Introduction:

The assignment given to us is a free choice designing and developing of a database system for any arbitrary organization. The process will include following things:

1. Defining and rendering a solution for a problem through normalization process with respect to the required entities of the organization. For this Visual Paradigm, will be used for graphical re-presentation.
2. Oracle MYSQL will be used for implementation.
3. Other tools such as MS word, PDF, MS snipping tools… will be used for proper documentation.

User Stories:

The organization 'Excel Antique' deals with selling and buying of different types of antiques from around the world. Their buyers and sellers and both national and international. The problem is that their old paper based database system is too out of date to use in this growing and technical world. To stand with the standards of transaction throughout the scopes an efficient electronic database system is to be developed.

I went through all the paper based transaction of some years and collected potential entities with proper analysis. Many interviews were done with the concerned authorities so that they are friendly with what is being developed.

The following topics represents the phases of development with each step in designing and developing:

Initial Design

Entity Relationship Diagram (ER Diagram):

Entity Relationship diagrams facilitates database developers by providing them tools and mediums for creating a virtual system from the real-life scenario. The developers can make defined relationships between real life objects which are to be implemented in database system.

Diagrammatic tools and relationship types with integrated concepts are the main idea behind it. For doing this we have many

Software like visual Paradism (which is a UML based software).

With respect to the given scenario ER diagram for the company using 'Visual Paradism' was made.

The entity relationship Diagram which is fully normalized to third normal form is shown below: ***"the required image in following figure is uploaded separately for more accurate view"***

Fig 1.1 ER Diagram Representing the DB system of the company

Data Dictionary

It is a user Accessible System catalog which defines the type/behavior/number of data that should be hold by Entities of each entity types in a database. It is also known as metadata or system catalog

So, from the ER diagram (fig 1.1) we can describe the entity types included as:

Entity Types:	Attributes & their Descriptions
1.**item**	**Item_code(pk) integer(2)**
	Country_code(fk) integer(1)
	Manufacturer_code(fk) integer(3)
	Item_type_name(fk) varchar(20)

2.**country_of_origin**	**Country_code(pk) integer(1)**
	Country_name(unique) varchar(20)
	Country_religion_type varchar(20)
3.**manufacturer**	**Manufacturer_code(pk) integer(3)**
	Manufacturer_type varchar(20) domain(modern,classical,mythilogical)
	Manufacturer_name varchar(20)
4.**item_type**	**Item_type_name(pk) varchar(20)**
	Item_type_specification varchar(20)
	Item_type_size varchar(20) domain(small,medium,large)
5.**item_valuation**	**Valuer_code(pk)(fk) integer(2)**
	Item_code(pk)(fk) integer(2)
	Valuation_amount integer(6)
6.**valuer**	**Valuer_code(pk) integer(2)**
	Valuer_company_code(fk) integer(3)
	Valuer_name varchar(20)
7.**valuer_company**	**Valuer_company_code(pk) integer(3)**
	Company_code(fk) integer(1)
	Valuer_company_rating integer(1)
8.**owner_history**	**Owner_code(pk)(fk) integer(3)**
	Item_code(pk)(fk) integer(2)
	Start_date(pk) integer(5)
	End_date(pk) integer(5)
9.**company**	**Company_code(pk) integer(1)**
	Company_address varchar(20)
	Country_code(fk) integer (1)
	Company_name(unique) varchar(20)
10.**owner**	**Owner_code(pk) integer(3)**
	Owner_name varchar(20)
	Owner_purpose varchar(20)

Primary key:

It is the property of an attribute by which a occurrence cannot repeated and never can be null. It also creates indexing to numerical attributes.

Foreign key:

Foreign key is an attribute or combination of attributes, applied to create and link between 2 tables. It can be created by a foreign key constraint when we create or alter table.

Composite key:

It is defined as candidate key which consists of two or more attributes. Simply is the collection of two or more attributes being primary key.

Domain:

It is the set of defined allowable values that can go into an attribute of a table.

