Normalization Exercise #1 Solution

1NF

(Repeating & Multivalued)

SID	CID	S_name	C_name	Grade	Faculty	F_phone
1	IS318	Adams	Database	A	Howser	60192
1	IS301	Adams	Program	B	Langley	45869
2	IS318	Jones	Database	A	Howser	60192
3	IS318	Smith	Database	B	Howser	60192
4	IS301	Baker	Program	A	Langley	45869
4	IS318	Baker	Database	B	Howser	60192

2NF

(Partial Dependence)

SID and CID → Grade

SID	CID	Grade
1	IS318	A
1	IS301	B
2	IS318	A
3	IS318	B
4	IS301	A
4	IS318	B

Normalization Exercise #1 Solution

SID → S_name

SID	S_name
1	Adams
2	Jones
3	Smith
4	Baker

CID → C_name

CID → Faculty

CID	C_name	Faculty	F_phone
IS318	Database	Smith	60192
IS301	Program	Johnson	45869
IS318	Database	Smith	60192
IS318	Database	Smith	60192
IS301	Program	Johnson	45869
IS318	Database	Smith	60192

3NF

(Transitive Dependency)

These two tables stay the same.

SID	CID	Grade
1	IS318	A

Normalization Exercise #1 Solution

1	IS301	B
2	IS318	A
3	IS318	B
4	IS301	A
4	IS318	B

SID	**S_name**
1	Adams
2	Jones
3	Smith
4	Baker

However, there is a transitive dependency: *Faculty → F_phone*

CID	**C_name**	**FID**
IS318	Database	1
IS301	Program	2
IS318	Database	1
IS318	Database	1
IS301	Program	2
IS318	Database	1

Normalization Exercise #1 Solution

FID	Faculty	F_phone
1	Howser	60192
2	Langley	45869

Final table list in 3NF:

Grade (SID*,CID*,Grade)

Student (SID, S_name)

Course (CID, C_name, FID*)

Faculty (FID, Faculty, F_phone)

Normalization Exercise #2 Solution

For the example below we have one big table. Put the table in normalized form.

OID = Order ID, O_Date= Order Date,
CID = Customer ID, C_Name = Customer Name, C_State = Customer's State,
PID = project id, P_Desc =Project Name, P_Price = Product Price, Qty = Quantity Purchased

Note: 7, 5, 4 means three Product IDs. Similarly, 1, 1, 5 means three Quantities.

Functional Dependencies are:

OID -> O_Date CID -> C_Name PID -> P_Desc PID -> P_Price
OID -> CID CID -> C_State PID and OID -> Qty

OID	**O_Date**	**CID**	**C_Name**	**C_State**	**PID**	**P_Desc**	**P_Price**	**Qty**
1006	10/24/09	2	Apex	NC	7, 5, 4	Table, Desk, Chair	800, 325, 200	1, 1, 5
1007	10/25/09	6	Acme	GA	11, 4	Dresser, Chair	500, 200	4, 6

Put the above table in 1NF Tables

In this step, remove any multivalued attributes and repeating groups by copying all of the information in that group to a separate row. The information under the PID, P_Desc, P_Price, and Qty has been separated into their own rows. The table below is now in first normal form.

OID	**O_Date**	**CID**	**C_Name**	**C_State**	**PID**	**P_Desc**	**P_Price**	**Qty**
1006	10/24/09	2	Apex	NC	7	Table	800	1
1006	10/24/09	2	Apex	NC	5	Desk	325	1
1006	10/24/09	2	Apex	NC	4	Chair	200	5
1007	10/25/09	6	Acme	GA	11	Dresser	500	4
1007	10/25/09	6	Acme	GA	4	Chair	200	6

Total # of Tables: 1

Normalization Exercise #2 Solution

Put the above table in 2NF

In the second normal form, we must remove any **partial functional dependencies**. These are attributes that depend directly on more than one primary key. If you look at the dependencies listed in the instructions, Qty is dependent on both OID and PID and that is your indication that it is a partial dependency. So the first step is to copy the OID and PID columns (together they make up the concatenated key) into new table and move the Qty column from the original table. Be sure to match the correct values from the original table to this new table:

OID	PID	Qty
1006	7	1
1006	5	1
1006	4	5
1007	11	4
1007	4	6

The next step is to split the remaining table into two separate tables for OID and PID and their dependencies. We chose those two attributes because of their use in the table above. The results are below:

PID	P_Desc	P_Price
7	Table	800
5	Desk	325
4	Chair	200
11	Dresser	500

OID	O_Date	CID	C_Name	C_State
1006	10/24/09	2	Apex	NC
1006	10/24/09	2	Apex	NC
1006	10/24/09	2	Apex	NC
1007	10/25/09	6	Acme	GA
1007	10/25/09	6	Acme	GA

All three tables make up the second normal form.

