This assignment is written by **Yuchen Yan (z5146418)**

Question 1:

1.

Because there are 3 dimensions, so there should be 2^3 = 8 cuboids.
Assume that Location to L, Time to L and Item to I. The cuboids are LTI, LI, TI, LT, L, T, I, ALL.
The table is shown below, where we use * to denote the special value ALL.

Cuboids	Location(L)	Time(T)	Item(I)	SUM(Quantity)
ALL	*	*	*	5100
LTI	Sydney	2005	PS2	1400
LTI	Sydney	2006	PS2	1500
LTI	Sydney	2006	Wii	500
LTI	Melbourne	2005	Xbox 360	1700
LI	Sydney	*	PS2	2900
LI	Sydney	*	Wii	500
LI	Melbourne	*	Xbox 360	1700
LT	Sydney	2005	*	1400
LT	Sydney	2006	*	2000
LT	Melbourne	2005	*	1700
TI	*	2005	PS2	1400
TI	*	2006	PS2	1500
TI	*	2006	Wii	500
TI	*	2005	Xbox 360	1700
L	Sydney	*	*	3400
L	Melbourne	*	*	1700
T	*	2005	*	3100
T	*	2006	*	2000
I	*	*	PS2	2900
I	*	*	Wii	500
I	*	*	Xbox 360	1700

2.

There should be 8 sql statements, which are following:

1.

SELECT SUM(Quantity)

FROM Sales
GROUP BY ALL
2.
SELECT Location, Time, Item, SUM(Quantity)
FROM Sales
GROUP BY Location, Time, Item
3.
SELECT Location, Time, SUM(Quantity)
FROM Sales
GROUP BY Location, Time
4.
SELECT Location, Item, SUM(Quantity)
FROM Sales
GROUP BY Location, Item
5.
SELECT Time, Item, SUM(Quantity)
FROM Sales
GROUP BY Time, Item
6.
SELECT Location, SUM(Quantity)
FROM Sales
GROUP BY Location
7.
SELECT Time, SUM(Quantity)
FROM Sales
GROUP BY Time
8.
SELECT Item, SUM(Quantity)
FROM Sales
GROUP BY Item

3.

Cuboids	Locaiton(L)	Time(T)	Item(I)	Count(*)	SUM(Quantity)
ALL	*	*	*	4	5100
LI	Sydney	*	PS2	2	2900
LT	Sydney	2006	*	2	2000
L	Sydney	*	*	3	3400
T	*	2005	*	2	3100
T	*	2006	*	2	2000
I	*	*	PS2	2	2900

4.

Using the mapping function given by the question, I firstly convert the original cube into this following cube.

Cuboids	Location(L)	Time(T)	Item(I)	SUM(Quantity)
ALL	0	0	0	5100
LTI	1	1	1	1400
LTI	1	2	1	1500
LTI	1	2	3	500
LTI	2	1	2	1700
LI	1	0	1	2900
LI	1	0	3	500
LI	2	0	2	1700
LT	1	1	0	1400
LT	1	2	0	2000
LT	2	1	0	1700
TI	0	1	1	1400
TI	0	2	1	1500
TI	0	2	3	500
TI	0	1	2	1700
L	1	0	0	3400

Cuboids	Location(L)	Time(T)	Item(I)	SUM(Quantity)
L	2	0	0	1700
T	0	1	0	3100
T	0	2	0	2000
I	0	0	1	2900
I	0	0	3	500
I	0	0	2	1700

Then I use the mapping function: F(Location, Time, Item) = Location*12 + time*4 + item

In order to map a multi-dimensional point to a one-dimensioinal point.
The final table is like this:

Offset	SUM(Quantity)
0	5100
17	1400
21	1500
23	500
30	1700
13	2900
15	500
26	1700
16	1400
20	2000
28	1700
5	1400
9	1500
11	500
6	1700
12	3400
24	1700
4	3100
8	2000
1	2900

Offset	SUM(Quantity)
3	500
2	1700

<u>Question 2:</u>
<u>1.</u>
Proof:
(1) The function of a linear classifier is:
y = 1, if $w^\top x>0$ and y = 0. otherwise.
$w^\top x = w0+\Sigma wi *xi$

(2) Now we consider Na¨ıve Bayes classifier:
h(x) = argmax P(h|X) = argmaxP(X|h)P(h)
h = argmax P(h) Π P(x|h)

(3) Here the h are 0 and 1, so we can convert the equation into a binary classifier like this:
g(x) = P(1) Π P(x|1) / P(0) Π P(x|0)
And if y = 1 if g(x) > 1, y = 0 otherwise.

