CSE 453

High Performance Database System

DW-Assignment-1

Assignment on Big Data Analytics (Data Warehouse)

Name : Abdur Rahman Fahad
Std ID : 1605069
Section : B1

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

Task 1

Design the architecture of the warehouse and explain the sources, pre-processing, noise reduction, transformation and uploading.

ETL (or Extract, Transform, Load) is a process of data integration that encompasses three steps — extraction, transformation, and loading. In a nutshell, ETL systems take large volumes of raw data from multiple sources, converts it for analysis, and loads that data into your warehouse

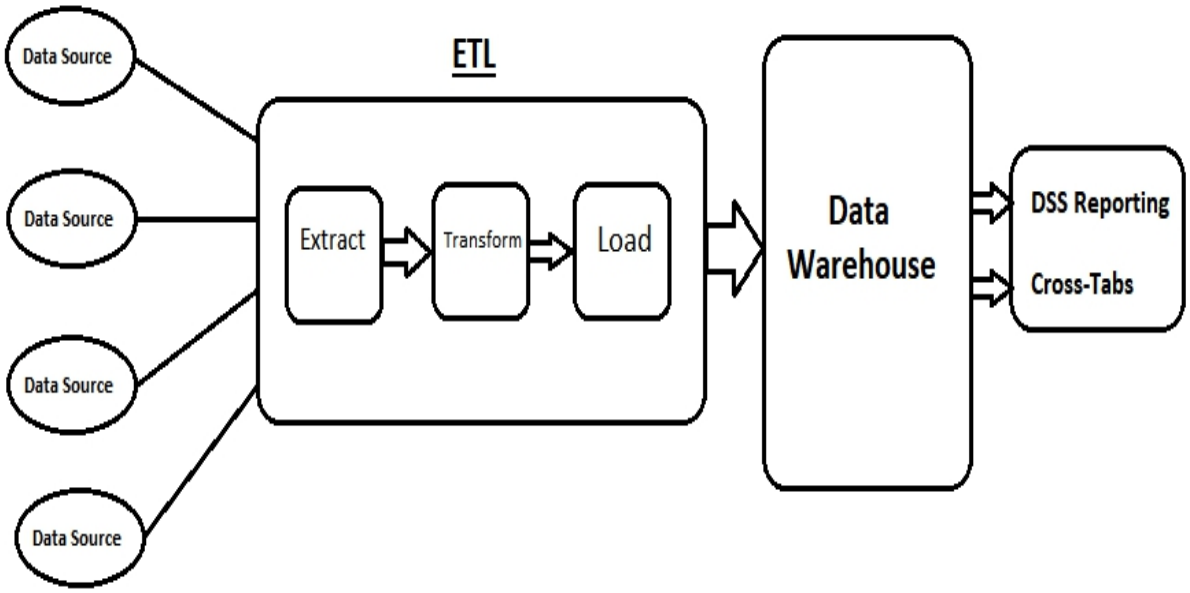


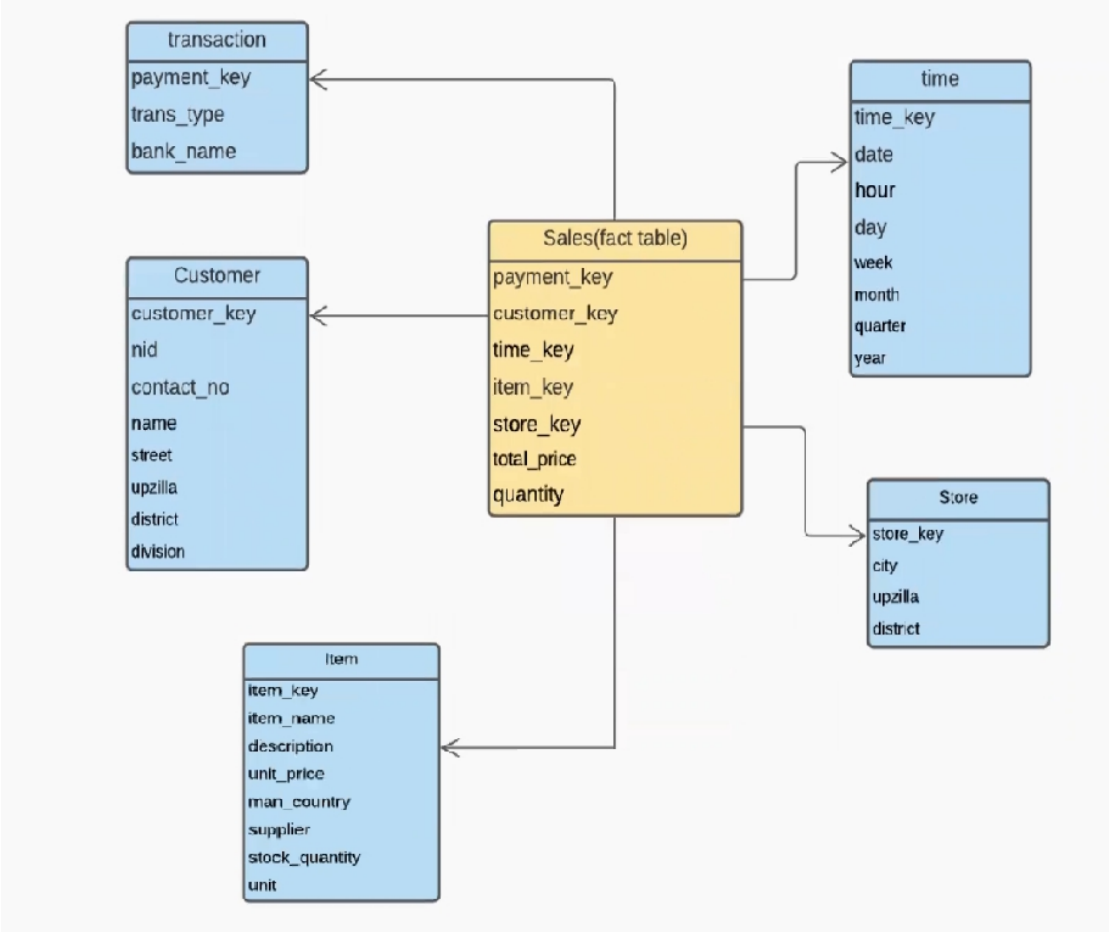
Figure : Architecture of the Warehouse

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 (source driven or destination driven).

The data of the superstore database will be collected to the DW using **source driven manner**.

<u>The Star Schema</u>



Task 3

Implement the star schema in PostgreSQL and upload the given data into the database.

The star schema was implemented in PostgreSQL. Here are some of the table creation processes.

```
CREATE TABLE IF NOT EXISTS public.customer_dim
(
    nid numeric(13,0),
    name character varying(100),
    contact_no numeric(14,0),
    address character varying(100),
    street character varying(100),
    upazila character varying(100),
    district character varying(100),
    division character varying(100),
    customer_key character varying(7) PRIMARY KEY
)

TABLESPACE pg_default;

ALTER TABLE public.customer_dim
    OWNER to postgres;


COPY customer_dim(customer_key,name,contact_no,nid,address,street,upazila,district,division)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\customer_dim.csv'
DELIMITER ','
CSV HEADER;
```

```
CREATE TABLE IF NOT EXISTS public.store_dim
(
    store_key character varying(5) PRIMARY KEY,
    location character varying(100),
    city character varying(20),
    upazila character varying(20),
    district character varying(20)
)

TABLESPACE pg_default;

ALTER TABLE public.store_dim
    OWNER to postgres;


COPY customer_dim(store_key,location,city,upazila,district)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\store_dim.csv'
DELIMITER ','
CSV HEADER;
```

```
CREATE TABLE IF NOT EXISTS public.item_dim
(
    item_key character varying(6) PRIMARY KEY,
    item_name character varying(50),
    description character varying(50),
    unit_price real,
    man_country character varying(20),
    supplier character varying(30),
