

**ECU178 Computer Science:
220CT Data and Information Retrieval
Coursework**

Due on March 13th 2015

Robert Rigler : 4939377

Contents

Task 1 Database Design	3
Activity 1: First Normal Form	3
Step 1: Identify Redundancy	3
Step 2: Create New Entity	3
Step 3: Relationships and Keys	3
1NF : Diagram	3
1NF : Data	4
Activity 2: Second Normal Form	5
Step 1: Testing each attribute.	5
Step 2: Entity, Relationship and Key	5
Entities in Second Normal Form: Diagram	5
Entities in Second Normal Form: With data	6
Activity 3: Third Normal Form	7
Step 1: Testing for Transitive Dependency	7
Step 2: Entity, Relationship and Key	7
Entities in Third Normal Form: Diagram	7
Entities in Third Normal Form: With Data	8
Activity 4: ER Diagram	9
Task 2: Database Development	9
1: Creating The Database Tables	9
Aircraft	9
Airline	10
Purchase_Order	11
Ordered_Aircraft	12

Task 1 Database Design

This task in Database Design consists of four activities. The first three involve normalising the given data to third normal form, and the fourth is to produce an Entity Relationship diagram of the normalised relations. For each activity I will give a detailed step by step explanation of how I completed each activity.

Activity 1: First Normal Form

To put this data into First Normal Form (1NF), I need to:

1. Identify any repeating redundant data and remove it from the current Entity
2. Place the data into a new Entity
3. Create a relationship with a primary key from one Entity as a foreign key in the other.

Step 1: Identify Redundancy

On inspecting the data, I can see that there are multiple instances of repeating data.

Orders' ID: *CON-2237*, *CON-2356* and *CON-1234* all have repeating data entries for fields: *Equipment*, *Qty*, and *Unit Price*.

Step 2: Create New Entity

Removing the *Equipment*, *Qty*, and *Unit Price* fields and placing them in a new entity, leaves me with two entities as shown below.

Step 3: Relationships and Keys

To complete the First Normal Form, a relationship needs to be created between the entities.

I created the relationship by including the *Order ID* attribute as a foreign key in the *ItemOrder* entity.

Order ID is used as a Primary Key for the *Order* entity. In the *ItemOrder* entity, no one attribute can be used to uniquely identify a single record. For this reason, I have created a concatenated key using the attributes *Equipment* and *Order ID*. The concatenated key can now be used to uniquely identify each record.

1NF : Diagram

Order	(<u>Order ID</u>	ItemOrder	(<u>*Order ID</u>
	Supplier ID		<u>Equipment</u>
	Client Name		Qty)
	Client Address		Unit Price)
	Date		
	Total Price)		

1NF : Data*Order entity*

Order ID	Supplier ID	Client Name	Client Address	Date	Total Price
CON-2237	168	Coventry Building Services Ltd	Units 2-4, Binley Industrial Estate, CV3 2WL	14-Dec-14	£99.00
CON-3664	527	Allied Construction Ltd	34, Lythalls La Industrial Estate, CV6 6RG	16-Jan-15	£36.00
CON-2356	169	Rioh Builds Ltd	Unit 12, Stoneleigh Park, CV8 2UV	12-Feb-15	£280.00
CON-1234	032	Grand Design Ltd	32-34, Bilton Industrial Estate, CV3 5YB	16-Apr-15	£23.00

ItemOrder entity

Order ID	Equipment	Qty	Unit Price
CON-2237	Butterfly Valve	2	£5.00
CON-2237	3/4" Locknut	6	£1.50
CON-2237	Sch 40 Blk Pipe	4	£20.00
CON-3664	Thin Stranded Copper Wire	6	£6.00
CON-2356	Sch 40 Blk Pipe	3	£20.00
CON-2356	4x8x3/4 Cos Plywood	2	£10.00
CON-2356	3/4" EMT	2	£50.00
CON-2356	Duplex Ivy Rec	1	£100.00
CON-1234	Sch 40 Blk Pipe	1	£20.00
CON-1234	3/4" Locknut	2	£1.50

Activity 2: Second Normal Form

To put this data into the Second Normal Form(2NF) I need to ensure that the attributes are completely dependant on the primary key, i.e., that no attribute is only dependant on one part of the primary key.

