NORTH SOUTH UNIVERSITY

Department of Electrical and Computer Engineering

PROJECT

CSE 411

Section 02

Design an E-commerce Data WareHouse

Submitted to

Abu Sayed Md. Latiful Hoque

Professor, CSE, BUET

Visiting Professor, NSU

Submitted by

Asif Anan 1621358042

Task: 1

Design the architecture of the warehouse and explain the sources, preprocessing, noise reduction, transformation and uploading.

The architecture of ecommerce data warehouse model is illustrated in the Figure 1. Data will be collected from different sources such as superstores, chain shops, suppliers.

Then data preprocessing, one of the major task for developing a Data warehouse from heterogeneous sources, will be performed. It includes data cleaning, missing values imputation, normalization, transformation etc.

As data are coming from different sources, different preprocessing steps will be executed on data such as cleaning, noise. reduction and normalization techniques. Using Extraction - Load - Transform (ETL) process, the preprocessed data will be integrated into a temporary repository. After that, data will be baded into DW. Online Analytical Processing (OLAP) queries and mining operations can be easily performed over this Data Warehouse.

TASK-1

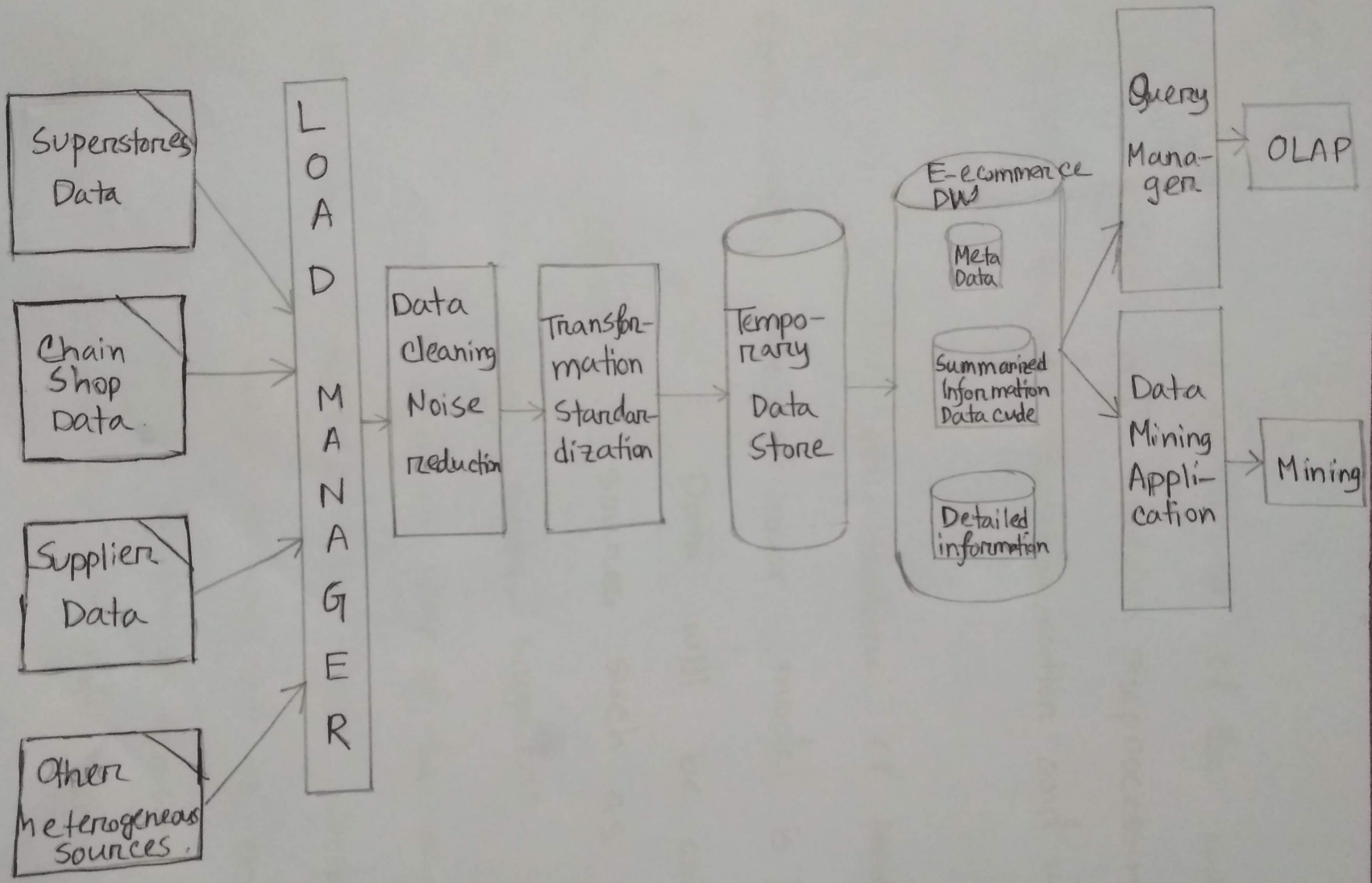


Figure: Design the architecture of the WareHouse.

