VicRoads System Design Report

Completed by:
Yuanyi Li (27552675)

Completed Date:
2019/10/19

Table Of Content

Introduction

Vicroads department has collected a series of daily traffic data in the particular sections, they would like to implement some data analysis which can provide an overview of the traffic condition. Throughout some key figures summarised and charts, the Vicroads department intends to have a deep understanding of the traffic pattern as well as the problems related. By doing so, the department will gain some ideas to take some actions to optimize the usage of the roads, reduce problems, hence increase the efficiency of the traffic flow. The report is a system design report that aims to provide the Vicroads department with a clear design process and explanation regarding the Data Warehouse and the proposed interface. The report includes an overview of our solution, the design process and the explanation regarding the operational database, the multi-dimensional model, the data warehouse design and interface. Also, a data dictionary will be provided to demonstrate the details of the entities and attributes for the star schema. An appendix that contains SQL code which is used to convert the raw data to the operational database and then transfer to the data warehouse is provided at the end.

Solution Plan

At first, the raw data set is analysed by knowing each attributing meaning and finding the relationship. Then we draw an Entity-relationship diagram(ERD) to implement an operational database. The operational database is used to store the data from the raw data set and better organize and manage information in the future. As the dataset is large and according to the business requirement, a data warehouse is recommended to be built. It could be designed by using a multi-dimensional model in the form of Thomsen diagram and implement by star-schema approach. In detail, the data in the operational database is extracted, then it would be done some transform such as be cleaned and changed the format if needed, finally load the processed data into the data warehouse. After that, a user interface will be designed and implemented based on the business need. We also create some OLAP queries to

manipulate business requirement so that VicRoads company can see some information clear.

Therefore, the process can be concluded as follows:

1. Analysis of the raw dataset and find the relationship
2. Design an ERD of the operational database
3. Implement the operational database to store the raw data
4. Design Thomsen diagrams
5. Design star-schema
6. Implement data warehouse
7. Create some OLAP queries
8. Design and implement a user interface

The data cleaning is made of need during the whole process.

Key design decisions

In terms of the data visualisation, on the one hand, we provide some key figures which come from the aggregation of the data model. These historical data will be able to let the viewer know the exact data under a certain time and location. On the other hand, we present the trend of the traffic flow by some line charts which make the viewer easy to observe the change of the flow over time.

In order to make an appropriate multidimensional schema, we transform the metadata into an entity relationship diagram which will be helpful to analyse the association between different entities. All of the required attributes and cardinality have been addressed in the initial diagram. We decide to make the road link information and the flow information as one to many relationships. And add one more attribute which is record_id to identify each record. Each hour flow will be a column in the table. Next go into the stage of design of the multi-dimensional schema, based on the entity relationship diagram created, we extract more tables and create some new keys for these entities. As a result, we make one fact table and

four dimension tables which are associated with the fact table. The schema contains the time, link, period, flow direction and flow table. Each table has its own id as a unique identifier.