    stock_quantity real,
    unit character varying(10)
)

TABLESPACE pg_default;

ALTER TABLE public.item_dim
    OWNER to postgres;


COPY item_dim(item_key, item_name, description, unit_price, man_country, supplier, stock_quantity, unit)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\item_dim.csv'
DELIMITER ','
CSV HEADER;
```

```sql
CREATE TABLE IF NOT EXISTS public.time_dim
(
    time_key character varying(7) PRIMARY KEY,
    date character varying(20),
    hour smallint,
    day smallint,
    week character varying(10),
    month smallint,
    quarter character varying(2),
    year smallint
)

TABLESPACE pg_default;

ALTER TABLE public.time_dim
    OWNER to postgres;


COPY time_dim(time_key, date, hour, day, week, month,quarter, year)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\time_dim.csv'
DELIMITER ','
CSV HEADER;
```

```sql
CREATE TABLE IF NOT EXISTS public.trans_dim
(
    payment_key character varying(6) PRIMARY KEY,
    trans_type character varying(10),
    bank_name character varying(100)

)

TABLESPACE pg_default;

ALTER TABLE public.trans_dim
    OWNER to postgres;


COPY trans_dim(payment_key, trans_type, bank_name)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\trans_dim.csv'
DELIMITER ','
CSV HEADER;
```

```sql
CREATE TABLE IF NOT EXISTS public.fact_table
(
    customer_key character varying(7),
    time_key character varying(7),
    item_key character varying(6),
    store_key character varying(5),
    quantity smallint,
    unit character varying(10),
    unit_price real,
    total_price real,
    payment_key character varying(6),
    CONSTRAINT customer_key FOREIGN KEY (customer_key)
        REFERENCES public.customer_dim(customer_key) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT item_key FOREIGN KEY (item_key)
        REFERENCES public.item_dim(item_key) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT payment_key FOREIGN KEY (payment_key)
        REFERENCES public.trans_dim(payment_key) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT store_key FOREIGN KEY (store_key)
        REFERENCES public.store_dim(store_key) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT time_key FOREIGN KEY (time_key)
        REFERENCES public.time_dim(time_key) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE public.fact_table
    OWNER to postgres;


COPY fact_table(payment_key, customer_key, time_key, item_key, store_key, quantity, unit, unit_price, total_price)
FROM 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\fact_table.csv'
DELIMITER ','
CSV HEADER;
```

