

Scenario Modelling and Database Implementation

CE205 Assignment 1 2017-18

Cieran Almond (1604959)

1. Choice of Entities, Attributes and Relationships

How you determined the entities to model, the attributes and the relationships between entities. Why you chose the entities you did.

1. Identify four entities which you would like to model. One of them must be 'camera' and the others must be linked to 'camera' in some way. I chose Lens because every camera has a lens, and additional lenses are very popular, especially on DSLR cameras. So, in my opinion, its an obvious entity that will be related to camera. I chose memory card because every camera uses a memory card to save pictures onto, mainly for removable storage of pictures, whether it be a digital camera or DSLR camera, and so again, would relate to the camera entity. Finally I chose manufacturer because every product has a manufacturer, so I could relate all the entities to the manufacturer entity. I will be using lens_name, manufacturer_name and memory_card_name as primary keys to link the entities to the Camera.

Camera(camera_name, camera_brand, camera_make, camera_price, camera_stock, camera_megapixel);

Len (lense_name, lense_make, lense_size, lense_price);

Manufacturer(manufacturer_name, manufacturer_address, manufacturer_phone, manufacturer_registration, manufacturer_website);

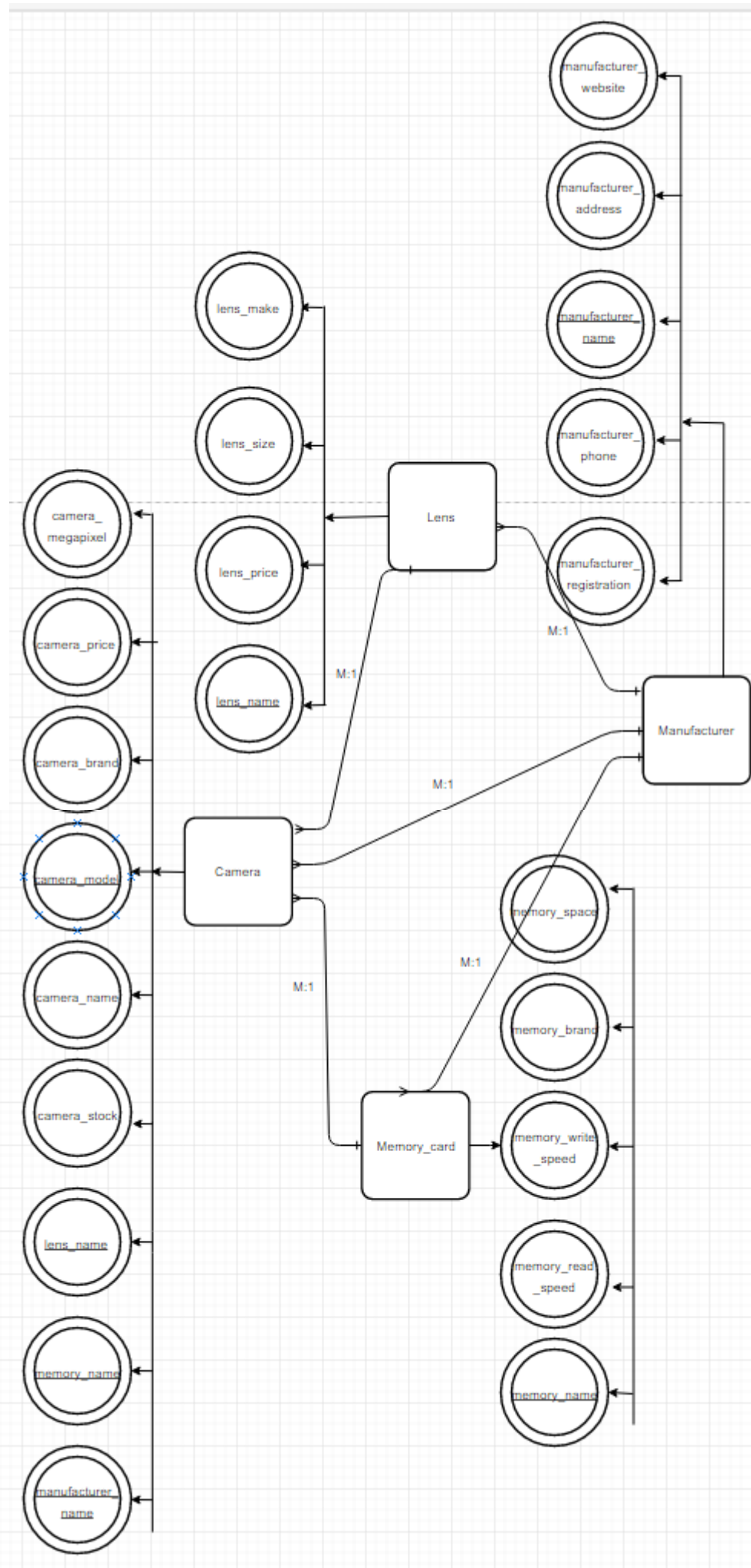
Memory_card(memory_card_name, memory_card_space, memory_card_brand, memory_write_speed, memory_read_speed);



2. Conversion of Model to 3NF

My diagram is already in third normal form. Firstly, the table fits the condition for 1NF since there are no duplicate entries, and each table has a primary key (memory_name, lens_name, manufacturer_name and camera_model). There are no subsets of data that apply to multiple rows, therefore it fulfils 2NF as well. Finally, all of the columns are dependent on the primary key, eliminating the functional dependencies on non-key fields and so the table is therefore already in 3NF. My table uses relations between camera, manufacturer, memory card and lens, and so through these relations, has a few null attributes in the Camera table because of these keys. This is an inefficient process that in hindsight I would change in the future, as the purpose of putting tables into 3NF is to reduce data redundancy and maintain integrity to the database.