Task 2:

Design the star schema for the warehouse using the scenario and the data set and explain how the data of the superstore database will be collected to the DW.

Here, a fact table sales would have dimension attributes payment key, customer key, time key, item key, store key, and measure attributes quantity, unit price, total price. The attribute payment key is a foreign key into a dimension table Transition. which has other attributes such as trans_type and bank_name.

The customer key attribute of the sales table would be a foreign key into a dimension table customer, which contain information such as the name, contact_no, nid, address, street, upazilla, district, division.

The time key attribute would be a foreign key into a Time table containing attributes such as date, hour, day, week, month, quarter and year. We can also view the store key attribute as a foreign key into a store table giving the location, city, upazilla and district.

The item key attribute would be a foreign key into a item table containing attributes such as item name, unit price, man_country, supplier and stock_quantity.

Task-2

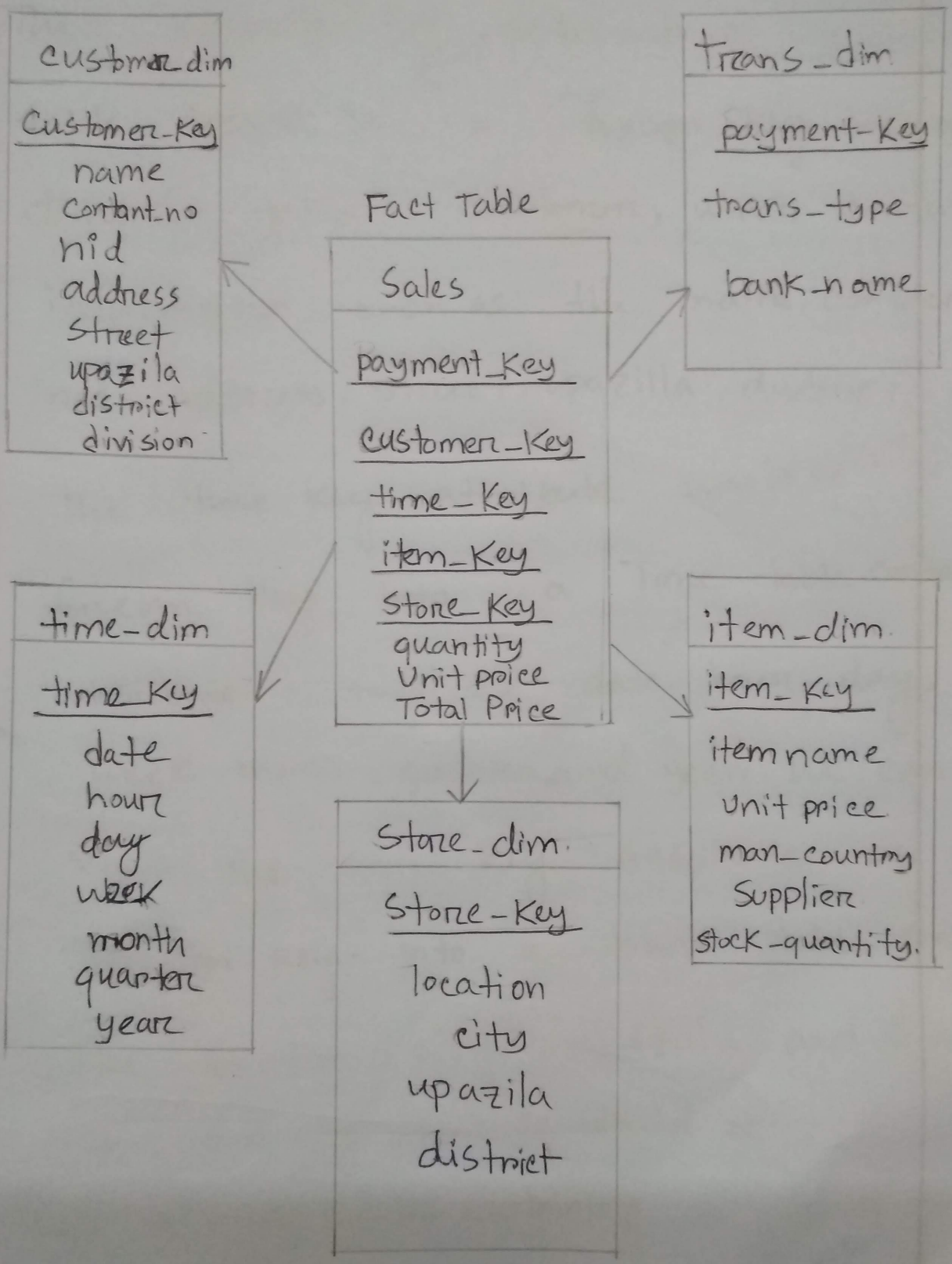


Figure: star schema for the warehouse

Task-3:

Write three different cross tabulation for three different dimension using total price/quantity and show the cross tabulation data using bar/pie charts. Write SQL to find the cross tabs.

Solve:

1) Sales_item (item_name, unit price, country, quantity).

```
Select item_name, unit price, country, quantity
From item_dim a, sales b.
    where a.item_key = b.item_key.
```

Country [all]

item name	unit price.	total
total		

SQL for country:

```
Select sum(quantity) from sales.
```

SQL for total of item_name:

```
Select item_name, sum(quantity)
From sales.
Group by item_name.
```

TASK-03

SQL for total Unit price:

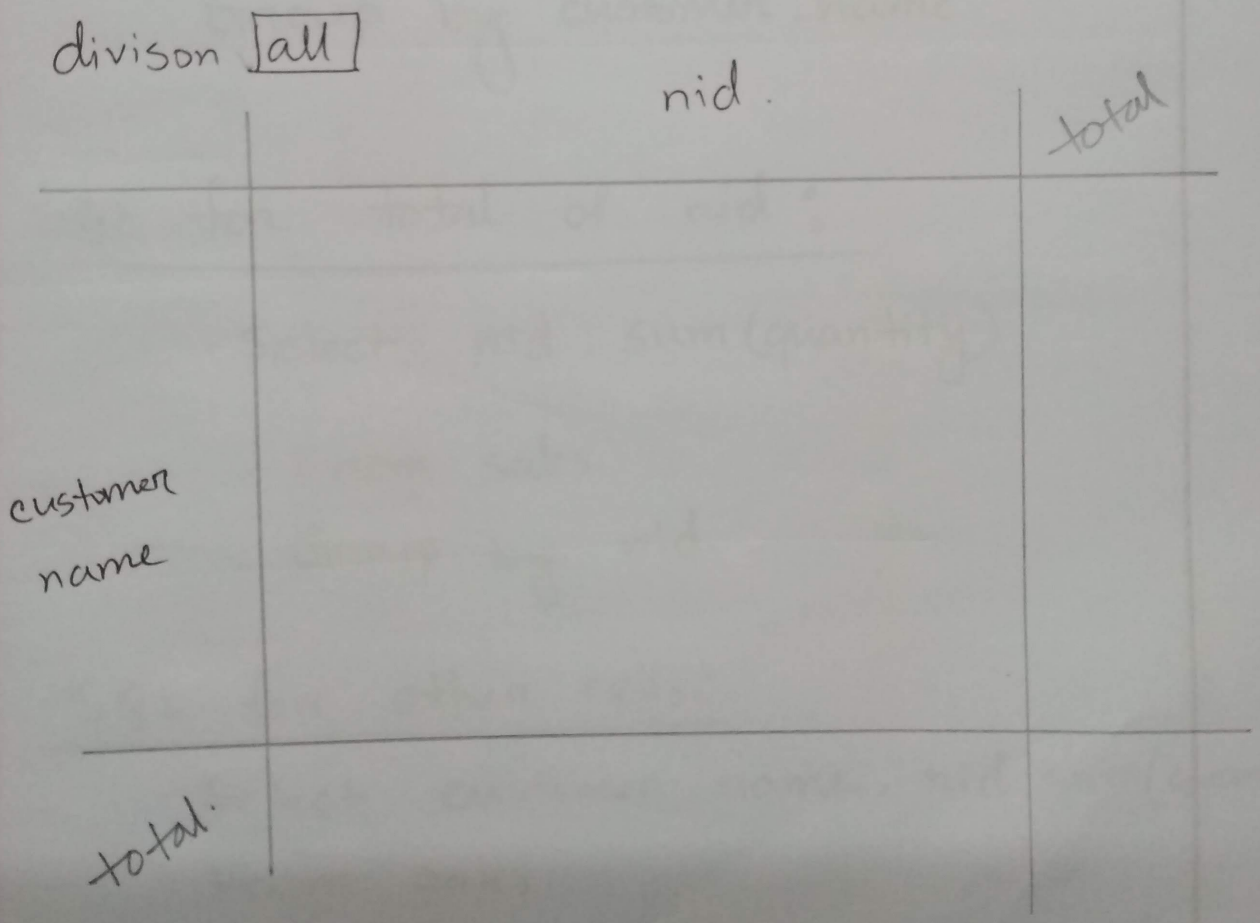
```
Select Unit price, sum (quantity)
From sales.
Group by unit_price.
```