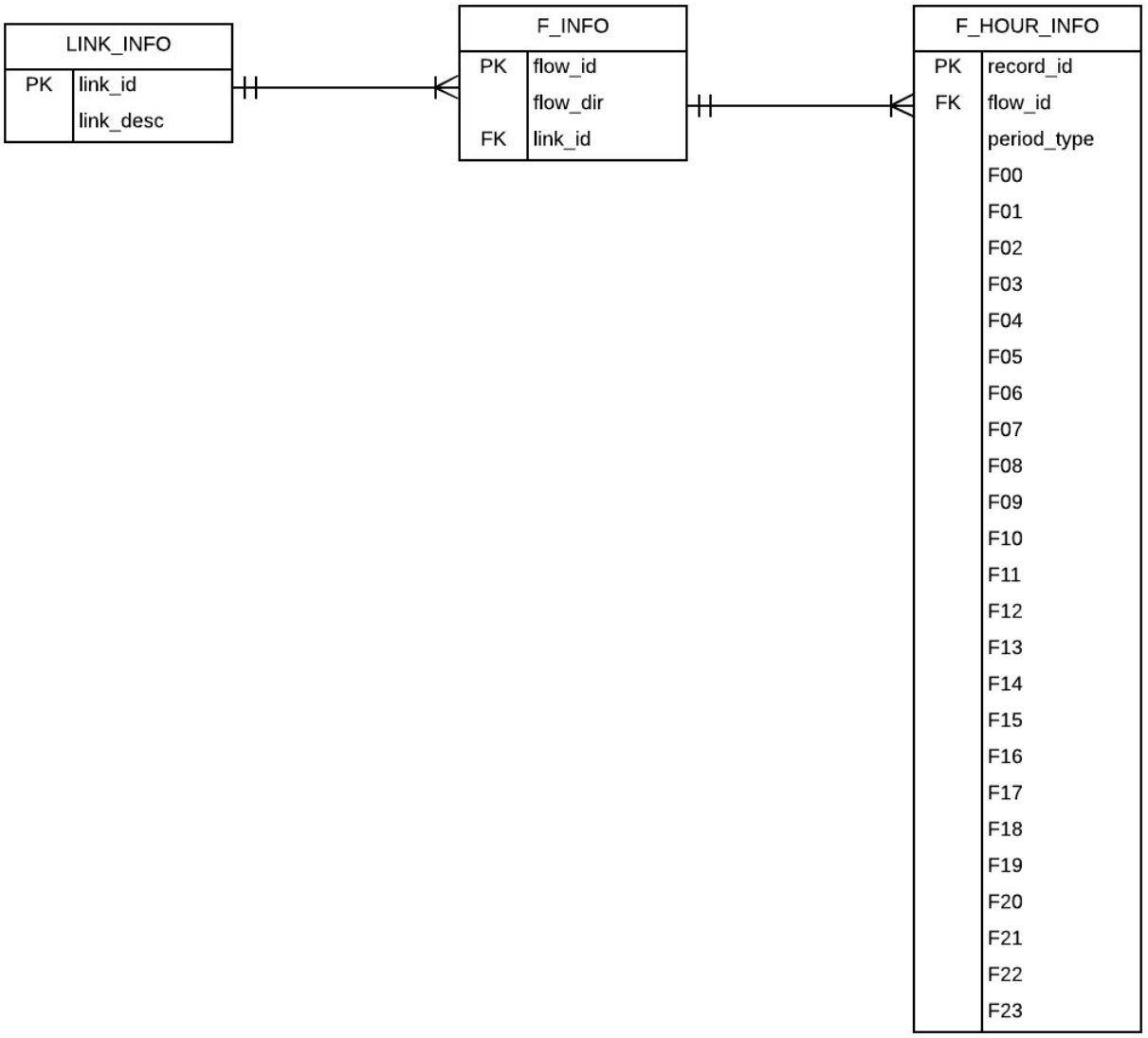
Process And Explanation

1.0 Operational Database Creation

1.1 ERD Design

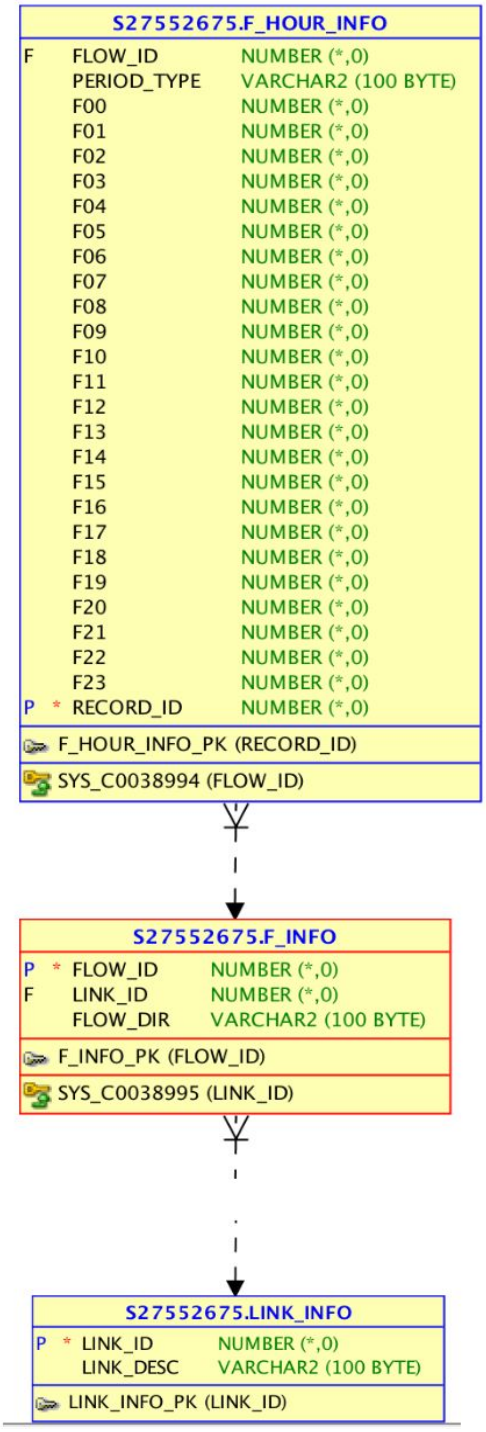
We found that a flow_id can identify a specific link_id and a flow_dir, but the same link can be recorded in many flow_id. Therefore, links and flows in the dataset have a one-to-many relationship. As link_id identifies a specific link that only has one link description, they can be in the same table. The design is shown below:

