRIGGER

WHEN-WINDOW-ACTIVATED

```
GO_BLOCK('CUSTOMER_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Category Form

CREATE BUTTON

```
SELECT CATEGORY_ID_SEQ.NEXTVAL
INTO :CATEGORY_BLOCK.CATEGORY_ID
FROM DUAL;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST CATEGORY BUTTON

```
GO_ITEM('CATEGORY_BLOCK.CATEGORY_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

SAVE BUTTON

```
INSERT INTO CATEGORY(CATEGORY_ID,CATEGORY_NAME)
VALUES(:CATEGORY_BLOCK.CATEGORY_ID, :CATEGORY_BLOCK.CATEGORY_NAME);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('CATEGORY_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_CATEGORY_FORM;

SET_ITEM_PROPERTY('LIST_CATEGORY_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE CATEGORY
SET CATEGORY_NAME = :CATEGORY_BLOCK.CATEGORY_NAME
WHERE CATEGORY_ID = :CATEGORY_BLOCK.CATEGORY_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('CATEGORY_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_CATEGORY_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_CATEGORY_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_CATEGORY_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_CATEGORY_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR CATEGORY FORM

```
PROCEDURE CLEAR_CATEGORY_FORM IS
BEGIN
  :CATEGORY_BLOCK.CATEGORY_ID:='';
  :CATEGORY_BLOCK.CATEGORY_NAME:='';
END;
```

<u>UPDATE ALERT</u>

```
UPDATE CATEGORY
SET CATEGORY_NAME = :CATEGORY_BLOCK.CATEGORY_NAME
WHERE CATEGORY_ID = :CATEGORY_BLOCK.CATEGORY_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('CATEGORY_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_CATEGORY_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_CATEGORY_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

TRIGGER

<u>WHEN-WINDOW-ACTIVATED</u>

```
GO_BLOCK('CATEGORY_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Color Form

CREATE BUTTON

```
SET_ITEM_PROPERTY('LIST_COLOR_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
```

LIST COLOR BUTTON

```
GO_ITEM('COLOR_BLOCK.COLOR');
LIST_VALUES;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
```

SAVE BUTTON

```
INSERT INTO COLOR
VALUES(:COLOR_BLOCK.COLOR);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('COLOR_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_COLOR_FORM;

SET_ITEM_PROPERTY('LIST_COLOR_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR COLOR FORM

```
PROCEDURE CLEAR_COLOR_FORM IS
BEGIN
  :COLOR_BLOCK.COLOR:='';
END;
```

TRIGGER

WHEN-WINDOW-ACTIVATED

```
GO_BLOCK('COLOR_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
```

Villa Form

Villa Details

Villa Id	Villa Name	Street	Town	Postal Code	Region	Rent Price	Num Bed	Owner FName	Owner LName	Color	Category Name
1	Bayou Oaks	12 Rue Barracks	Port Louis	20161	West	1500	2	Wayne	Rooney	Light Grey	First Class
2	Diamond Blue	Avenu Rd	Ebene	50652	West	1200	1	Marcus	Rashford	Blue	Economie
3	Auroville	Eagles	Beau Bassin	30651	West	2000	4	Rayan	Giggs	Bright Pink	Luxury
4	Cumberland	43 St Jean Road	Quatre Bornes	40258	West	1700	3	Paul	Pogba	Royal	First Class
5	Mayberry	St Pierre Rd	St Pierre	65203	West	1900	3	Mandela	Nelson	Eggplant	Luxury
6	Queens Mead	Geranium Road	Grand Baie	15975	North	1300	2	Eldeson	Cavani	Yellow	Economie

CREATE BUTTON

```sql
SELECT VILLA_ID_SEQ.NEXTVAL
INTO :VILLA_BLOCK.VILLA_ID
FROM DUAL;


SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST VILLA BUTTON

```sql
GO_ITEM('VILLA_BLOCK.VILLA_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST OWNER BUTTON

```sql
GO_ITEM('VILLA_BLOCK.OWNER_ID');
LIST_VALUES;
```

LIST COLOR BUTTON

```sql
GO_ITEM('VILLA_BLOCK.COLOR');
LIST_VALUES;
```

LIST CATEGORY BUTTON

```sql
GO_ITEM('VILLA_BLOCK.CATEGORY_ID');
LIST_VALUES;
```

SAVE BUTTON

```
INSERT INTO VILLA(VILLA_ID,VILLA_NAME,STREET,TOWN,POSTAL_CODE,REGION,RENT_PRICE,NUM_BED,OWNER_ID,COLOR,CATEGORY_ID)
VALUES(:VILLA_BLOCK.VILLA_ID,
:VILLA_BLOCK.VILLA_NAME,
:VILLA_BLOCK.STREET,
:VILLA_BLOCK.TOWN,
:VILLA_BLOCK.POSTAL_CODE,
:VILLA_BLOCK.REGION,
:VILLA_BLOCK.RENT_PRICE,
:VILLA_BLOCK.NUM_BED,
:VILLA_BLOCK.OWNER_ID,
:VILLA_BLOCK.COLOR,
:VILLA_BLOCK.CATEGORY_ID);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('VILLA_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_VILLA_FORM;

SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE VILLA
SET VILLA_NAME = :VILLA_BLOCK.VILLA_NAME,
STREET = :VILLA_BLOCK.STREET,
TOWN = :VILLA_BLOCK.TOWN,
POSTAL_CODE = :VILLA_BLOCK.POSTAL_CODE,
REGION = :VILLA_BLOCK.REGION,
RENT_PRICE = :VILLA_BLOCK.RENT_PRICE,
NUM_BED = :VILLA_BLOCK.NUM_BED,
OWNER_ID = :VILLA_BLOCK.OWNER_ID,
COLOR = :VILLA_BLOCK.COLOR,
CATEGORY_ID = :VILLA_BLOCK.CATEGORY_ID
WHERE VILLA_ID = :VILLA_BLOCK.VILLA_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');
DISPLAY_ALERT;

GO_BLOCK('VILLA_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_VILLA_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_VILLA_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR VILLA FORM

```
PROCEDURE CLEAR_VILLA_FORM IS
BEGIN
  :VILLA_BLOCK.VILLA_ID:='';
  :VILLA_BLOCK.VILLA_NAME:='';
  :VILLA_BLOCK.STREET:='';
  :VILLA_BLOCK.TOWN:='';
  :VILLA_BLOCK.POSTAL_CODE:='';
  :VILLA_BLOCK.REGION:='';
  :VILLA_BLOCK.RENT_PRICE:='';
  :VILLA_BLOCK.NUM_BED:='';
  :VILLA_BLOCK.OWNER_ID:='';
  :VILLA_BLOCK.COLOR:='';
  :VILLA_BLOCK.CATEGORY_ID:='';
END;
```

UPDATE ALERT

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

TRIGGER

WHEN-WINDOW-ACTIVATED

```
GO_BLOCK('VILLA_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Options Form

The picture below shows an update alert when updating a record.

CREATE BUTTON

```
SELECT OPTION_ID_SEQ.NEXTVAL
INTO :OPTIONS_BLOCK.OPTION_ID
FROM DUAL;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('LIST_OPTIONS_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST OPTIONS BUTTON

```
GO_ITEM('OPTIONS_BLOCK.OPTION_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

SAVE BUTTON

```
INSERT INTO OPTIONS(OPTION_ID,OPTION_NAME)
VALUES(:OPTIONS_BLOCK.OPTION_ID, :OPTIONS_BLOCK.OPTION_NAME);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('OPTIONS_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_OPTIONS_FORM;

SET_ITEM_PROPERTY('LIST_OPTIONS_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE OPTIONS
SET OPTION_NAME = :OPTIONS_BLOCK.OPTION_NAME
WHERE OPTION_ID = :OPTIONS_BLOCK.OPTION_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('OPTIONS_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_OPTIONS_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_OPTIONS_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_OPTIONS_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_OPTIONS_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR OPTIONS FORM

```
PROCEDURE CLEAR_OPTIONS_FORM IS
BEGIN
  :OPTIONS_BLOCK.OPTION_ID:='';
  :OPTIONS_BLOCK.OPTION_NAME:='';
END;
```

UPDATE ALERT

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

TRIGGER

<u>WHEN-WINDOW-ACTIVATED</u>

```
GO_BLOCK('OPTIONS_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Rental Form

CREATE BUTTON

```
SELECT RENT_ID_SEQ.NEXTVAL
INTO :RENTAL_BLOCK.RENT_ID
FROM DUAL;

SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_CUSTOMER_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST RENTAL BUTTON

```
GO_ITEM('RENTAL_BLOCK.RENT_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST VILLA BUTTON

```
GO_ITEM('RENTAL_BLOCK.VILLA_ID');
LIST_VALUES;
```

LIST CUSTOMER BUTTON

```
GO_ITEM('RENTAL_BLOCK.CUST_ID');
LIST_VALUES;
```

SAVE BUTTON

```
INSERT INTO RENTAL(RENT_ID,CHECK_IN_DATE,CHECK_OUT_DATE,STATUS,CANCEL_DATE,VILLA_ID,CUST_ID)
VALUES(:RENTAL_BLOCK.RENT_ID,
:RENTAL_BLOCK.CHECK_IN_DATE,
:RENTAL_BLOCK.CHECK_OUT_DATE,
:RENTAL_BLOCK.STATUS,
:RENTAL_BLOCK.CANCEL_DATE,
:RENTAL_BLOCK.VILLA_ID,
:RENTAL_BLOCK.CUST_ID);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');


GO_BLOCK('RENTAL_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_RENTAL_FORM;

SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE RENTAL
SET CHECK_IN_DATE = :RENTAL_BLOCK.CHECK_IN_DATE,
CHECK_OUT_DATE = :RENTAL_BLOCK.CHECK_OUT_DATE,
STATUS = :RENTAL_BLOCK.STATUS,
CANCEL_DATE = :RENTAL_BLOCK.CANCEL_DATE,
VILLA_ID = :RENTAL_BLOCK.VILLA_ID,
CUST_ID = :RENTAL_BLOCK.CUST_ID
WHERE RENT_ID = :RENTAL_BLOCK.RENT_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');

DISPLAY_ALERT;

GO_BLOCK('RENTAL_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_RENTAL_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_RENTAL_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR RENTAL FORM

```
PROCEDURE CLEAR_RENTAL_FORM IS
BEGIN
  :RENTAL_BLOCK.RENT_ID:='';
  :RENTAL_BLOCK.CHECK_IN_DATE:='';
  :RENTAL_BLOCK.CHECK_OUT_DATE:='';
  :RENTAL_BLOCK.STATUS:='';
  :RENTAL_BLOCK.CANCEL_DATE:='';
  :RENTAL_BLOCK.VILLA_ID:='';
  :RENTAL_BLOCK.CUST_ID:='';
END;
```