See table below:



3. Sample Queries and Output

```
Camera.sql
1 DROP DATABASE IF EXISTS Camera;
2 CREATE DATABASE Camera;
3 USE Camera;
4
5 #created camera table
6
7 CREATE TABLE Camera (
8     camera_price double,
9     camera_brand VarChar (50),
10    camera_model VarChar (50),
11    camera_name VarChar (50),
12    camera_stock int,
13    camera_megapixel int,
14    lens_name VarChar (50),
15    memory_name VarChar (50),
16    manufacturer_name VarChar (50),
17    PRIMARY KEY (camera_model)
18 );
19
20 # added cameras to database
21
22 INSERT INTO Camera VALUES (949.00, "Pentax", "B01HMH0JJI", "K-70 DSLR", 34, 24, lens_name, memory_name, n
23 INSERT INTO Camera VALUES (297.00, "Canon", "B01CUGJZPU", "EOS 1300D DSLR", 78, 15, lens_name, memory_nam
24 INSERT INTO Camera VALUES (1017.93, "Pentax", "B00YX2QYPG", "K-3II DSLR", 152, 20, lens_name, memory_name
25 INSERT INTO Camera VALUES (550.00, "Canon", "B00T9OEYXM", "EOS 750D DSLR", 124, 24, lens_name, memory_nam
26 INSERT INTO Camera VALUES (383.90, "Nikon", "B01KOADMG4", "D3400 DSLR", 112, 18, lens_name, memory_name,
27 INSERT INTO Camera VALUES (279.99, "Nikon", "B01M3RFM2F", "D60 Digital DSLR", 32, 16, lens_name, memory_r
28
29 #created lens table
30
31 CREATE TABLE Lens (
32     lens_make VarChar (50),
33     lens_size VarChar (50),
34     lens_price double,
35     lens_name VarChar (50) REFERENCES Camera(lens_name),
36     PRIMARY KEY (lens_name)
37 );
38
39 # added lenses to database
40
41 INSERT INTO Lens VALUES ("Canon", "75-300mm", 115.97, "EF III Lens");
42 INSERT INTO Lens VALUES ("Canon", "50mm", 106.00, "IF-S IS STM Lens");
43 INSERT INTO Lens VALUES ("Canon", "24mm", 134.00, "EF-S STM Lens");
44 INSERT INTO Lens VALUES ("Nikon", "55-300mm", 216.99, "AF-S VR Lens");
45 INSERT INTO Lens VALUES ("Nikon", "18-140mm", 259.00, "AF-S ED VR Lens");
46 INSERT INTO Lens VALUES ("Nikon", "35mm", 159.00, "AF-S DX Lens");
47 INSERT INTO Lens VALUES ("Pentax", "35mm", 118.49, "SMC AL Lens");
48 INSERT INTO Lens VALUES ("Pentax", "50-200mm", 169.00, "SMC ED WR Lens");
49
50 #created memory card table
51
52 CREATE TABLE Memory_card (
53     memory_space int,
54     memory_brand VarChar (50),
55     memory_write_speed double,
56     memory_read_speed double,
57     memory_name VarChar (50) REFERENCES Camera(memory_name),
58     PRIMARY KEY (memory_name)
59 );
60
61 # added memory cards to database
62
63 INSERT INTO Memory_card VALUES (32, "SanDisk", 15, 80, "SanDisk Ultra 32 GB" );
64 INSERT INTO Memory_card VALUES (64, "Lexar", 20, 95, "Lexar 64GB Professional" );
65 INSERT INTO Memory_card VALUES (64, "Samsung", 20, 95, "Samsung 64GB Evo Plus" );
66 INSERT INTO Memory_card VALUES (2, "Fujifilm", 2, 5, "Fujifilm 2GB SD Memory Card" );
67 INSERT INTO Memory_card VALUES (32, "Integral", 10, 40, "Integral UltimaPro 32GB MicroSDHC" );
68 INSERT INTO Memory_card VALUES (64, "Toshiba", 25, 90, "Toshiba Exceria M302 64GB Micro SD" );
69
70 #created manufacturer table
71
72 CREATE TABLE Manufacturer (
73     manufacturer_address VarChar (100),
74     manufacturer_name VarChar (50) REFERENCES Camera(manufacturer_name),
75     manufacturer_phone double,
76     manufacturer_registration int,
77     manufacturer_website VarChar (50),
78     PRIMARY KEY (manufacturer_name)
79 );
80
81 # added manufacturers to database
82
83 INSERT INTO Manufacturer VALUES ("3 The Square, Stockley Park, Uxbridge, Middlesex, United Kingdom, UB11
84 INSERT INTO Manufacturer VALUES ("380 Richmond Rd, Kingston upon Thames, KT2 5DB", "Nikon UK Ltd", 020854
85 INSERT INTO Manufacturer VALUES ("Customer Service, PO Box 81226, Seattle, WA, 98108-1226", "Amazon", 888
86 INSERT INTO Manufacturer VALUES ("1st Floor, 3 Furzeground Way, Stockley Park, Uxbridge, Middlesex, UB11
87 INSERT INTO Manufacturer VALUES ("951 SanDisk Drive, Milpitas, CA 95035-7933", "Sandisk", 4088011000, 594
88 INSERT INTO Manufacturer VALUES ("Samsung Electronics (UK) Ltd, Samsung House 1000 Hillwood Drive Chert
89
90 # query database
91
92 #query 1
93
94 SELECT camera_brand, camera_name, camera_stock, camera_megapixel, camera_model, camera_price
95 FROM Camera
96 WHERE camera_price > 400
97 ORDER BY camera_name, camera_name DESC;
98
99 #query 2
100
101
102
103
104
105
```

