Class Exercise
Normalize User Views and Create a Final 3NF solution

Good News Grocers

User View 1 - Price Update List

Department	Product Code	Aisle Number	Price	Unit of Measure
Produce	4081	1	0.35	lb
Produce	4027	1	0.90	ea
Produce	4108	1	1.99	lb
Butcher	331100	5	1.50	lb
Butcher	331105	5	2.40	lb
Butcher	332110	5	5.00	lb
Freezer	411100	6	1.00	ea
Freezer	521101	6	1.00	ea
Freezer	866503	6	5.00	ea
Freezer	866504	6	5.00	ea

This report is used by the department managers to update the prices that are displayed in the grocery store for these products.

UNF:

dept [dept, aisle_no (prod_code, , price, um)]

1NF:

dept [dept_id, dept_name, aisle_no]
dept_product [dept, prod_code, price, um]

2NF:

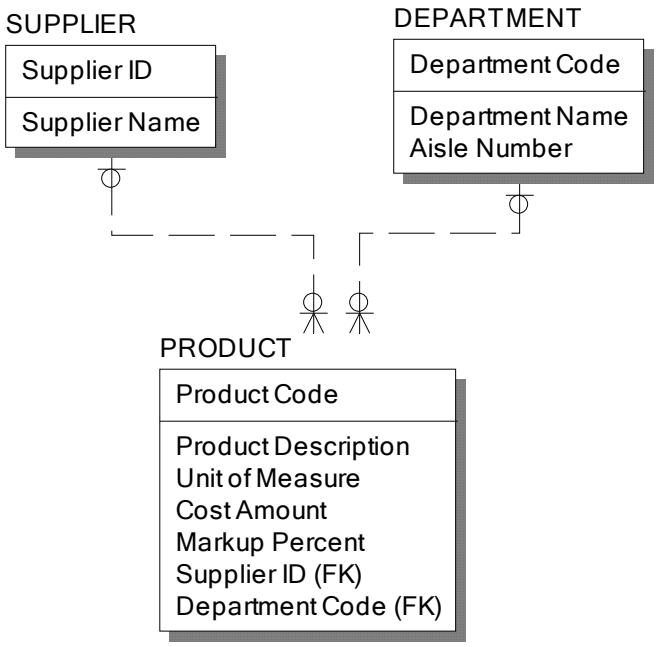
dept [dept_id, dept_name, aisle_no]
product [prod_code, price, um, dept_id (FK)]

3NF:

Note: **1.** examing the relationship between department and product, we discover that it is a 1:M, therefore we do not need the composite table, dept_product. So it is eliminated and the foreign key placed in the product table.
2. it may be debatable whether department actually determines aisle number. This may be true in a small grocery store but in a large grocery store, products from a department may be found in multiple aisles. This should be discussed with your database client to determine their exact needs.

Class Exercise
Normalize User Views and Create a Final 3NF solution

Class Normalization Exercise Solution:
Good News Grocers



Class Exercise
Normalize User Views and Create a Final 3NF solution

Good News Grocers
User View 2: Product Cost Report

Supplier	Product	Cost	Markup	Price	Dept Code
21 – Very Veggie	4108 – tomatoes, plum	1.89	5%	1.99	PR
32 – Fab Fruits	4081 – bananas	0.20	75%	0.35	PR
32 – Fab Fruits	4027 – grapefruit	0.45	100%	0.90	PR
32 – Fab Fruits	4851 – celery	1.00	100%	2.00	PR
08 – Meats R Us	331100 – chicken wings	0.50	300%	1.50	BU
08 – Meats R Us	331105 – lean ground beef	0.60	400%	2.40	BU
08 – Meats R Us	332110 – boneless chicken breasts	2.50	100%	5.00	BU
10 – Jerry's Juice	411100 – orange juice	0.25	400%	1.00	FR
10 – Jerry's Juice	521101 – apple juice	0.25	400%	1.00	FR
45 – Icey Creams	866503 – vanilla ice cream	2.50	100%	5.00	FR
45 – Icey Creams	866504 – chocolate ice cream	2.50	100%	5.00	FR

This report is used by the grocery store manager to determine the final selling price of his products.

UNF:

supplier [supplier_id, supplier_name, (prod_code, prod_desc, cost, markup, dept_cd)]

1NF:

supplier [supplier_id, supplier_name]
supplier_product [supplier_id, prod_code, prod_desc, cost, markup, dept_cd]

2NF:

supplier [supplier_id, supplier_name]
supplier_product [supplier_id, prod_code]
product [prod_code, prod_desc, cost, markup, dept_cd]

note: if we were getting a product from more than 1 supplier, then the cost attribute would go ino the supplier_product table.

3NF:

supplier [supplier_id, supplier_name]
product [prod_code, prod_desc, cost, markup, dept_cd, supplier_id (FK)]

note: examing the relationship between supplier and product, we discover that it is a 1:M, therefore we do not need the composite table, supplier_product. So it is eliminated and the foreign key placed in the product table.

Class Exercise
Normalize User Views and Create a Final 3NF solution

View 1 solution:

3NF:

dept [dept_id, dept_name, aisle_no]
product [prod_code, price, um, dept_id (FK)]

View 2 Solution:

3NF:

supplier [supplier_id, supplier_name]
product [prod_code, prod_desc, cost, markup, dept_cd, supplier_id (FK)]

Merged 3NF solution:

product [prod_code, prod_desc, um, dept_cd (FK), supplier_id (FK), cost, markup]
dept [dept_cd, dept_name, aisle_no]
supplier [supplier_id, supplier_name]

NOTE: although we took two different approaches to this question, we end up with the same 3NF solution!