Task 4

Three different cross tabulation for three different dimensions using total price/quantity. Write SQL to find the cross-tabs.

Cross Tabulation of item_dim by item_name and man_country.

	India	China	- - - - - - - - - -	Total
Topo Chico	0	0		30
Honey Packets	0	200		200
- - - -				
Total	1467	4637	- - - - - - - - -	32430

Queries used to obtain cross tabulation are given below

```
select item_name, sum(stock_quantity)
from item_dim
group by item_name;

select man_country, sum(stock_quantity)
from item_dim
group by man_country;

select sum(stock_quantity) from item_dim;

select item_name, man_country, sum(stock_quantity)
from item_dim
group by item_name, man_country;
```

Cross Tabulation of fact_table by store_key and item_key.

	I00141	I00121	- - - - - - - - - -	Total
S0021	811	819		206057
S0011	836	872		205762
- - - -				
Total	23471	23103	- - - - - - - - - -	5993859

Queries used to obtain cross tabulation are given below

```
select item_key, sum(quantity)
from fact_table
group by item_key;

select sum(quantity) from fact_table;

select store_key, sum(quantity)
from fact_table
group by store_key;

select store_key, item_key, sum(quantity)
from fact_table
group by store_key, item_key;
```

Cross Tabulation of fact_table by store_key and item_key.

	C000001	C000002	- - - - - - - - - -	Total
S0001	140	454		3.605168e+06
S0002	293	494		3.675206e+06
- - - -				
Total	1090125	11106	- - - - - - - - - -	1.05217816e+08

Queries used to obtain cross tabulation are given below

```
select store_key, customer_key, sum(total_price)
from fact_table
group by store_key, customer_key;

select store_key, sum(total_price)
from fact_table
group by store_key;

select customer_key, sum(total_price)
from fact_table
group by customer_key;

select sum(total_price) from fact_table;
```