As there are lots of repeated record that with the same flow_id with the same link_id with the same flow direction and same period type but different counts of per hour, which means it is impossible to implement a reasonable primary key or reasonable primary key sets for the original dataset. So we add one more attribute called record_id and make it as a primary key to identify each record. Then store the rest of the attribute combined with flow_id in one table for convenience. Flow_id here is used to link to F_INFO table as a one-to-many relationship. The final design is as below.



1.2 Implement Operational Databases

In SQL Developer, the raw data is imported. Then we create three empty tables with the entity name and attribute. Finally, insert selectively the data from the raw data table to the particular table. The detail code is shown in the appendix. The model came out is as follows.



1.3 Clean Data

To ensure the data quality, we apply data sourcing and cleansing functions on ETL process. Firstly, as there are some null values in the operational database which may affect the performance of the data analysis, we decide to delete these objects. Some link_ids in the tables have no corresponding descriptions, we delete these kind of data from the table f_link_info, hence delete the link_ids which match the no description links deleted from the f_info table and corresponding flow_ids from f_hour_info table. Therefore, all of the data will be in an integrated format with full information presented. Secondly, in order to fix the issue of referential integrity, we turn the period_type in f_hour_info, flow_dir in f_info table, link_desc in f_link_info table into upper case. By doing so, the occurrence of no data found will be eliminated during the search of the end user.