This can be done in two steps:

1. Test each attribute for complete dependency on the primary key.
2. Remove any partially dependent attributes to a new entity and assign a primary key.

For this particular set of data, the *Order* entity does not have a concatenated key and therefore is already in the second normal form.

Step 1: Testing each attribute.

Primary Key	Attribute	Functionally Dependant?
Order ID, Equipment	Qty	Yes, dependant on both
Order ID, Equipment	Unit Price	No, dependant on Equipment only

Step 2: Entity, Relationship and Key

From my testing, I found that the attribute *Unit Price* is not functionally dependant as it is only dependent on *Equipment*, but not *Order ID*.

I moved *Equipment* and *Unit Price* into a new entity called *Item* and made *Equipment* the primary key.

I created a relationship between the *Item* and *ItemOrder* entities by repeating the *Equipment* attribute in *ItemOrder* as a foreign key.

Entities in Second Normal Form: Diagram

Order	(<u>Order ID</u>	Item	(<u>Equipment</u>	ItemOrder	(* <u>Order ID</u>
Supplier ID		Unit Price)		* <u>Equipment</u>	
Client Name				Qty)	
Client Address					
Date					
Total Price)					

Entities in Second Normal Form: With data*Order entity*

Order ID	Supplier ID	Client Name	Client Address	Date	Total Price
CON-2237	168	Coventry Building Services Ltd	Units 2-4, Binley Industrial Estate, CV3 2WL	14-Dec-14	£99.00
CON-3664	527	Allied Construction Ltd	34, Lythalls La Industrial Estate, CV6 6RG	16-Jan-15	£36.00
CON-2356	169	Rioh Builds Ltd	Unit 12, Stoneleigh Park, CV8 2UV	12-Feb-15	£280.00
CON-1234	032	Grand Design Ltd	32-34, Bilton Industrial Estate, CV3 5YB	16-Apr-15	£23.00

Item entity

Equipment	Unit Price
3/4" EMT	50.00
3/4" Locknut	1.50
4x8x3/4 Cos Plywood	10.00
Butterfly Valve	5.00
Duplex Ivy Rec	100.00
Sch 40 Blk Pipe	20.00
Thin Stranded Copper Wire	6.00

ItemOrder entity

Order ID	Equipment	Qty	Unit Price
CON-2237	Butterfly Valve	2	£5.00
CON-2237	3/4" Locknut	6	£1.50
CON-2237	Sch 40 Blk Pipe	4	£20.00
CON-3664	Thin Stranded Copper Wire	6	£6.00
CON-2356	Sch 40 Blk Pipe	3	£20.00
CON-2356	4x8x3/4 Cos Plywood	2	£10.00
CON-2356	3/4" EMT	2	£50.00
CON-2356	Duplex Ivy Rec	1	£100.00
CON-1234	Sch 40 Blk Pipe	1	£20.00
CON-1234	3/4" Locknut	2	£1.50

Activity 3: Third Normal Form

To place the data into Third Normal Form 3NF I need to ensure that all attributes are only dependent on the Primary Key, and not Non-Key Attributes. This can be achieved in two steps:

1. Test each attribute for dependency on the primary key.
2. Remove all transitive dependencies to a new entity with the correct primary key and relationship.

For this particular set of data, both the *Item* and *ItemOrder* entities are already in the Third Normal Form.