Total # of Tables: 3

Normalization Exercise #2 Solution

Put the above table in 3NF Tables

For third normal form, any **transitive dependencies** must be removed. Transitive dependency occurs when a non-primary key attribute such as C_Name is dependent upon more than one key. The first two tables from the second normal form do not have transitive dependencies so they are already in 3NF and just need to be copied over:

CID	PID	Qty
2	7	1
2	5	1
2	4	5
6	11	4
6	4	6

PID	P_Desc	P_Price
7	Table	800
5	Desk	325
4	Chair	200
11	Dresser	500

The third table in the 2NF has two transitive dependencies: C_Name and C_State. Both of those attributes are dependent on both OID and CID. Because they are directly dependent on CID, they are separated into their own table along with CID. In addition, CID is still dependent on OID so it becomes a foreign key in the OID table. Shown below are the old table and the 2 new tables:

Old Table:

OID	O_Date	CID	C_Name	C_State
1006	10/24/09	2	Apex	NC
1006	10/24/09	2	Apex	NC
1006	10/24/09	2	Apex	NC
1007	10/25/09	6	Acme	GA
1007	10/25/09	6	Acme	GA

New Tables:

OID	O_Date	CID
1006	10/24/09	2
1007	10/25/09	6

CID	C_Name	C_State
2	Apex	NC
6	Acme	GA

Please note that in all four of these tables, the rows are unique and none of the primary keys are repeated.

Total # of Tables: 4

Normalization Exercise #2 Solution

Final set of Tables with meaningful names and PKs and FKs

In this step, you must name the tables that were created and finalized in 3NF. Table names must have a meaningful name such that some third party looking at your design will know immediately what that table is used for. In the case where the table has a single primary key, you can call the table by the key such as Order, Product, or Customer. The other strategy is to call it Order_Info, Product_Info, or Customer_Info. The choice is entirely up to you and both are correct. The first table does not have a single primary key; instead it has a concatenated key (something found in an associative entity) and so the naming convention is somewhat different. One strategy is to call it by one of its owning entities and adding Detail to the end such as Order_Detail. Another strategy is to name the table by all of its owning entities such as Order-Product.

Please be sure to underline the primary keys and put an asterisk (*) at the end of a foreign key. Do not forget the other attributes and do not do anything special to them. Finally, for a concatenated key, be sure to underline all attributes of the key and put an asterisk after each one. The results are the following four table names:

Order-Product (OID*, PID*, Qty)
Product_Info (PID, P_Desc, P_Price)
Order_Info (OID, O_Date, CID*)
Customer_Info (CID, C_Name, C_State)

Normalization Exercise #3 Solution

1NF

(Repeating & Multivalued)

DID	Dname	EID	Ename	PID	Pname	Btime
10	Finance	1	Huey	27	Alpha	4.5
10	Finance	5	Dewey	25	Beta	3
10	Finance	11	Louie	22	Gamma	7
14	R&D	2	Jack	26	Pail	8
14	R&D	4	Jill	21	Hill	9

2NF

(Partial Dependence)

EID and PID → Btime

EID	PID	Btime
1	27	4.5
5	25	3
11	22	7
2	26	8
4	21	9

PID → Pname

PID	Pname
27	Alpha
25	Beta
22	Gamma
26	Pail
21	Hill

EID → Ename

EID → DID

EID	Ename	DID	Dname
1	Huey	10	Finance
5	Dewey	10	Finance
11	Louie	10	Finance
2	Jack	14	R&D
4	Jill	14	R&D

Normalization Exercise #3 Solution

3NF

(Transitive Dependency)

These two tables stay the same.

EID	PID	Btime
1	27	4.5
5	25	3
11	22	7
2	26	8
4	21	9

PID	Pname
27	Alpha
25	Beta
22	Gamma
26	Pail
21	Hill

However, there is a transitive dependency: *DID* → *Dname*

EID	Ename	DID
1	Huey	10
5	Dewey	10
11	Louie	10
2	Jack	14
4	Jill	14

DID	Dname
10	Finance
14	R&D

Final table list in 3NF:

Budgeted Time (EID*,PID*, Btime)

Project (PID, Pname)

Employee (EID, Ename, DID*)

Department (DID, Dname)