SQL for other cells:

```
Select item_name, unitprice, sum (quantity).
From sales.
Group by item_nam, unit price.
```

2) Sales Item (customer name, nid , divison)

```
Select  customer_name, nid , divison , quantity
From  customer_dim a, sales b.
  Where  a.customer_key = b.customer_key.
```



SQL for divison:

```
Select sum(quantity) from sales.
```

SQL for total of customer_name:

```
Select customer_name, sum(quantity).
From sales.
Group by customer_name.
```

SQL for total of nid:

```
Select nid, sum(quantity)
From sales.
Group by nid.
```

SQL for other cells:

```
Select customer_name, nid. sum(quantity)
From sales.
Group by customer_name, nid.
```

3) Sales item (location, city, district, quantity)

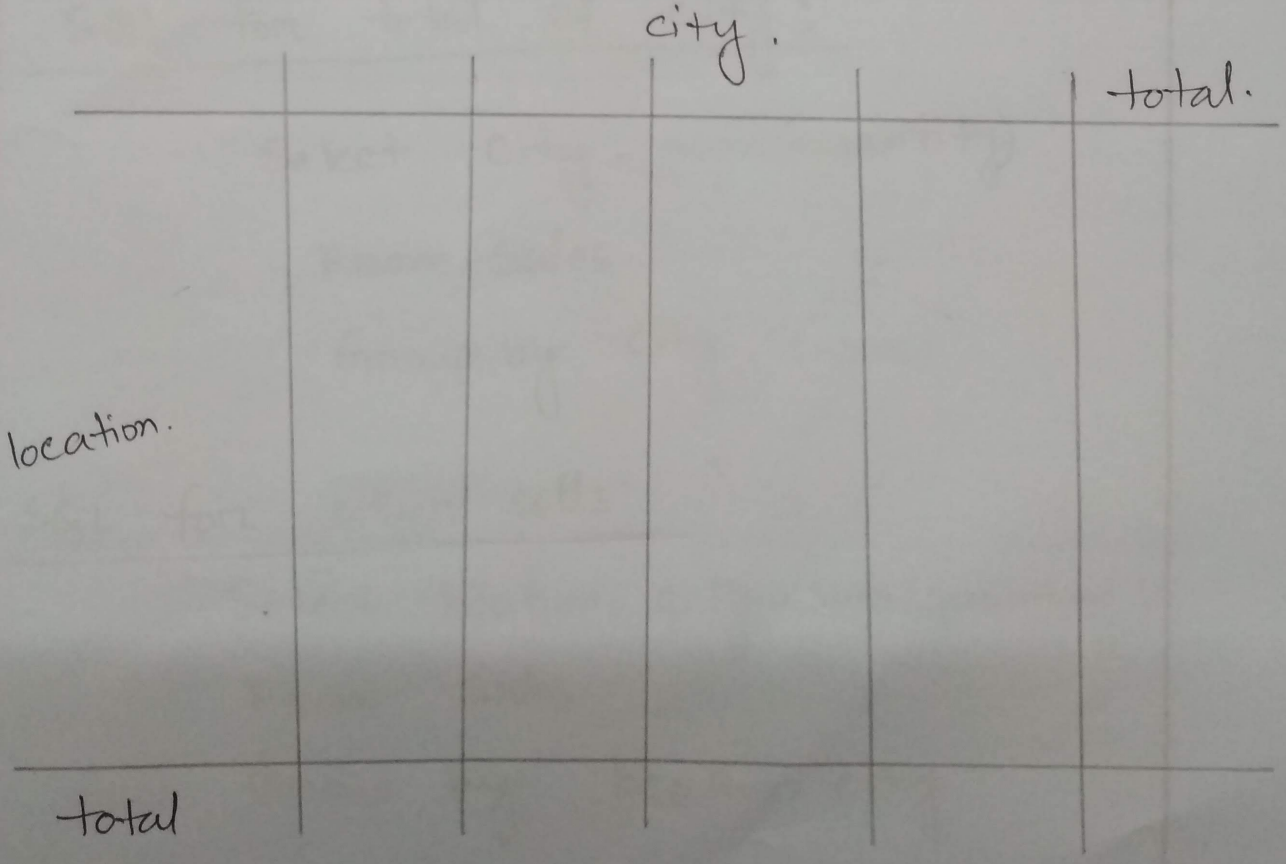
Select location, city, district, quantity

From store_dim a, sales b.