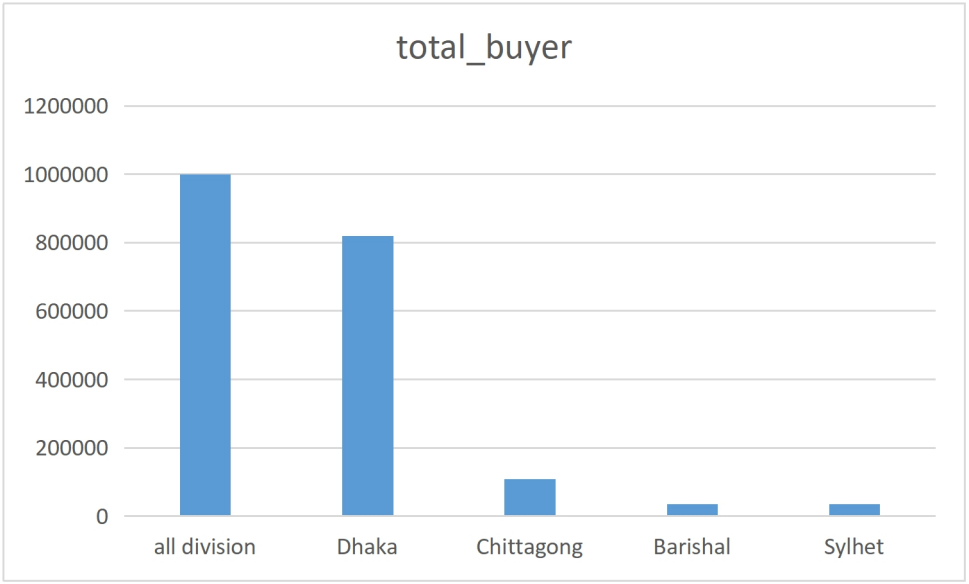
Task 5

Find at least 5 important DSS reports (one for each dimension) as bar chart. Use cube operation in SQL to find the reports data.

DSS Report on Division-wise Buyers Count

```
copy (select coalesce(division, 'all division') division, count(*) as total_buyer
from fact_table, customer_dim
where fact_table.customer_key = customer_dim.customer_key
group by cube(division)
order by total_buyer desc)
to 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\dss\division_buyer.csv'
DELIMITER ','
CSV HEADER;
```

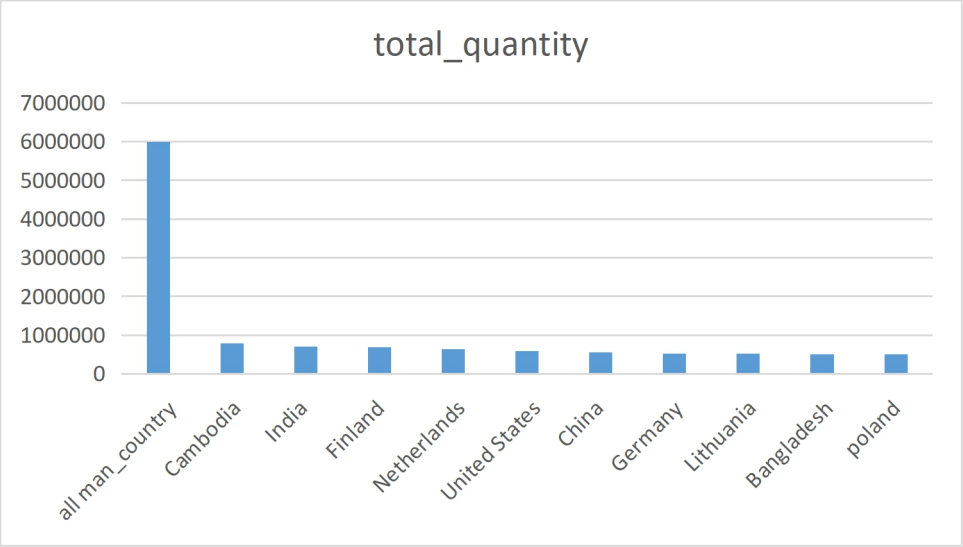
Bar Chart



DSS Report on Country-wise Item Count

```
copy (select coalesce(man_country, 'all man_country') man_country, sum(quantity) as total_quantity
from fact_table, item_dim
where fact_table.time_key = item_dim.item_key
group by cube(man_country)
order by total_quantity desc)
to 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\dss\man_country.csv'
DELIMITER ','
CSV HEADER;
```