Implementation & Data

(1) Creation of all the tables:

All the created tables on the database system are given below with its descriptions in the form of snaps:

```
SQL> create table item(item_code number(2) primary key,
  2  country_code number(1),
  3  manufacturer_code number(3),
  4  item_tyoe_name varchar(20));

Table created.

SQL> desc item;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_CODE                                 NOT NULL NUMBER(2)
 COUNTRY_CODE                                       NUMBER(1)
 MANUFACTURER_CODE                                  NUMBER(3)
 ITEM_TYOE_NAME                                     VARCHAR2(20)

SQL>
```

This command creates a table named '**item**' with 'item_code ' as a primary key which uniquely defines each row.

```
SQL> create table country_of_origin(country_code number(1) primary key,
  2  country_name varchar(20) unique,
  3  country_religion_type varchar(20));

Table created.

SQL> desc country_of_origin;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COUNTRY_CODE                              NOT NULL NUMBER(1)
 COUNTRY_NAME                                       VARCHAR2(20)
 COUNTRY_RELIGION_TYPE                              VARCHAR2(20)
```

The command above is to create table named '**country_of_origin**', in which attribute country_name is unique type i.e. a country name cannot be repeated.

```
SQL> create table manufacturer(manufacturer_code number(3),
  2  manufacturer_type varchar(20),
  3  manufacturer_name varchar(20),
  4  check(manufacturer_type in('classical','mordern','mythological')));

Table created.

SQL> desc manufacturer_type;
ERROR:
ORA-04043: object manufacturer_type does not exist


SQL> desc manufacturer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MANUFACTURER_CODE                                  NUMBER(3)
 MANUFACTURER_TYPE                                  VARCHAR2(20)
 MANUFACTURER_NAME                                  VARCHAR2(20)
```

It creates table named '**manufacturer**' in which a domain is set into attribute 'manufacturer_type' by which it allows only (classical,myhtological,mordern) to enter.

```
SQL> create table item_type(item_type_name varchar(20) primary key,
  2  item_type_specification varchar(20),
  3  item_type_size varchar(20),
  4  check(item_type_size in('small','medium','large')));

Table created.

SQL> desc manufacturer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MANUFACTURER_CODE                                  NUMBER(3)
 MANUFACTURER_TYPE                                  VARCHAR2(20)
 MANUFACTURER_NAME                                  VARCHAR2(20)

SQL> desc item_type;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_TYPE_NAME                            NOT NULL VARCHAR2(20)
 ITEM_TYPE_SPECIFICATION                            VARCHAR2(20)
 ITEM_TYPE_SIZE                                     VARCHAR2(20)
```

It creates table '**item_type**' , attribute item_type_name is primary key, in item_type_size a domain is set which allows only (small,medium &large) to enter.

```
SQL> create table item_valuation(valuer_code number(2),
  2  item_code number(2),
  3  valuation_amount number(6),
  4  constraint pk_vit primary key(valuer_code,item_code));

Table created.

SQL> desc item_valuation
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 VALUER_CODE                               NOT NULL NUMBER(2)
 ITEM_CODE                                 NOT NULL NUMBER(2)
 VALUATION_AMOUNT                                   NUMBER(6)
```

'**Item_valuaton'** table is created where item_code & valuer_code are made composite key by which a occurrence of these two attributes combinely cannot be repeated.

```
SQL> create table valuer(valuer_code number(2) primary key,
  2  valuer_company_code number(3),
  3  valuer_name varchar(20));

Table created.

SQL> desc item_valuation;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 VALUER_CODE                               NOT NULL NUMBER(2)
 ITEM_CODE                                 NOT NULL NUMBER(2)
 VALUATION_AMOUNT                                   NUMBER(6)
```

'**valuer**' table created , valuer_code attribute as primary key.

```
SQL> create table valuer_company(valuer_company_code number(3) primary key,
  2  company_code number(1),
  3  valuer_company_rating number(1),
  4  check(valuer_company_rating in('1','2','3','4','5')));

Table created.

SQL> desc valuer_company;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 VALUER_COMPANY_CODE                       NOT NULL NUMBER(3)
 COMPANY_CODE                                       NUMBER(1)
 VALUER_COMPANY_RATING                              NUMBER(1)
```