<u>UPDATE ALERT</u>

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

TRIGGER

<u>WHEN-WINDOW-ACTIVATED</u>

```
GO_BLOCK('RENTAL_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('LIST_CUSTOMER_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Payment Form

Payment Details

Payment ID	Payment Date	Amount	Payment Method	Rent ID	Status	Cust FName	Cust LName
1	04-FEB-2020	7500	CASH	1	Paid	Taaj	Beeharry
2	01-APR-2020	1200	CARD	2	Cancel	Kawthar	Dowlut
3	14-MAR-2020	428000	CARD	3	Paid	Poujan	Jagoo
4	16-MAY-2020	15600	CASH	6	Paid	John	Smith
5	22-JUN-2020	1900	CASH	5	Cancel	Mahima	Mohun
6	19-DEC-2020	51000	CASH	4	Paid	Tom	Swaryer

Below the picture shows a delete alert when deleting a record in the database.

CREATE BUTTON

```
SELECT PAYMENT_ID_SEQ.NEXTVAL
INTO :PAYMENT_BLOCK.PAYMENT_ID
FROM DUAL;


SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('SAVE_BUTTON');
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

LIST PAYMENT BUTTON

```
GO_ITEM('PAYMENT_BLOCK.PAYMENT_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_TRUE);
```

LIST RENTAL BUTTON

```
GO_ITEM('PAYMENT_BLOCK.RENT_ID');
LIST_VALUES;
```

SAVE BUTTON

```
INSERT INTO PAYMENT(PAYMENT_ID,PAYMENT_DATE,AMOUNT,PAYMENT_METHOD,RENT_ID)
VALUES(:PAYMENT_BLOCK.PAYMENT_ID,
:PAYMENT_BLOCK.PAYMENT_DATE,
:PAYMENT_BLOCK.AMOUNT,
:PAYMENT_BLOCK.PAYMENT_METHOD,
:PAYMENT_BLOCK.RENT_ID);
COMMIT;
MESSAGE('RECORD SAVE SUCCESSFULLY');

GO_BLOCK('PAYMENT_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_PAYMENT_FORM;

SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

UPDATE BUTTON

```
UPDATE PAYMENT
SET PAYMENT_DATE = :PAYMENT_BLOCK.PAYMENT_DATE,
AMOUNT = :PAYMENT_BLOCK.AMOUNT,
PAYMENT_METHOD = :PAYMENT_BLOCK.PAYMENT_METHOD,
RENT_ID = :PAYMENT_BLOCK.RENT_ID
WHERE PAYMENT_ID = :PAYMENT_BLOCK.PAYMENT_ID;

--COMMIT;
--MESSAGE('RECORD UPDATED SUCCESSFULLY');
DISPLAY_ALERT;

GO_BLOCK('PAYMENT_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_PAYMENT_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

DELETE BUTTON

```
--DELETE FROM PAYMENT
--WHERE PAYMENT_ID = :PAYMENT_BLOCK.PAYMENT_ID;

--COMMIT;
--MESSAGE('RECORD DELETED SUCCESSFULLT');

DISPLAY_DELETE_ALERT;

GO_BLOCK('PAYMENT_VIEW_BLOCK');
EXECUTE_QUERY;

CLEAR_PAYMENT_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

CLEAR BUTTON

```
CLEAR_PAYMENT_FORM;

SET_ITEM_PROPERTY('NEW_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('NEW_BUTTON');
SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_TRUE);
SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Program Unit

CLEAR PAYMENT FORM

```
PROCEDURE CLEAR_PAYMENT_FORM IS
BEGIN
  :PAYMENT_BLOCK.PAYMENT_ID:='';
  :PAYMENT_BLOCK.PAYMENT_DATE:='';
  :PAYMENT_BLOCK.AMOUNT:='';
  :PAYMENT_BLOCK.PAYMENT_METHOD:='';
  :PAYMENT_BLOCK.RENT_ID:='';
END;
```

UPDATE ALERT

```
PROCEDURE DISPLAY_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('UPDATE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    COMMIT;
    MESSAGE('RECORD UPDATED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    ROLLBACK;
    MESSAGE('RECORD NOT UPDATED');
    END IF;
END;
```

DELETE ALERT

```
PROCEDURE DISPLAY_DELETE_ALERT IS
alert_button NUMBER;
BEGIN
  alert_button := SHOW_ALERT('DELETE_ALERT');
  IF alert_button = ALERT_BUTTON1 THEN
    DELETE FROM PAYMENT
    WHERE PAYMENT_ID = :PAYMENT_BLOCK.PAYMENT_ID;
    COMMIT;
    MESSAGE('RECORD DELETED SUCCESSFULLY');
    ELSIF alert_button = ALERT_BUTTON2 THEN
    RETURN;
  END IF;
END;
```

TRIGGER

WHEN-WINDOW-ACTIVATED

```
GO_BLOCK('PAYMENT_VIEW_BLOCK');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('SAVE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('UPDATE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('DELETE_BUTTON',ENABLED,PROPERTY_FALSE);
SET_ITEM_PROPERTY('CANCEL_BUTTON',ENABLED,PROPERTY_FALSE);
```

Trigger

These two triggers are applied in every forms.