Step 1: Testing for Transitive Dependency

Order Entity

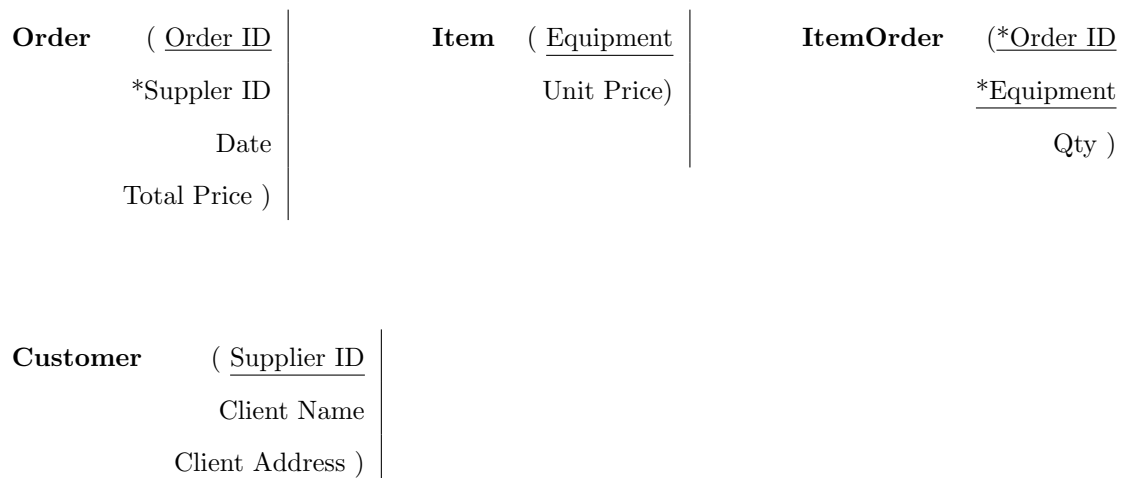
Primary Key	Attribute	Transitive Dependency?
Order ID	Supplier ID	Yes: Supplier ID can be found if we know Client Name or Address
Order ID	Client Name	Yes: Client Name can be found if we know Supplier ID or Address
Order ID	Client Address	Yes: Client Address can be found if we know Client Name or Supplier ID
Order ID	Date	No: Only dependent on Primary Key
Order ID	Total Cost	No: Only dependent on Primary Key

Using this table I have identified that *Supplier ID*, *Client Name*, and *Client Address* all have transitive dependencies, and need to be moved to a new entity.

Step 2: Entity, Relationship and Key

I created a new entity called *Customer*, and moved the three attributes with transitive dependencies into it. I then made *Supplier ID* the Primary Key and created a relationship between the *Customer* and *Order* entities by repeating the *Supplier ID* attribute as a foreign key in the *Order* entity.

Entities in Third Normal Form: Diagram



Entities in Third Normal Form: With Data

Order entity

Order ID	Supplier ID	Date	Total Price
CON-1234	032	16/04/2015	23.00
CON-2237	168	14/12/2014	99.00
CON-2356	169	12/02/2015	280.00
CON-3664	527	16/01/2015	36.00

Item entity

Equipment	Unit Price
3/4" EMT	50.00
3/4" Locknut	1.50
4x8x3/4 Cos Plywood	10.00
Butterfly Valve	5.00
Duplex Ivy Rec	100.00
Sch 40 Blk Pipe	20.00
Thin Stranded Copper Wire	6.00

ItemOrder entity

Order ID	Equipment	Qty
CON-1234	3/4" Locknut	2
CON-1234	Sch 40 Blk Pipe	1
CON-2237	3/4" Locknut	6
CON-2237	Butterfly Valve	2
CON-2237	Sch 40 Blk Pipe	4
CON-2356	3/4" EMT	2
CON-2356	4x8x3/4 Cos Plywood	2
CON-2356	Duplex Ivy Rec	1
CON-2356	Sch 40 Blk Pipe	3
CON-3664	Thin Stranded Copper Wire	6

Customer entity

Supplier ID	Client Name	Client Address
032	Grand Design Ltd	Unit 12, Stoneleigh Park, CV8 2UV
168	Coventry Building Services Ltd	Units 2-4, Binley Industrial Estate, CV3 2WL
169	Ricoh Builds Ltd	34, Lythalls La Industrial Estate, CV6 6RG
527	Allied Construction Ltd	32-34, Bilton Industrial Estate, CV3 5YB

Activity 4: ER Diagram

Task 2: Database Development

For this task I will provide the SQL statement I used to complete each question activity, I will then explain each part of the SQL statement, and give screenshots of before and after the statement was executed (if applicable).

Each SQL statement was written and tested using ORACLE 11g Express Edition and ORACLE Application Express.