'**Valuer_company'** table created , attribute valuer_company_code as primary key & a domain set in valuer_company_rating as (1,2,3,4,5) to be entered

```
SQL> create table owner_history(owner_code number(3),
  2  item_code number(2),
  3  start_date number(5),
  4  end_date number(5),
  5  constraint ois_pk primary key(owner_code,item_code,start_date));

Table created.

SQL> desc owner_history
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER_CODE                                NOT NULL NUMBER(3)
 ITEM_CODE                                 NOT NULL NUMBER(2)
 START_DATE                                NOT NULL NUMBER(5)
 END_DATE                                           NUMBER(5)
```

Creation of table '**owner_history**' where attributes owner_code,item_code,start_date all are made composite key so that a unique occurrence of each attribute in row do not repeat itself.

```
SQL> create table owner(owner_code number(3) primary key,owner_name varchar(20),
  2  owner_purpose varchar(20));

Table created.

SQL> desc owner;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER_CODE                                NOT NULL NUMBER(3)
 OWNER_NAME                                         VARCHAR2(20)
 OWNER_PURPOSE                                      VARCHAR2(20)
```

Table '**owner**' created with owner_code as primary key.

```
SQL> create table company(company_code number(1) primary key,
  2  address varchar(20),
  3  country_code number(1),company_name varchar(20) unique);

Table created.

SQL> desc company;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMPANY_CODE                              NOT NULL NUMBER(1)
 ADDRESS                                            VARCHAR2(20)
 COUNTRY_CODE                                       NUMBER(1)
 COMPANY_NAME                                       VARCHAR2(20)
```

Table '**company**' created with 'company_code ' as primary key & company_name as unique, so that company name is not repeated

Alterations of table with respect to ER diagram relationships:

```
SQL> alter table item add constraint cc_fk foreign key(country_code)
  2  references country_of_origin(country_code);

Table altered.
```

Country_code of '**country_of_origin**' made foreign key to '**item**'

```
SQL> alter table owner_history add constraint ic_fk foreign key(item_code)
  2  references item(item_code) on delete set null;

Table altered.
```

Item_code of **'item'** made foreign key to '**owner_history'** and on deletion from item_code the default value will be set null.

```
SQL> alter table item add constraint type_fk foreign key(item_type_name)
  2  references item_type(item_type_name);

Table altered.
```

Item_type_name of '**item_type'** made foreign key to '**item**'

```
SQL> alter table item add constraint mc_fk foreign key(manufacturer_code)
  2  references manufacturer(manufacturer_code);

Table altered.
```

Manufacturer_code of '**manufacturer**' made foreign key to '**item**'

```
SQL> alter table item_valuation add constraint icc_fk foreign key(item_code)
  2  references item(item_code);

Table altered.
```

item_code of '**item**' made foreign key to '**item_valuation'**

```
SQL> alter table item_valuation add constraint vc_fk foreign key(valuer_code)
  2  references valuer(valuer_code);

Table altered.
```

Valuer_code of '**valuer**' made foreign key to '**item_valuation'**

```
SQL> alter table valuer add constraint vcc_fk foreign key(valuer_company_code)
  2  references valuer_company(valuer_company_code);

Table altered.
```

Valuer_company_code of '**valuer_company'** made foreign key to '**valuer**'

```
SQL> alter table valuer_company add constraint ccc_fk foreign key(company_code)
  2  references company(company_code);

Table altered.
```

Company_code of '**company**' made foreign key to '**valuer_company'**

```
SQL> alter table company add constraint c2c_fk foreign key(country_code)
  2  references country_of_origin(country_code);

Table altered.
```

Country_code of '**country_of_origin'** made foreign key to '**company**'

```
SQL> alter table owner_history add constraint ooc_fk foreign key(owner_code)
  2  references owner(owner_code);

Table altered.
```

Owner_code of '**owner**' made foreign key to '**owner history**'

Now all the tables are normalized with the commands and are related according to the ER diagram.so, now we can efficiently insert the given/assumed data into the tables:

(2) Insertion to tables:

Insertion scripts used and all the completed tables are demonstrated below:

```
SQL> insert into country_of_origin values(&country_code
  2  ,&country_name,&country_religion_type);
```