ON-ERROR

```
IF ERROR_CODE = 50016 THEN

MESSAGE('Please enter a value that is a number');

ELSIF ERROR_CODE = 40735 THEN

IF DBMS_ERROR_CODE = -2292 THEN

MESSAGE('This record is referenced by other database objects');

ELSIF DBMS_ERROR_CODE =-01400 THEN

MESSAGE('Field cannot be empty.');

ELSIF DBMS_ERROR_CODE =-0001 THEN

MESSAGE('Field must be unique!');

ELSE

MESSAGE('DBMS_ERROR_TEXT');

END IF;

ELSE

MESSAGE('ERROR_TEXT');

END IF;
```

PRE-FORM

```
:SYSTEM.MESSAGE_LEVEL := 25;
```

Queries

Customer_Payment_Summary_View

LIST CUSTOMER BUTTON

```
GO_ITEM('CUSTOMER_PAYMENT_BLOCK.CUST_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('LIST_PAYMENT_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('LIST_PAYMENT_BUTTON');
```

LIST PAYMENT BUTTON

```
GO_BLOCK('CUSTOMER_PAYMENT_SUMMARY_VIEW');
EXECUTE_QUERY;
```

Customer_Villa_Summary_View

LIST CUSTOMER BUTTON

```
GO_ITEM('CUSTOMER_VILLA_BLOCK.CUST_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('LIST_RENTAL_BUTTON');
```

LIST RENTAL BUTTON

```
GO_BLOCK('CUSTOMER_VILLA_SUMMARY_VIEW');
EXECUTE_QUERY;

SET_ITEM_PROPERTY('LIST_RENTAL_BUTTON',ENABLED,PROPERTY_FALSE);
```

List_Rental_Summary_View

LIST RENTAL BUTTON

```
GO_ITEM('RENTAL_BLOCK.RENT_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('RENTAL_HISTORY_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('RENTAL_HISTORY_BUTTON');
```

LIST RENTAL HISTORY BUTTON

```
GO_BLOCK('LIST_RENTAL_SUMMARY_VIEW');
EXECUTE_QUERY;
```

List_Villa_Summary_View

LIST VILLA BUTTON

```
GO_BLOCK('LIST_VILLA_SUMMARY_VIEW');
EXECUTE_QUERY;
```

Options_Villa_Summary_View

LIST OPTION BUTTON

```
GO_ITEM('OPTIONS_VILLA_BLOCK.OPTION_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('LIST_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('LIST_VILLA_BUTTON');
```

LIST VILLA BUTTON

```
GO_BLOCK('VILLA_OPTIONS_SUMMARY_VIEW');
EXECUTE_QUERY;
```

Owner_Villa_Summary_View

LIST OWNER BUTTON

```
GO_ITEM('OWNER_BLOCK.OWNER_ID');
LIST_VALUES;

SET_ITEM_PROPERTY('LIST_ALL_VILLA_BUTTON',ENABLED,PROPERTY_TRUE);
GO_ITEM('LIST_ALL_VILLA_BUTTON');
```

LIST ALL VILLA BUTTON

```
GO_BLOCK('OWNER_VILLA_SUMMARY_VIEW');
EXECUTE_QUERY;
```

LIST CLEAR ALL BUTTON

```
BEGIN
:OWNER_BLOCK.OWNER_ID:='';
:OWNER_BLOCK.F_NAME:='';
:OWNER_BLOCK.L_NAME:='';
:OWNER_BLOCK.PHONE:='';
:OWNER_BLOCK.STREET:='';
:OWNER_BLOCK.TOWN:='';
Clear_Block(No_Validate);
END;
SET_ITEM_PROPERTY('LIST_ALL_VILLA_BUTTON',ENABLED,PROPERTY_FALSE);
```

SQL Script

--Script to create Villa Rental System--

DROP TABLE OWNER CASCADE CONSTRAINTS;

DROP TABLE CUSTOMER CASCADE CONSTRAINTS;

DROP TABLE CATEGORY CASCADE CONSTRAINTS;

DROP TABLE COLOR CASCADE CONSTRAINTS;

DROP TABLE VILLA CASCADE CONSTRAINTS;

DROP TABLE OPTIONS CASCADE CONSTRAINTS;

DROP TABLE OPTION_VILLA CASCADE CONSTRAINTS;

DROP TABLE RENTAL CASCADE CONSTRAINTS;

DROP TABLE PAYMENT CASCADE CONSTRAINTS;

