Question i

Table Structures:

Artist (A_id, Name, DOB, Nationality)
Owner (O_id, Name),
Introduced(O_id, O_new, I_id),
Paints (A_id, P_id),
Painting (P_id, Title, Acquisition_date, O_id, Pol_id, insurance_value),
In_Gallery (G_id, Room, P_id)
On_Loan (L_id, Exhibition_venue, P_id)

Used Method 1 in Lecture 5 to separate subtypes from Painting: Create separate relations for the super-type and sub-type entities. Post the identifiers from the super-type to the sub-type.

Question ii

```
create table Artist (
        A_id varchar2(10) constraint pk_artist primary key,
        Name varchar2(20),
        dob date,
        Nationality varchar2(20));

create table Owner (
        O_id varchar2(10) constraint pk_id_owner primary key,
        Name    varchar2(20));

create table Introduced(
        O_id varchar2(10) constraint fk_o references Owner(O_id),
        O_new varchar2(10) constraint fk_o2 references Owner(O_id),
        I_id varchar2(10) constraint pk_id_introduced primary key);

create table Painting (
        P_id varchar2(10) constraint pk_id_painting primary key,
        Title varchar2(20),
        Acquisition_date date,
        O_id varchar2(10),
        Pol_id varchar2(10),
        Insurance_value number(9, 2) constraint no_nl_val not null
                                Constraint check_val check(Insurance_value >0));

create table Paints (
        A_id varchar2(10) constraint fk_art references Artist(A_id),
        P_id varchar2(10) constraint fk_p references Painting(P_id));


create table In_Gallery(
        G_id varchar2(10) constraint pk_id_gallery primary key,
        Room varchar2(10),
        P_id varchar2(10) constraint fk_p_id references Painting(P_id));

create table On_Loan (
        L_id varchar2(10) constraint pk_id_loan primary key,
        Exhibition_venue varchar2(10),
        P_id varchar2(10) constraint fk_pid references Painting(P_id));
```

Question iii

```
Insert into Artist values('387_JS', 'Jim Smith', date'1978-06-15', 'British');

Insert into Artist values('387_BJ', 'Bill Jones', date'1992-01-16', 'British');

Insert into Artist values('387_JB', 'Joe Bloggs', date'2003-11-23', 'British');

Insert into Owner values('386_SS', 'Steve Smith');

Insert into Owner values('386_PP', 'Paul Peters');

Insert into Owner values('386_TJ', 'Tom Jerry');

Insert into Introduced values('386_TJ', '386_SS', 'Intro_1');

Insert into Introduced values('386_PP', '386_TJ', 'Intro_2');

Insert into Painting values('123_a', 'Scenery of Britain', date'2020-11-23', '386_PP', 'POL_345', 350);

Insert into Painting values('123_b', 'Scenery of Europe', date'2020-05-07', '386_SS', 'POL_344', 699);

Insert into Painting values('123_c', 'Scenery of Asia', date'2020-09-19', '386_TJ', 'POL_346', 900);

Insert into Painting values('123_d', 'Scenery of America', date'2020-03-04', '386_TJ', 'POL_347', 200);

Insert into Painting values('123_e', 'Scenery of Africa', date'2020-07-15', '386_SS', 'POL_349', 300);

Insert into Painting values('123_f', 'Scenery of India', date'2020-04-24', '386_PP', 'POL_350', 400);

Insert into Paints values('387_JS', '123_a');

Insert into Paints values('387_BJ', '123_b');

Insert into Paints values('387_JB', '123_c');

Insert into Paints values('387_JB', '123_d');

Insert into Paints values('387_JS', '123_e');

Insert into Paints values('387_BJ', '123_f');

Insert into In_Gallery values('G_2k', '24b', '123_a');

Insert into In_Gallery values('G_2l', '24c', '123_b');

Insert into In_Gallery values('G_2m', '24a', '123_c');

Insert into On_Loan values('L_1', 'London', '123_d');

Insert into On_Loan values('L_2', 'Manchester', '123_e');

Insert into On_Loan values('L_3', 'Liverpool', '123_f');
```

Question iv

i)
```
select Name, Count(*) Paintings
 from Artist, Paints
 where Artist.A_id = Paints.A_id
 group by Name;
```

This query allows for us to see how many paintings each artist has painted. Displays the name of the artist along with a count of the amount of paintings they each have painted.

This is the output:

```
NAME                PAINTINGS
------------------- ----------
Jim Smith                    2
Joe Bloggs                   2
Bill Jones                   2
```

ii)
```
select Pol_id, P_id
from Painting
where insurance_value > (select avg(insurance_value)
                                 from Painting);
```

This query allows for us to see which paintings have an insurance value greater than the average of all the paintings. Displays the paintings id along with the policy id.

This is the output:

```
POL_ID     P_ID
---------- ----------
POL_344    123_b
POL_346    123_c
```

iii)
```
select P_id, O_id, insurance_value
from Painting
where insurance_value > (select avg(insurance_value)
                                   from Painting P
                                   Where Painting.O_id = P.O_id);
```

This query prints the paintings with owners and insurance value where the value of the insurance is above the average value of the painting for that owner.

This is the output:

```
P_ID       O_ID       INSURANCE_VALUE
---------- ---------- ---------------
123_b      386_SS                 699
123_c      386_TJ                 900
123_f      386_PP                 400
```

iv)
```
select X.Name, N.Name as Person_Introduced
from Owner X
left join Introduced I
on X.O_id = I.O_id
left join Owner N
```

on N.O_id = I.O_new;

This is the output:

```
NAME                 PERSON_INTRODUCED
-------------------- --------------------
Paul Peters          Tom Jerry
Steve Smith
Tom Jerry            Steve Smith
```

Question v

This database design satisfies the criteria for 1NF, 2NF and 3NF. The strengths of this database design is it allows for minimal redundancy, reduces dependences and allows for securer data. This is due to the tables being broken down into smaller tables to ensure that they are stored more logically. A weakness of this design is the face that due to the method used, there are several more tables created when dealing with the super-type and sub-type tables. Another weakness is there is a possibility of having null values as in this case. For example, an owner needs to be introduced but they don't necessarily need to introduce someone else. This leaves the introduced table with the ability to have null values. This is just one example of where a value isn't required.