Allowing to enter value to '**country_of_origin**'

```
SQL> select *from country_of_origin;

COUNTRY_CODE COUNTRY_NAME          COUNTRY_RELIGION_TYP
------------ --------------------- --------------------
           1 china                 buddhism
           2 germany               christian orthodox
           3 england               christian
           4 nepal                 hinduism
           5 australia             christian
           6 japan                 budhism
           7 saudi arabia          muslim

7 rows selected.
```

Entered values in '**country_of_origin**'

```
SQL> insert into manufacturer values(&manufacturer_code,
  2  &manufacturer_type,&manufacturer_name);
```

Allowing to enter value to '**manufacturer**'

```
SQL> select *from manufacturer;

MANUFACTURER_CODE MANUFACTURER_TYPE    MANUFACTURER_NAME
----------------- -------------------- --------------------
              123 mordern              ming dynasty
              657 mythological         heil of berlin
              379 mythological         fitzgerald of hull
              900 classical            aris of nurenberg
```

Entered values in '**manufacturer**'

```
SQL> insert into item_type values(&item_type_name,&item_type_specification,
  2  &item_type_size);
```

Allowing to enter value to '**item_type**'

```
SQL> select *from item_type;

ITEM_TYPE_NAME       ITEM_TYPE_SPECIFICAT ITEM_TYPE_SIZE
-------------------- -------------------- --------------------
ming vase            ceramics             small
2flintlock pistols   firearms             medium
fowling musket       firearms             large
clockwork bird       toys and automata    small
mechanical ship      toys and automata    large
```

Entered values in '**item_type**'

```
SQL> insert into item values(&item_code,&country_code,&manufacturer_code,
  2  &item_type_name);
```

Allowing to enter value to '**item**'

```
SQL> select *from item;

 ITEM_CODE COUNTRY_CODE MANUFACTURER_CODE ITEM_TYPE_NAME
---------- ------------ ------------------ --------------------
        11            1                123 ming vase
        46            2                657 2flintlock pistols
        99            3                379 fowling musket
         3            2                900 clockwork bird
        67            2                900 mechanical ship
```

Entered values in '**item**'

(4)Insertion to tables:

```
SQL> insert into company values(&company_code,&address,
  2  &country_code,&company_name);
```

Allowing to enter value to '**company**'

```
COMPANY_CODE ADDRESS              COUNTRY_CODE COMPANY_NAME
------------ -------------------- ------------ --------------------
           1 beijing                         1 hakuwa
           2 berlin                          2 seldenvian
           3 london                          3 little & associates
           4 devon                           3 hatchets
           5 surrey                          3 webb
           6 notts                           3 christophers
           7 pequolia                        2 explorea dent
           8 denind                          2 loyalitia comp
           9 kathmandu                       4 hamro antiques

9 rows selected.
```

Entered values in '**company**'

```
SQL> insert into valuer_company values(&valuer_company_code,
  2  &company_code,&valuer_company_rating);
```

Allowing to enter value to '**valuer_company**'

```
VALUER_COMPANY_CODE COMPANY_CODE VALUER_COMPANY_RATING
------------------- ------------ ----------------------
                456            3                      2
                457            4                      3
                458            5                      4
                459            6                      5
                556            1                      4
                557            2                      3
                558            7                      2
                559            8                      5
                656            9                      5

9 rows selected.
```

Entered values in '**valuer_company**'

```
SQL> insert into valuer values(&valuer_code,&valuer_company_code,
  2  &valuer_name);
```

Allowing to enter value to '**valuer**'

```
VALUER_CODE VALUER_COMPANY_CODE VALUER_NAME
----------- ------------------- --------------------
         23                 456 joan little
         24                 457 raymond delray
         25                 458 granville reed
         26                 459 helen smith
         27                 556 xing chao
         28                 557 johan pehars
         29                 558 petereni gebes
         30                 559 jonathsn herty

8 rows selected.
```

Entered values in '**valuer**'

(3)Insertion to tables:

```
SQL> insert into item_valuation values(&valuer_code,
  2  &item_code,&valuation_amount);
```