Example of the database being created, and adding values to them. The database was written in Notepad++ and was executed using a .bat file to generate a log textpad, as seen below. This screenshot shows the tables being created and the data being successfully entered into the database.

```

log - Notepad
File Edit Format View Help
-----
DROP DATABASE IF EXISTS Camera
-----
Query OK, 4 rows affected (0.27 sec)

-----
CREATE DATABASE Camera
-----
Query OK, 1 row affected (0.02 sec)

-----
CREATE TABLE Camera (
  camera_price double,
  camera_brand VarChar (50),
  camera_model VarChar (50),
  camera_name VarChar (50),
  camera_stock int,
  camera_megapixel int,
  lens_name varChar (50),
  memory_name varChar (50),
  manufacturer_name varChar (50),
  PRIMARY KEY (camera_model)
)
-----
Query OK, 0 rows affected (0.11 sec)

-----
INSERT INTO Camera VALUES (949.00, "Pentax", "B01HMH0JI", "K-70 DSLR", 34, 24, lens_name, memory_name, manufacturer_name)
-----
Query OK, 1 row affected (0.03 sec)

```

SQL queries:

```

# query database

#query 1

SELECT camera_brand, camera_name, camera_stock, camera_megapixel, camera_model, camera_price
FROM Camera
WHERE camera_price > 400
ORDER BY camera_name, camera_name DESC;

```

1. This query I chose to demonstrate simple select searches on the camera table, given the condition of the price of the cameras and to sort it by camera name in descending order.

camera_brand	camera_name	camera_stock	camera_megapixel	camera_model	camera_price
Canon	EOS 750D DSLR	124	24	B00T90EYXM	550
Pentax	K-3II DSLR	152	20	B00YX2QYPG	1017.93
Pentax	K-70 DSLR	34	24	B01HMH0JI	949

3 rows in set (0.00 sec)

#query 2

```
SELECT manufacturer_name, manufacturer_address, manufacturer_phone, manufacturer_website
FROM Manufacturer
WHERE manufacturer_name LIKE '%Amazon%';
```

2. This query I chose to demonstrate sorting of attributes by searching for a specific string in this query, just to demonstrate how to access manufacturer data from the database.

manufacturer_name	manufacturer_address	manufacturer_phone	manufacturer_website
Amazon	Customer Service, PO Box 81226, Seattle, WA, 98108-1226	8882803321	www.Amazon.com

1 row in set (0.00 sec)

```
SELECT *
FROM Lens
WHERE lens_price BETWEEN 10 AND 300;
```

3. This query I chose because it demonstrates how to query a set of numerical data providing there is condition of price between 10 and 300, returning data between those values.

lens_make	lens_size	lens_price	lens_name
Nikon	35mm	159	AF-S DX Lens
Nikon	18-140mm	259	AF-S ED VR Lens
Nikon	55-300mm	216.99	AF-S VR Lens
Canon	75-300mm	115.97	EF III Lens
Canon	24mm	134	EF-S STM Lens
Canon	50mm	106	IF-S IS STM Lens
Pentax	35mm	118.49	SMC AL Lens
Pentax	50-200mm	169	SMC ED WR Lens

8 rows in set (0.00 sec)

```
SELECT Camera.camera_name, Camera.camera_model, Lens.lens_make, Lens.lens_size
FROM Camera
INNER JOIN Lens ON Camera.lens_name = Lens.lens_name
WHERE Camera.lens_name LIKE "%Canon%";
```

4. I chose this query to demonstrate that the keys work, and that all the database is interlinked via foreign keys, the purpose of the inner join is to link Camera and Lens together. The final condition being that it searches a particular string.