(4)If we take the logarithm of the threshold(In1 = 0) and g, then we can get the function:
h(x) = In(P(1)/P(0)) + Σ In(P(x|1) / P(x|0)) * xi (where xi = 1 or xi = 0)
And if y = 1 if h(x) > 0, y = 0 otherwise.
Now it becomes a linear classifier where W = In(P(x|1) / P(x|0))

<u>2.</u>
When to calculate the Wnb, you only need to calculate the In(P(xi|1)/P(xi|0)) for each wi. It can be calculated **one by one xi** and it is relatively easy.

But when to find the Wir. For example, if you use gradient ascent method to find the Wi. Then for each Wi you need to calculate the sum of (y - p(xi)*xi for **every xi**. So compare to wi of Na¨ıve Bayes, it is a little bit hard to calculate.

<u>Question 3:</u>
<u>1.</u>
Proof:
The likelihood of x = p if yi = 1

The likelihood of x = 1 - p, otherwise.

As a result, the likelihood is p^y * (1-p)^(1-y)

So the likelihood for whole training set is L(w) = ∏p(xi)^yi * ((1-p(xi))^(1-yi))

(2)As a result the log-likelihood is l(w) = ∑yi * log p(xi) + (1 - yi) log(1-p(xi))

(3)In order to find the parameter w we should to maximize l, so the loss function should be -1 *(log-likelihood function)
Loss function = ∑- yi * log p(xi) - (1 - yi) log(1-p(xi)) where p is the **sigmoid function.**

(4) convert the loss function above to the loss function like the question give here:

$$l(w) = \sum_{i=1}^{n} - y_i \ln\left(\frac{1}{1+e^{-w^Tx}}\right) - (1-y_i)\ln\left(1-\frac{1}{1+e^{-w^Tx}}\right)$$

$$l(w) = \sum_{i=1}^{n} -y_i \ln\left(\frac{1}{1+e^{-w^Tx}}\right) - (1-y_i)\ln\left(\frac{e^{-w^Tx}}{1+e^{-w^Tx}}\right)$$

$$= \sum_{i=1}^{n} - y_i \ln\left(\frac{e^{w^Tx}}{1+e^{w^Tx}}\right) - (1-y_i)\ln\left(\frac{\frac{1}{e^{w^Tx}}}{\frac{1+e^{w^Tx}}{e^{w^Tx}}}\right)$$

$$= \sum_{i=1}^{n} - y_i \ln\left(e^{w^Tx}\right) - \ln\left(\frac{1}{1+e^{w^Tx}}\right)$$

$$= \sum_{i=1}^{n} - y_i \ln\left(e^{w^Tx}\right) - \ln\left(\frac{1}{1+e^{w^Tx}}\right)$$

$$= \sum_{i=1}^{n} - y_i w^Tx + \ln\left(1+e^{w^Tx}\right)$$

As a result the equation given is the loss function of the logistic regression. Proof done.

2.
The likelihood of x $= f(w^\top x)$ if yi = 1

The likelihood of x = $1 - f(w^\top x)$, otherwise.

As a result, the likelihood is p^y * (1-p)^(1-y)

So the likelihood for whole training set is $L(w) = \prod f(w^\top x)$^yi * $((1-f(w^\top x))$^(1-yi))

(2)As a result the log-likelihood is $l(w) = \sum$yi * log $f(w^\top x)$ + (1 - yi) log$(1-f(w^\top x))$

(3)In order to find the parameter w we should to maximize l, so the loss function should be -1 *(log-likelihood function)

As a result the loss function should be:
$l(w) = \sum$-yi * log $f(w^\top x)$ - (1 - yi) log$(1-f(w^\top x))$