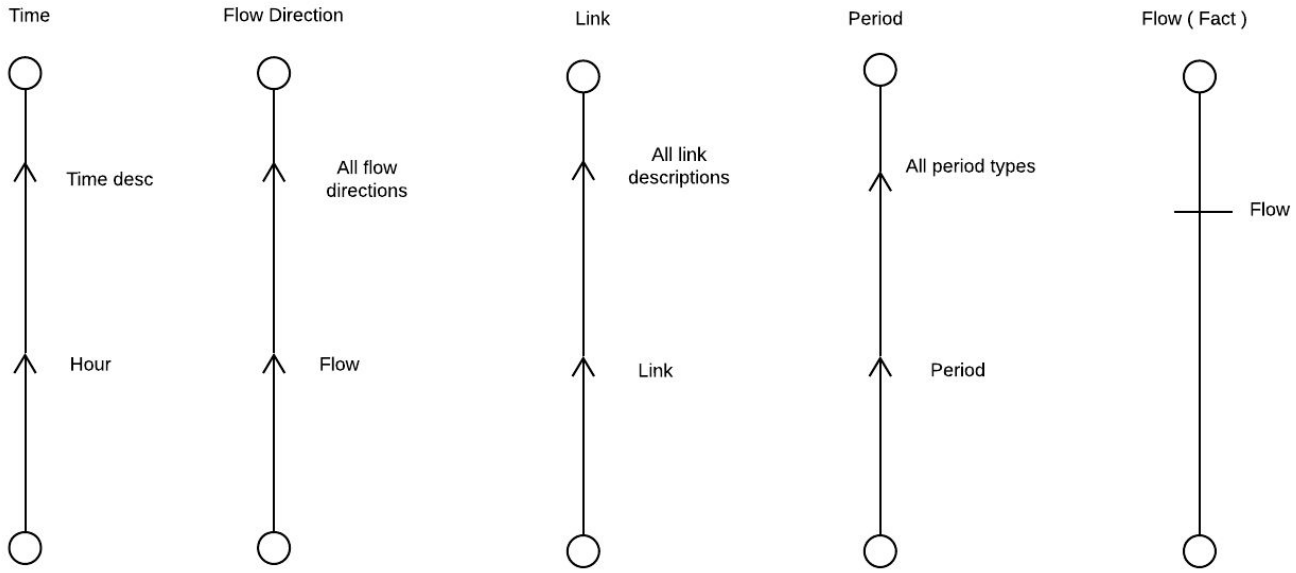
2.0 Data Warehouse Creation

Compared to Database, Data warehouse is better to handle the large data from multiple or single sources, and then analyze, report, integrate transaction data. It is highly normalised and structured which help to ensure the consistency between the systems. The time variant characteristic it possesses presents the variation trend of the traffic flow along the time. Data warehouse is also non volatile which means it is stable as there are no frequently updates.

2.1 Design Thomsen diagram

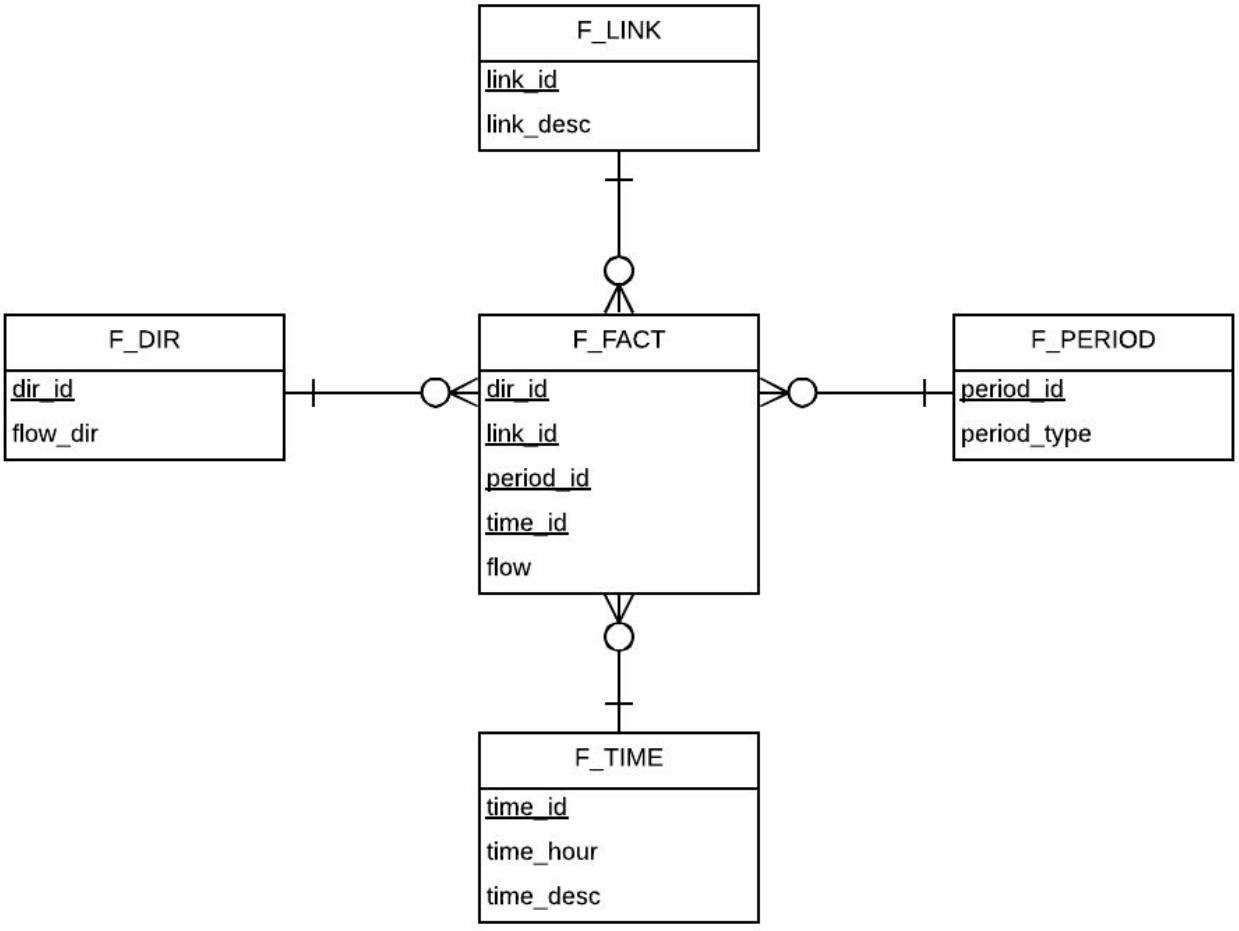
According to the requirements of the Vicroads department, we focus on the measurement of traffic under various conditions which are represented by the flow. As the data we obtained, we group the dimensions as time, link, flow direction and period which are the critical sections of the multi-dimensional model. Each textual description analyse the traffic from a unique perspective. Link dimension has all link