~~Where a. item_id = b~~

where a. store_key = b. store_key.

district [all]



SQL for district: ~~fall~~

```
Select sum(quantity) from sales.
```

SQL for total of location:

```
Select location, sum(quantity)
From sales.
Group by location.
```

SQL for total of city:

```
Select city, sum(quantity)
From sales.
Group by city.
```

SQL for other cells:

```
Select location, city, sum(quantity).
From sales.
Group by location, city.
```

Task-4:

Show the structure of a 4 dimensional data cube and explain.

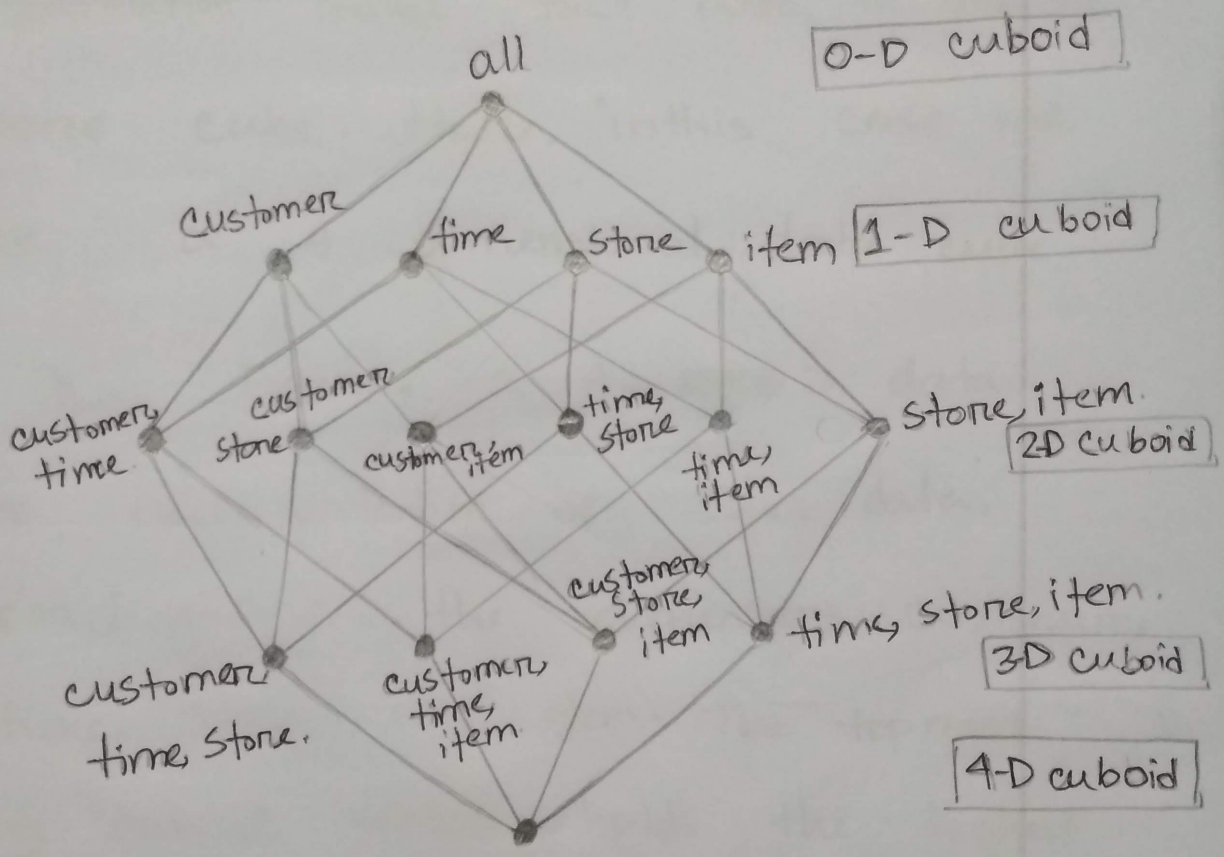


Figure: 4 dimensional data cube.

A database dimension is a collection of related objects, called attributes, which can be used to provide information about fact data in one or more cube. Here, inthis case we use a 4 dimensional data cube.

In this Figure, a 4-D data cube representation of sales data, according to the dimensions customer, time, store and item. The topmost 0-D cuboid which holds the highest level of summarization, is also known as the apex cuboid. In this figure, this is the total sales, summarized over all four dimensions.