Allowing to enter value in '**item_valuation**'

```
VALUER_CODE  ITEM_CODE VALUATION_AMOUNT
----------- ---------- -----------------
         23         99              3000
         24         99              3500
         25         99              3000
         26         99              4000
         27         11              5000
         28         11              2500
         29         11              6000
         30         11              3000
         24         46              4500
         26         46              2500
         23         46              6500

VALUER_CODE  ITEM_CODE VALUATION_AMOUNT
----------- ---------- -----------------
         30         46              5500
         28          3              1500
         27          3              8000
         29          3              4500
         24          3              7500
         30         67              3000
         24         67              8500
         29         67              2000
         28         67              5500

20 rows selected.
```

Entered values in '**item valuation**'

Query

(1)Query1

For the selection of **items and all the valuations** accordingly the following command wa executed:

```
SQL> select item.item_type_name,item_valuation.valuation_amount,
  2  item_valuation.valuer_code,item_code
  3  from item join item_valuation using(item_code);

ITEM_TYPE_NAME       VALUATION_AMOUNT VALUER_CODE  ITEM_CODE
-------------------- ---------------- ----------- ----------
clockwork bird                 7500          24          3
clockwork bird                 4500          29          3
clockwork bird                 8000          27          3
clockwork bird                 1500          28          3
ming vase                      5000          27         11
ming vase                      2500          28         11
ming vase                      3000          30         11
ming vase                      6000          29         11
2flintlock pistols             4500          24         46
2flintlock pistols             6500          23         46
2flintlock pistols             2500          26         46

ITEM_TYPE_NAME       VALUATION_AMOUNT VALUER_CODE  ITEM_CODE
-------------------- ---------------- ----------- ----------
2flintlock pistols             5500          30         46
mechanical ship                2000          29         67
mechanical ship                5500          28         67
mechanical ship                8500          24         67
mechanical ship                3000          30         67
fowling musket                 3000          23         99
fowling musket                 3500          24         99
fowling musket                 3000          25         99
fowling musket                 4000          26         99

20 rows selected.
```

The above command creates a view selecting item_type_name from tabel '**item**' & valuation_amount,valuer_code from **item_valuation**. The item_code is the attribute which is common in both. This is a dynamic creation of view by joining 2 tables to get precise information. We can see that which items are valued by which owners at which valuation amount.

(2)Query2

For the selection of all **items according to its type**, the following command was executed

```
SQL> select item.item_code,item.country_code,
  2  item_type.item_type_specification,item_type_name
  3  from item join item_type using(item_type_name);

 ITEM_CODE COUNTRY_CODE ITEM_TYPE_SPECIFICAT ITEM_TYPE_NAME
---------- ------------ -------------------- --------------------
        46            2 firearms             2flintlock pistols
         3            2 toys and automata    clockwork bird
        99            3 firearms             fowling musket
        67            2 toys and automata    mechanical ship
        11            1 ceramics             ming vase
```

Creation of a view from table '**item**' and '**item_type'** selecting the attributes item_code,country_code from table '**item**',item_type_specification from table item_type. The item_type_name is the common attribute for both the table. It gives a specified information from the two tables. We can observe that which item has what type of specification with its country of origin.

Assessment:

The company basically requires a DB system which can keep information of the items they stock and sell.

This system which I have designed is more integrated than the method used in the sample records given by the company . Firstly, the table item by which they can easily know name of items, its manufacturer's code, codes of country origination and separately keeping the information of country, manufacturer and item type making it a more classified way of keeping records.

When the items will be valued the records will be in different table 'item_valuation' where there are code of item, valuer code, by which a valuer cannot do valuation to an item twice, however a valuer can do valuations for one or many items. Each item must be valued at least 3 times as shown in above valuations listing.

There might be many owners of the item for certain period of time, but a item cannot be owned by 2 owners simultaneously. Every item is owned at a unique time with respect to end time of ownership of an owner. So, an item can be owned by one owner for certain time and then it can be owned by another owner at the end of ownership of the first owner. This situation can be repeated number of times for a single item, as there can be many owners over many periods of time of an item. This is a very efficient way of keeping owners history which is a vital information for the company.

There is addition of a new entity type named 'company' in which a company can keep record of different companies related to this field of business not necessarily being valuer company, but by this we can have a separate set of information of valuer companies and other companies which may be future valuer company. Hence, the fully normalized database system has been developed for keeping records efficiently by the company.