descriptions and link, flow direction dimension has all flow directions and flow. Period dimension has all period types and period. Then we decide to measure the traffic flow in per hour and per part of a day. The design is shown below.



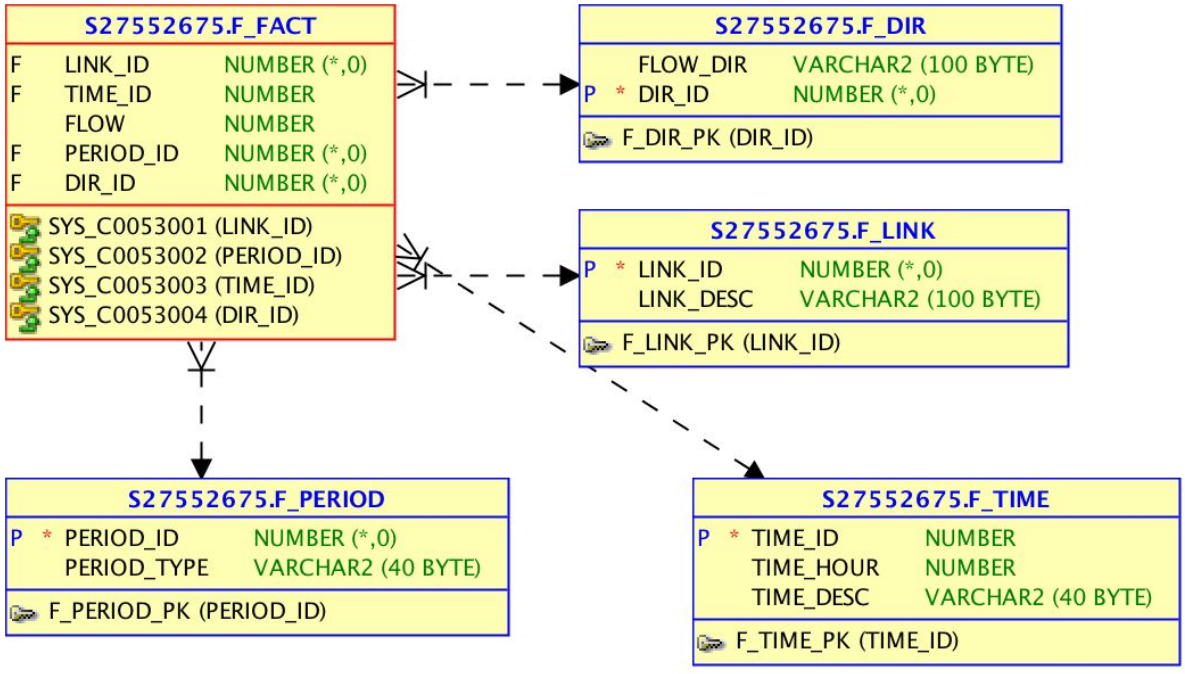
2.2 Design star-schema

We transform conceptual dimensional model directly into star schema. The four dimensions become the dimension table which are F_LINK, F_PERIOD, F_DIR and F_TIME. These tables represent the descriptive information of the attributes in the fact table. F_LINK has link_id and link_desc, F_PERIOD contains period_id and period_type, F_DIR has dir_id and flow_dir, F_TIME contains time_id, time_hour and time_desc. The fact flow has become the corresponding fact table F_FACT. The relationship between the dimension tables and fact table is one-to-optional many. The primary key of the fact table is the composite of foreign keys which come from these dimension tables. These keys are dir_id, link_id, period_id and time_id. The measurement flow is the non key attribute in the fact table. The data in the entity relationship diagram will be transferred into the star schema. The design is shown below.