--CREATE OWNER--

CREATE TABLE OWNER

(OWNER_ID NUMBER(6),

F_NAME VARCHAR2(15) CONSTRAINT OWNER_F_NAME_NN NOT NULL,

L_NAME VARCHAR2(15) CONSTRAINT OWNER_L_NAME_NN NOT NULL,

PHONE NUMBER(15),

STREET VARCHAR2(50) CONSTRAINT OWNER_STREET_NN NOT NULL,

TOWN VARCHAR2(50) CONSTRAINT OWNER_TOWN_NN NOT NULL,

CONSTRAINT OWNER_OWNER_ID_PK PRIMARY KEY(OWNER_ID));

```sql
--CREATE CUSTOMER--

CREATE TABLE CUSTOMER

(CUST_ID NUMBER(6),

F_NAME VARCHAR2(15) CONSTRAINT CUSTOMER_F_NAME_NN NOT NULL,

L_NAME VARCHAR2(15) CONSTRAINT CUSTOMER_L_NAME_NN NOT NULL,

EMAIL VARCHAR2(50),

PHONE NUMBER(15) CONSTRAINT CUSTOMER_PHONE_NN NOT NULL,

NIC VARCHAR2(17) CONSTRAINT CUSTOMER_NIC_NN NOT NULL,

STREET VARCHAR2(50) CONSTRAINT CUSTOMER_STREET_NN NOT NULL,

TOWN VARCHAR2(50) CONSTRAINT CUSTOMER_TOWN_NN NOT NULL,

POSTAL_CODE NUMBER(5),

CONSTRAINT UC_CUSTOMER UNIQUE(EMAIL),

CONSTRAINT CUSTOMER_CUST_ID_PK PRIMARY KEY(CUST_ID));


--CREATE CATEGORY--

CREATE TABLE CATEGORY

(CATEGORY_ID NUMBER(6),

CATEGORY_NAME VARCHAR2(50),

CONSTRAINT UC_CATEGORY UNIQUE(CATEGORY_NAME),

CONSTRAINT CATEGORY_CATEGORY_ID_PK PRIMARY KEY(CATEGORY_ID));


--CREATE COLOR--

CREATE TABLE COLOR

(COLOR VARCHAR2(15),

CONSTRAINT COLOR_COLOR_PK PRIMARY KEY(COLOR));


--CREATE VILLA--

CREATE TABLE VILLA

(VILLA_ID NUMBER(6),

VILLA_NAME VARCHAR(50) CONSTRAINT VILLA_VILLA_NAME_NN NOT NULL,

STREET VARCHAR2(50) CONSTRAINT VILLA_STREET_NN NOT NULL,
```

TOWN VARCHAR2(50) CONSTRAINT VILLA_TOWN_NN NOT NULL,

POSTAL_CODE NUMBER(5),

REGION VARCHAR2(15) CONSTRAINT VILLA_REGION_NN NOT NULL,

RENT_PRICE NUMBER(8,2) CONSTRAINT VILLA_RENT_PRICE_NN NOT NULL,

NUM_BED NUMBER(2) CONSTRAINT VILLA_NUM_BED_NN NOT NULL,

OWNER_ID NUMBER(6),

COLOR VARCHAR2(15),

CATEGORY_ID NUMBER(6),

CONSTRAINT VILLA_VILLA_ID_PK PRIMARY KEY(VILLA_ID),

CONSTRAINT VILLA_OWNER_ID_FK FOREIGN KEY(OWNER_ID) REFERENCES OWNER(OWNER_ID),

CONSTRAINT VILLA_COLOR_FK FOREIGN KEY(COLOR) REFERENCES COLOR(COLOR),

CONSTRAINT VILLA_CATEGORY_ID_FK FOREIGN KEY(CATEGORY_ID) REFERENCES
CATEGORY(CATEGORY_ID));

--CREATE OPTION--

CREATE TABLE OPTIONS

(OPTION_ID NUMBER(6),

OPTION_NAME VARCHAR2(50),

CONSTRAINT UC_OPTIONS UNIQUE(OPTION_NAME),

CONSTRAINT OPTIONS_OPTION_ID_PK PRIMARY KEY(OPTION_ID));

--CREATE OPTION_VILLA--

CREATE TABLE OPTION_VILLA

(OPTION_ID NUMBER(6),

VILLA_ID NUMBER(6),

CONSTRAINT OPTION_VILLA_PK PRIMARY KEY(OPTION_ID, VILLA_ID));

--CREATE RENTAL--

CREATE TABLE RENTAL

(RENT_ID NUMBER(6),

CHECK_IN_DATE DATE,

```sql
CHECK_OUT_DATE DATE,

STATUS VARCHAR2(10),

CANCEL_DATE DATE,

VILLA_ID NUMBER(6),

CUST_ID NUMBER(6),

CONSTRAINT RENTAL_RENT_ID_PK PRIMARY KEY(RENT_ID),

CONSTRAINT RENTAL_VILLA_ID_FK FOREIGN KEY(VILLA_ID) REFERENCES VILLA(VILLA_ID),

CONSTRAINT RENTAL_CUST_ID_FK FOREIGN KEY(CUST_ID) REFERENCES CUSTOMER(CUST_ID));


--CREATE PAYMENT--

CREATE TABLE PAYMENT

(PAYMENT_ID NUMBER(6),

PAYMENT_DATE DATE,

AMOUNT NUMBER(8,2),

PAYMENT_METHOD CHAR(4),

RENT_ID NUMBER(6),

CONSTRAINT PAYMENT_PAYMENT_ID_PK PRIMARY KEY(PAYMENT_ID),

CONSTRAINT PAYMENT_RENT_ID_FK FOREIGN KEY(RENT_ID) REFERENCES RENTAL(RENT_ID));



--INSERT OWNER--

INSERT INTO OWNER (OWNER_ID, F_NAME, L_NAME, PHONE, STREET, TOWN)

VALUES (1,'Paul','Pogba',8154751982,'1156 Water Street, Apt. #3','Osseo');

INSERT INTO OWNER VALUES (2,'Marcus','Rashford',8112233445,'2211 Pine Drive','Radisson');

INSERT INTO OWNER VALUES (3,'Wayne','Rooney',8223344775,'699 Pluto St. NW','Silver Lake');

INSERT INTO OWNER VALUES (4,'Eldeson','Cavani',8112247514,'987 Durham Rd.','Apple Valley');

INSERT INTO OWNER VALUES (5,'Rayan','Giggs',8297125433,'4204 Garner Street','Washburn');

INSERT INTO OWNER VALUES (6,'Mandela','Nelson',8451145254,'232 Echo Rd.','Minnetonka');



--INSERT CUSTOMER--
```

INSERT INTO CUSTOMER (CUST_ID, F_NAME, L_NAME, EMAIL, PHONE, NIC, STREET, TOWN, POSTAL_CODE)

VALUES (1,'Taaj','Beeharry','beeharrytaaj@gmail.com',9921475328,'B1605001407602','Royal Road Pont Praslin','Pamplemousses',54705);