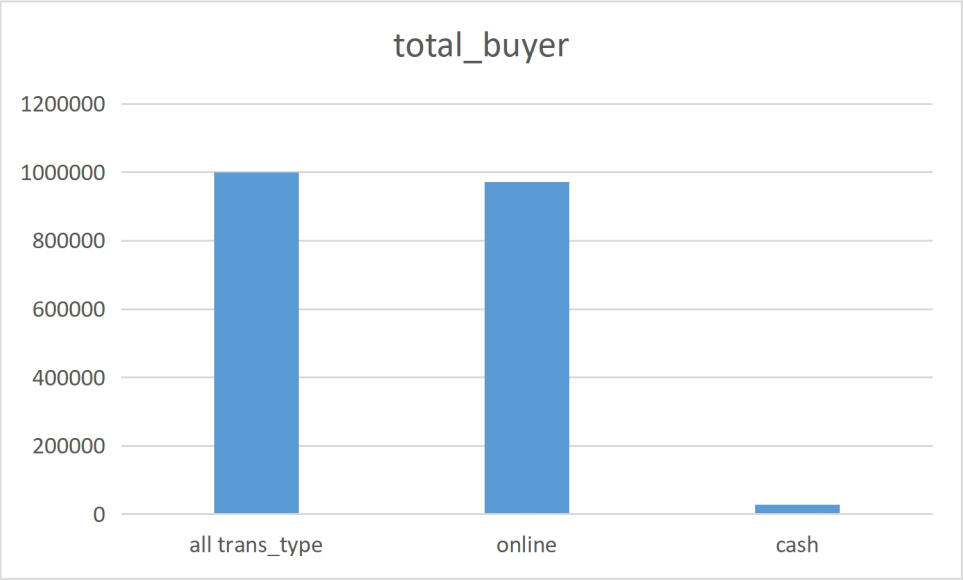
Bar Chart



DSS Report on Transaction-wise Buyers Count

```
copy (select coalesce(trans_type, 'all trans_type') trans_type, count(*) as total_buyer
from fact_table, trans_dim
where fact_table.payment_key = trans_dim.payment_key
group by cube(trans_type)
order by total_buyer desc)
to 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\dss\trans_buyer.csv'
DELIMITER ','
CSV HEADER;
```

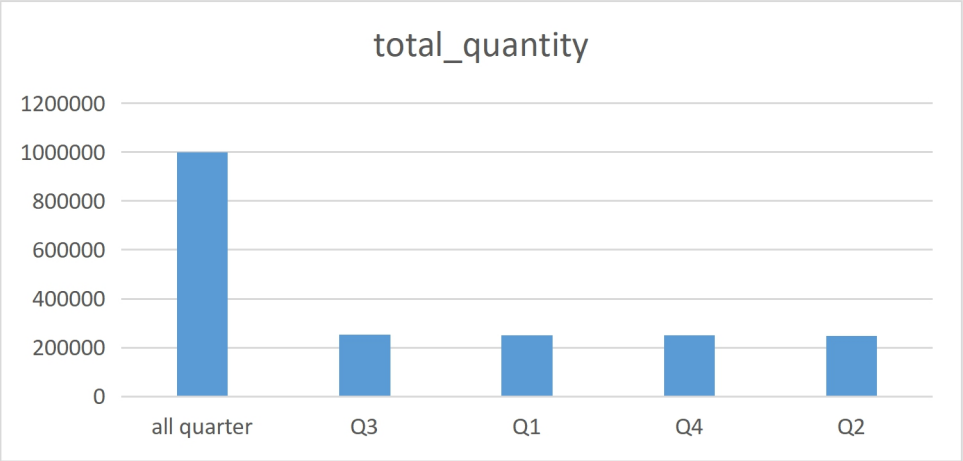
Bar Chart



DSS Report on Quarter-wise Sales Count

```
copy (select coalesce(quarter, 'all quarter') quarter, count(*) as total_quantity
from fact_table, time_dim
where fact_table.time_key = time_dim.time_key
group by cube(quarter)
order by total_quantity desc)
to 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\dss\quarter.csv'
DELIMITER ','
CSV HEADER;
```

Bar Chart



DSS Report on District-wise Sales Count

```
copy (select coalesce(district, 'all district') district, sum(total_price) as total_price
from fact_table, store_dim
where fact_table.store_key = store_dim.store_key
group by cube(district)
order by total_price desc)
to 'D:\4-1\CSE 453 HPDS\assignment\dw-assgnment-datasets\dss\earning_district.csv'
DELIMITER ','
CSV HEADER;
```

Bar Chart