1: Creating The Database Tables

To begin I created the two tables which have no dependency on any other table: *Aircraft* and *Airline*

Aircraft

Listing 1: CREATE AIRCRAFT

```
1 CREATE TABLE Aircraft
2 (   aircraft_code VARCHAR2(5) PRIMARY KEY,
3     aircraft_type VARCHAR2(30) NOT NULL,
4     aircraft_price NUMBER(11,2) NOT NULL,
5 );
```

The statement begins with '*CREATE TABLE Aircraft*' which will create a table called *aircraft*. Inside the brackets the three fields, their data types, and any constraints are listed. Below is a table which will explain why I chose these data types and constraints for each field.

Identifier	Data-Type	Constraint	Explanation
aircraft_code	VARCHAR2(5)	PRIMARY KEY	All aircraft_codes start with 'C' and are followed by up to four numerical digits. This field will be used as the Primary Key of the table.
aircraft_type	VARCHAR(30)	NOT NULL	Contains a combination of alphanumeric characters of up to 30 characters in length. This field cannot be left empty.
aircraft_price	NUMBER(11,2)	NOT NULL	Stores large numeric values, with a precision of 11 and a scale of 2. This allows for prices up to 99 Thousand Million and two decimal places. This field cannot be left empty.

Airline

Listing 2: CREATE AIRLINE

```
1 CREATE TABLE Airline
2 (   airline_code CHAR(4) PRIMARY KEY,
3     airline_name VARCHAR2(20) NOT NULL,
4     airline_address VARCHAR2(60) NOT NULL,
5     airline_city VARCHAR2(15) NOT NULL,
6     airline_country VARCHAR2(15) NOT NULL
7 );
```

Identifier	Data-Type	Constraint	Explanation
airline_code	CHAR(4)	PRIMARY KEY	A combination of 4 alphanumeric characters. airline_code is also the Primary Key
airline_name	VARCHAR2(20)	NOT NULL	This field allows a variable length of characters, up to a maximum of 20. This field cannot be left empty.
airline_address	VARCHAR2(60)	NOT NULL	This field allows for a combination of alphanumeric characters up to a maximum length of 60. This field cannot be left empty.
airline_city	VARCHAR2(15)	NOT NULL	This field allows a combination of alphanumeric characters up to a maximum length of 15. This field cannot be left empty.
airline_country	VARCHAR2(15)	NOT NULL	This field allows for a combination of alphanumeric characters up to a maximum length of 15. This field cannot be left empty.

Purchase_Order

Listing 3: CREATE PURCHASE_ORDER

```
1 CREATE TABLE purchase_order
2 ( purchase_order_no NUMBER(3,0) PRIMARY KEY ,
3   airline_code CHAR(4) NOT NULL,
4   date_of_purchase DATE NOT NULL,
5   CONSTRAINT FK_1 FOREIGN KEY (airline_code) REFERENCES airline(airline_code)
6 );
```

Identifier	Data-Type	Constraint	Explanation
purchase_order_no	NUMBER(3,0)	PRIMARY KEY	This field allows for numeric input with a precision of 3 and a scale of 0, this allows values in the range of 001 to 999. This field also acts as the Primary Key.
airline_code	CHAR(4)	NOT NULL, FOREIGN KEY	A combination of 4 alphanumeric characters. This field is a foreign key; Referencing the <i>airline_code</i> field from the <i>Airline</i> table.
date_of_purchase	DATE	NOT NULL	

Ordered_Aircraft

Listing 4: CREATE ORDERED_AIRCRAFT

```
1 CREATE TABLE ordered_aircraft
2 (
3 purchase_order_no NUMBER(3,0) NOT NULL,
4 aircraft_code VARCHAR2(5) NOT NULL,
5 aircraft_quantity NUMBER(3,0) NOT NULL,
6 CONSTRAINT PK_ORDERCODE PRIMARY KEY(purchase_order_no, aircraft_code),
7 CONSTRAINT FK_PO FOREIGN KEY (purchase_order_no)
8 REFERENCES purchase_order(purchase_order_no);
9 CONSTRAINT FK_AC FOREIGN KEY (aircraft_code)
10 REFERENCES aircraft(aircraft_code)
11 );
```