INSERT INTO CUSTOMER VALUES (2,'Poujan','Jagoo','sanyogjagoo21@gmail.com',9957481358,'J2105003524567','Denis Street','Roche Noire',54867);

INSERT INTO CUSTOMER VALUES (3,'Kawthar','Dowlut','Kawthar2807@gmail.com',9174820540,'D1402985128153','30 Avenue','Port Louis',53821);

INSERT INTO CUSTOMER VALUES (4,'Mahima','Mohun','mahima@gmail.com',9874501400,'M3003007539514','Birla Road Cottage','Cottage',55712);

INSERT INTO CUSTOMER VALUES (5,'John','Smith','johnS@yahoo.com',9874001479,'S0101982148539','Kadeer Road','Belvedere',54891);

INSERT INTO CUSTOMER VALUES (6,'Tom','Swaryer','tomc@gmail.com',9017080542,'S2005861475201','7th mille','Triolet',55438);

--INSERT CATEGORY--

INSERT INTO CATEGORY (CATEGORY_ID, CATEGORY_NAME)

VALUES (1,'Economie');

INSERT INTO CATEGORY VALUES (2,'First Class');

INSERT INTO CATEGORY VALUES (3,'Luxury');

--INSERT COLOR--

INSERT INTO COLOR (COLOR)

VALUES ('Sky Blue');

INSERT INTO COLOR VALUES ('Light Grey');

INSERT INTO COLOR VALUES ('Khaki');

INSERT INTO COLOR VALUES ('Navy');

INSERT INTO COLOR VALUES ('Royal');

INSERT INTO COLOR VALUES ('Eggplant');

INSERT INTO COLOR VALUES ('Blue');

INSERT INTO COLOR VALUES ('Spruce');

INSERT INTO COLOR VALUES ('Turquoise');

INSERT INTO COLOR VALUES ('Bright Pink');

INSERT INTO COLOR VALUES ('White');

INSERT INTO COLOR VALUES ('Black');

INSERT INTO COLOR VALUES ('Yellow');

--INSERT VILLA--

INSERT INTO VILLA (VILLA_ID, VILLA_NAME, STREET, TOWN, POSTAL_CODE, REGION, RENT_PRICE, NUM_BED, OWNER_ID, COLOR, CATEGORY_ID)

VALUES (1,'Bayou Oaks','12 Rue Barracks','Port Louis',20161,'West',1500.00,2,3,'Light Grey',2);

INSERT INTO VILLA VALUES (2,'Diamond Blue','Avenu Rd','Ebene',50652,'West',1200.00,1,2,'Blue',1);

INSERT INTO VILLA VALUES (3,'Auroville','Eagles','Beau Bassin',30651,'West',2000.00,4,5,'Bright Pink',3);

INSERT INTO VILLA VALUES (4,'Cumberland','43 St Jean Road','Quatre Bornes',40258,'West',1700.00,3,1,'Royal',2);

INSERT INTO VILLA VALUES (5,'Mayberry','St Pierre Rd','St Pierre',65203,'West',1900.00,3,6,'Eggplant',3);

INSERT INTO VILLA VALUES (6,'Queens Mead','Geranium Road','Grand Baie',15975,'North',1300.00,2,4,'Yellow',1);

--INSERT OPTION--

INSERT INTO OPTIONS (OPTION_ID, OPTION_NAME)

VALUES (1,'Garden');

INSERT INTO OPTIONS VALUES (2,'Swimming Pool');

INSERT INTO OPTIONS VALUES (3,'Kids corner');

--OPTION_VILLA--

INSERT INTO OPTION_VILLA (OPTION_ID, VILLA_ID)

VALUES(3,4);

INSERT INTO OPTION_VILLA VALUES(2,5);

INSERT INTO OPTION_VILLA VALUES(1,2);

INSERT INTO OPTION_VILLA VALUES(1,1);

INSERT INTO OPTION_VILLA VALUES(3,3);

INSERT INTO OPTION_VILLA VALUES(2,6);

INSERT INTO OPTION_VILLA VALUES(3,1);

INSERT INTO OPTION_VILLA VALUES(2,1);

INSERT INTO OPTION_VILLA VALUES(2,2);

INSERT INTO OPTION_VILLA VALUES(2,3);

INSERT INTO OPTION_VILLA VALUES(1,3);

--INSERT RENTAL--

INSERT INTO RENTAL (RENT_ID, CHECK_IN_DATE, CHECK_OUT_DATE, STATUS, CANCEL_DATE, VILLA_ID, CUST_ID)
VALUES
(1,TO_DATE('04/02/2020','DD/MM/YYYY'),TO_DATE('09/02/2020','DD/MM/YYYY'),'Paid',null,1,1);

INSERT INTO RENTAL VALUES
(2,TO_DATE('01/04/2020','DD/MM/YYYY'),null,'Cancel',TO_DATE('02/Apr/2020','DD/MM/YYYY'),2,3)
;

INSERT INTO RENTAL VALUES
(3,TO_DATE('14/03/2020','DD/MM/YYYY'),TO_DATE('14/09/2020','DD/MM/YYYY'),'Paid',null,3,2);

INSERT INTO RENTAL VALUES
(4,TO_DATE('16/05/2020','DD/MM/YYYY'),TO_DATE('16/06/2020','DD/MM/YYYY'),'Paid',null,4,6);