2.3 Implement The Data Warehouse

Now comes to the stage of implementing the data warehouse. As we have fixed some issues about the data quality, the data which has been reserved are highly integrated and refined. Before we load the data into the data warehouse, we make a brand new table called AGGRTABLE(aggregate table) to hold the aggregates of the hourly flow during a day for each link. Then, in a temporary fact table called UNPIVOTTABLE, the columns of traffic flow for each hour from the aggregate table have been converted into rows, the other columns kept the same as before. Eventually, the temporary fact table would be turned into the actual fact table after some transfer like changing the data type and rename columns. All of the necessary data of the temporary fact table are kept and a series of foreign keys are referenced. A newly created data warehouse is ready to be presented and used in practice. The design is shown below.



2.4 Data Dictionary

The data dictionary is provided to demonstrate the details of the entities and attributes for the star schema.

F_DIR:

Data Item	Data Type	Constraint	Description	Method to obtain
dir_id	Number	Primary Key	Unique identifier for the flow direction	Set sequence then automatically update flow_id.
flow_dir	Varchar2 (100)	Not null	Description of the traffic flow	Selectively insert from FLOW_DIR table by using distinct ().

F_LINK:

Data Item	Data Type	Constraint	Description	Method to obtain
link_id	Number	Primary Key	Unique identifier for the road link being measured	Selectively insert from LINK_INFO table by using distinct ().
link_desc	Varchar2 (100)	Not null	Description of the link road link being measured	Selectively insert from LINK_INFO table by using distinct ().

F_PERIOD:

Data Item	Data Type	Constraint	Description	Method to obtain
period_id	Number	Primary Key	Unique identifier for the type of period	Set sequence then automatically update period_id.
period_type	Varchar2 (100)	Not null	The type of period when traffic was measured	Selectively insert from F_HOUR_INFO table by using distinct ().

F_TIME:

Data Item	Data Type	Constraint	Description	Method to obtain
time_id	Number	Primary Key	Unique identifier for the hour being measured	Set sequence then automatically update period_id.
time_hour	Number		Description of hour	Copy from time_id
time_desc	Varchar2 (100)	Not null	The description of the hour belongs to. 1.Hour 18-23 and 00 to 05 belong is EVENING 2. Hour 06-11 is MORNING 3. Hour 12-17 is AFTERNOON	Manually update.

F_FACT:

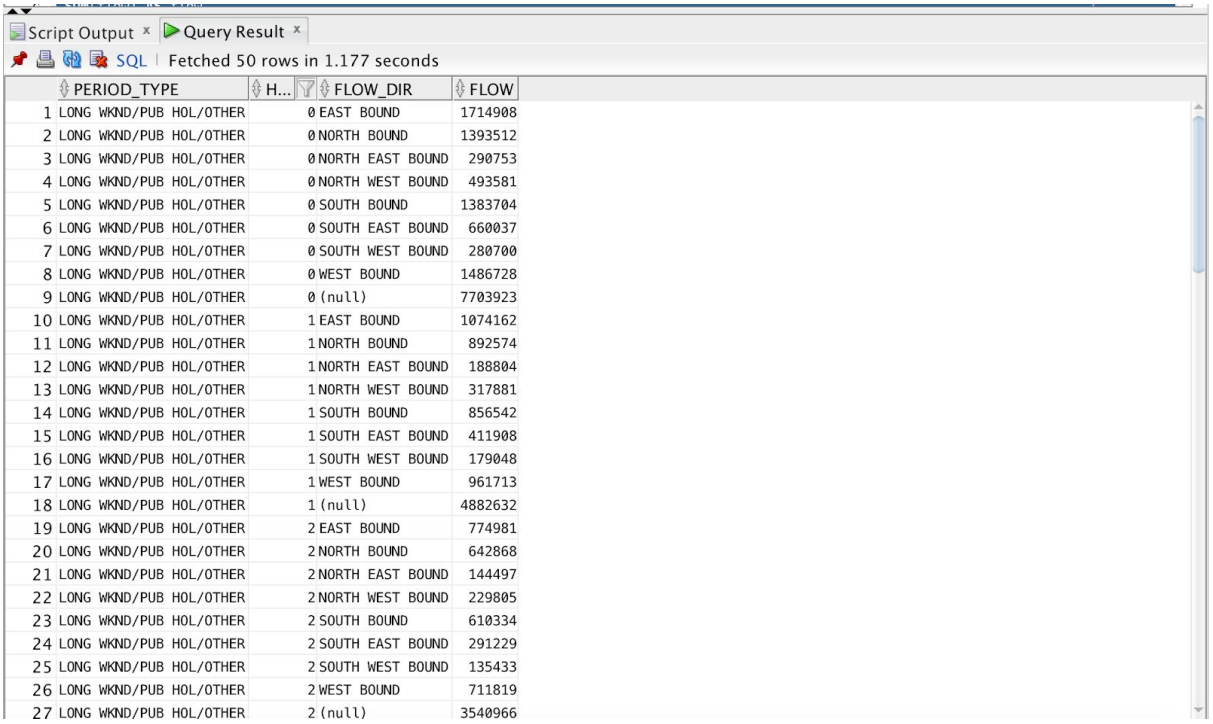
Data Item	Data Type	Constraint	Description	Method to obtain
link_id	Number	Foreign Key	Unique identifier for the road link being measured	Selectively insert from F_LINK table
time_id	Number	Foreign Key	Unique identifier for the hour being measured	Selectively insert from F_TIME table
flow	Number		The traffic flow measured	In F_HOUR_INFO table, convert column F00 - F23 from rows, then select the traffic flow measured
period_id	Number	Foreign Key	Unique identifier for the type of period	Selectively insert from F_PERIOD table
dir_id	Number	Foreign Key	Unique identifier for the flow direction	Selectively insert from F_DIR table