INSERT INTO RENTAL VALUES
(5,TO_DATE('22/06/2020','DD/MM/YYYY'),null,'Cancel',TO_DATE('23/Jun/2020','DD/MM/YYYY'),5,4);

INSERT INTO RENTAL VALUES
(6,TO_DATE('19/12/2020','DD/MM/YYYY'),TO_DATE('31/12/2020','DD/MM/YYYY'),'Paid',null,6,5);

--INSERT PAYMENT--

INSERT INTO PAYMENT (PAYMENT_ID, PAYMENT_DATE, AMOUNT, PAYMENT_METHOD, RENT_ID)

VALUES (1,TO_DATE('04/02/2020','DD/MM/YYYY'),7500.00,'CASH',1);

INSERT INTO PAYMENT VALUES (2,TO_DATE('01/04/2020','DD/MM/YYYY'),1200.00,'CARD',2);

INSERT INTO PAYMENT VALUES (3,TO_DATE('14/03/2020','DD/MM/YYYY'),428000.00,'CARD',3);

INSERT INTO PAYMENT VALUES (4,TO_DATE('16/05/2020','DD/MM/YYYY'),15600.00,'CASH',6);

INSERT INTO PAYMENT VALUES (5,TO_DATE('22/06/2020','DD/MM/YYYY'),1900.00,'CASH',5);

INSERT INTO PAYMENT VALUES (6,TO_DATE('19/12/2020','DD/MM/YYYY'),51000.00,'CASH',4);

COMMIT;

-- V I E W S ---

--OWNER_VIEW--

CREATE OR REPLACE VIEW OWNER_VIEW

AS

SELECT OWNER_ID, F_NAME, L_NAME, PHONE, STREET, TOWN

FROM OWNER

ORDER BY OWNER_ID ASC;

--CUSTOMER_VIEW--

CREATE OR REPLACE VIEW CUSTOMER_VIEW

AS

SELECT CUST_ID, F_NAME, L_NAME, EMAIL, PHONE, NIC, STREET, TOWN, POSTAL_CODE

FROM CUSTOMER

ORDER BY CUST_ID ASC;

--CATEGORY_VIEW--

CREATE OR REPLACE VIEW CATEGORY_VIEW

AS

```sql
SELECT CATEGORY_ID, CATEGORY_NAME

FROM CATEGORY

ORDER BY CATEGORY_ID ASC;




--COLOR_VIEW--

CREATE OR REPLACE VIEW COLOR_VIEW

AS

SELECT COLOR

FROM COLOR

ORDER BY COLOR ASC;




--VILLA_VIEW--

CREATE OR REPLACE VIEW VILLA_VIEW

AS

SELECT V.VILLA_ID, V.VILLA_NAME, V.STREET, V.TOWN, V.POSTAL_CODE, V.REGION, V.RENT_PRICE,
V.NUM_BED, O.F_NAME, O.L_NAME, CO.COLOR, CA.CATEGORY_NAME

FROM VILLA V, OWNER O, COLOR CO, CATEGORY CA

WHERE V.OWNER_ID=O.OWNER_ID

AND V.COLOR=CO.COLOR

AND V.CATEGORY_ID=CA.CATEGORY_ID

ORDER BY VILLA_ID ASC;




--OPTIONS_VIEW--

CREATE OR REPLACE VIEW OPTIONS_VIEW

AS

SELECT OPTION_ID, OPTION_NAME

FROM OPTIONS
```

ORDER BY OPTION_ID ASC;

--OPTION_VILLA_VIEW--

CREATE OR REPLACE VIEW OPTION_VILLA_VIEW

AS

SELECT OPTION_NAME, VILLA_NAME

FROM OPTIONS, VILLA

WHERE OPTION_ID=VILLA_ID;

--RENTAL_VIEW--

CREATE OR REPLACE VIEW RENTAL_VIEW

AS

SELECT R.RENT_ID, R.CHECK_IN_DATE, R.CHECK_OUT_DATE, R.STATUS, R.CANCEL_DATE,
VA.VILLA_NAME, C.F_NAME, C.L_NAME

FROM RENTAL R, VILLA VA, CUSTOMER C

WHERE R.VILLA_ID=VA.VILLA_ID

AND R.CUST_ID=C.CUST_ID

ORDER BY RENT_ID ASC;

--PAYMENT_VIEW--

CREATE OR REPLACE VIEW PAYMENT_VIEW

AS

SELECT PAY.PAYMENT_ID, PAY.PAYMENT_DATE, PAY.AMOUNT, PAY.PAYMENT_METHOD,
REN.RENT_ID, REN.STATUS, CUST.F_NAME, CUST.L_NAME

FROM PAYMENT PAY, RENTAL REN, CUSTOMER CUST

WHERE PAY.RENT_ID=REN.RENT_ID

AND REN.CUST_ID=CUST.CUST_ID

ORDER BY PAYMENT_ID ASC;

--OWNER_VILLA_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW OWNER_VILLA_SUMMARY_VIEW

AS

SELECT OW.OWNER_ID, VIL.VILLA_ID, VIL.VILLA_NAME, VIL.STREET, VIL.TOWN, VIL.POSTAL_CODE, VIL.REGION, VIL.RENT_PRICE, VIL.NUM_BED, CATE.CATEGORY_ID, CATE.CATEGORY_NAME, COL.COLOR

FROM OWNER OW, VILLA VIL, CATEGORY CATE, COLOR COL

WHERE VIL.OWNER_ID=OW.OWNER_ID

AND COL.COLOR=VIL.COLOR

AND CATE.CATEGORY_ID=VIL.CATEGORY_ID

ORDER BY OWNER_ID ASC;

--CUSTOMER_VILLA_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW CUSTOMER_VILLA_SUMMARY_VIEW

AS

SELECT CUSTO.F_NAME, CUSTO.L_NAME, VILL.VILLA_ID, VILL.VILLA_NAME, VILL.STREET, VILL.TOWN, VILL.POSTAL_CODE, VILL.REGION, VILL.RENT_PRICE, VILL.NUM_BED, CL.COLOR, CA.CATEGORY_ID, CA.CATEGORY_NAME, RE.RENT_ID, RE.CHECK_IN_DATE, RE.CHECK_OUT_DATE, RE.STATUS, RE.CANCEL_DATE

FROM CUSTOMER CUSTO, VILLA VILL, COLOR CL, CATEGORY CA, RENTAL RE

WHERE RE.VILLA_ID=VILL.VILLA_ID

AND RE.CUST_ID=CUSTO.CUST_ID

AND VILL.CATEGORY_ID=CA.CATEGORY_ID

AND VILL.COLOR=CL.COLOR;

--VILLA_OPTIONS_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW VILLA_OPTIONS_SUMMARY_VIEW

AS

SELECT V.VILLA_ID, O.OPTION_ID, VILLA_NAME, OPTION_NAME, STREET, TOWN, POSTAL_CODE, REGION, RENT_PRICE, NUM_BED

FROM VILLA V, OPTIONS O, OPTION_VILLA OV

WHERE O.OPTION_ID=OV.OPTION_ID

AND OV.VILLA_ID=V.VILLA_ID

ORDER BY V.VILLA_ID ASC;

--CUSTOMER_PAYMENT_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW CUSTOMER_PAYMENT_SUMMARY_VIEW

AS

SELECT C.CUST_ID, C.F_NAME, C.L_NAME, C.EMAIL, C.PHONE, C.NIC, C.STREET, C.TOWN, C.POSTAL_CODE, P.PAYMENT_ID, V.VILLA_ID, V.VILLA_NAME, P.PAYMENT_DATE, P.AMOUNT, P.PAYMENT_METHOD

FROM CUSTOMER C, PAYMENT P, VILLA V ,RENTAL R

WHERE V.VILLA_ID=R.VILLA_ID

AND R.RENT_ID=P.RENT_ID

AND R.CUST_ID=C.CUST_ID

ORDER BY C.CUST_ID ASC;

--LIST_RENTAL_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW LIST_RENTAL_SUMMARY_VIEW

AS

SELECT R.RENT_ID, R.CHECK_IN_DATE, R.CHECK_OUT_DATE, R.STATUS, R.CANCEL_DATE, V.VILLA_NAME, CA.CATEGORY_NAME, CO.COLOR, CU.F_NAME, CU.L_NAME, CU.NIC, CU.PHONE

FROM RENTAL R, VILLA V, CATEGORY CA, COLOR CO, CUSTOMER CU

WHERE R.VILLA_ID=V.VILLA_ID

AND R.CUST_ID=CU.CUST_ID

AND V.COLOR=CO.COLOR

AND V.CATEGORY_ID=CA.CATEGORY_ID

ORDER BY R.RENT_ID ASC;

--LIST_VILLA_SUMMARY_VIEW_QUERY--

CREATE OR REPLACE VIEW LIST_VILLA_SUMMARY_VIEW

AS

SELECT VI.VILLA_ID, VI.VILLA_NAME, CAT.CATEGORY_ID, CAT.CATEGORY_NAME, C.COLOR, OP.OPTION_NAME, VI.STREET, VI.TOWN, VI.POSTAL_CODE, VI.REGION, VI.RENT_PRICE, VI.NUM_BED, O.F_NAME, O.L_NAME

FROM VILLA VI, CATEGORY CAT, COLOR C, OPTIONS OP, OWNER O, OPTION_VILLA OPV

WHERE VI.OWNER_ID=O.OWNER_ID

AND VI.CATEGORY_ID=CAT.CATEGORY_ID

AND VI.COLOR=C.COLOR

AND OPV.VILLA_ID=VI.VILLA_ID

AND OPV.OPTION_ID=OP.OPTION_ID

ORDER BY VI.VILLA_ID ASC;

-- S E Q U E N C E S --

DROP SEQUENCE OWNER_ID_SEQ;

DROP SEQUENCE CUST_ID_SEQ;

DROP SEQUENCE CATEGORY_ID_SEQ;

DROP SEQUENCE VILLA_ID_SEQ;

DROP SEQUENCE OPTION_ID_SEQ;

DROP SEQUENCE RENT_ID_SEQ;

DROP SEQUENCE PAYMENT_ID_SEQ;

--OWNER_SEQUENCE--

CREATE SEQUENCE OWNER_ID_SEQ

START WITH 10;

--CUSTOMER_SEQUENCE--

CREATE SEQUENCE CUST_ID_SEQ

START WITH 10;

--CATEGORY_SEQUENCE--

CREATE SEQUENCE CATEGORY_ID_SEQ

START WITH 10;

--VILLA_SEQUENCE--

CREATE SEQUENCE VILLA_ID_SEQ

START WITH 10;

--OPTIONS_SEQUENCE--

CREATE SEQUENCE OPTION_ID_SEQ

START WITH 10;

--RENTAL_SEQUENCE--

CREATE SEQUENCE RENT_ID_SEQ

START WITH 10;

--PAYMENT_SEQUENCE--

CREATE SEQUENCE PAYMENT_ID_SEQ

START WITH 10;