2.5 Create OLAP Queries

1. The first OLAP query is used to check daily traffic flow and average daily flow in each hour (which is calculated by daily flow divided by 24) of each flow direction in each period type in each road link. It mainly focuses on the show of the daily traffic flow and average hourly flow in each road link. Part of the result is shown below.

	LINK_DESC	PERIOD_TYPE	FLOW_DIR	DAILY_FLOW	AVG_DAILY_FLOW
1	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	LONG WKND/PUB HOL/OTHER	EAST BOUND	32684	1361.83
2	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	LONG WKND/PUB HOL/OTHER	WEST BOUND	31876	1328.17
3	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	LONG WKND/PUB HOL/OTHER	(null)	64560	1345
4	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL HOLIDAY	EAST BOUND	24038	1001.58
5	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL HOLIDAY	WEST BOUND	21827	909.46
6	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL HOLIDAY	(null)	45865	955.52
7	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL TERM/NORMAL	EAST BOUND	32327	1346.96
8	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL TERM/NORMAL	WEST BOUND	31920	1330
9	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	SCHOOL TERM/NORMAL	(null)	64247	1338.48
10	ABERDEEN STREET BTWN DEVIATION ROAD & MINERVA ROAD	(null)	(null)	174672	1213
11	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	LONG WKND/PUB HOL/OTHER	EAST BOUND	45488	1895.33
12	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	LONG WKND/PUB HOL/OTHER	WEST BOUND	44347	1847.79
13	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	LONG WKND/PUB HOL/OTHER	(null)	89835	1871.56
14	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL HOLIDAY	EAST BOUND	33462	1394.25
15	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL HOLIDAY	WEST BOUND	30366	1265.25
16	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL HOLIDAY	(null)	63828	1329.75
17	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL TERM/NORMAL	EAST BOUND	42990	1791.25
18	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL TERM/NORMAL	WEST BOUND	42380	1765.83
19	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	SCHOOL TERM/NORMAL	(null)	85370	1778.54
20	ABERDEEN STREET BTWN MINERVA ROAD & SHANNON AVENUE	(null)	(null)	239033	1659.95
21	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	LONG WKND/PUB HOL/OTHER	EAST BOUND	50976	2124
22	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	LONG WKND/PUB HOL/OTHER	WEST BOUND	63518	2646.58
23	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	LONG WKND/PUB HOL/OTHER	(null)	114494	2385.29
24	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	SCHOOL HOLIDAY	EAST BOUND	43877	1828.21
25	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	SCHOOL HOLIDAY	WEST BOUND	43487	1811.96
26	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	SCHOOL HOLIDAY	(null)	87364	1820.08
27	ABERDEEN STREET BTWN PAKINGTON STREET & LATROBE TERRACE	SCHOOL TERM/NORMAL	EAST BOUND	46132	1922.17

2. The second OLAP query is used to check the traffic flow of each flow direction in each hour in each period. It mainly focuses on the present of the total traffic flow in each hour in different period type. Part of the result is shown below.

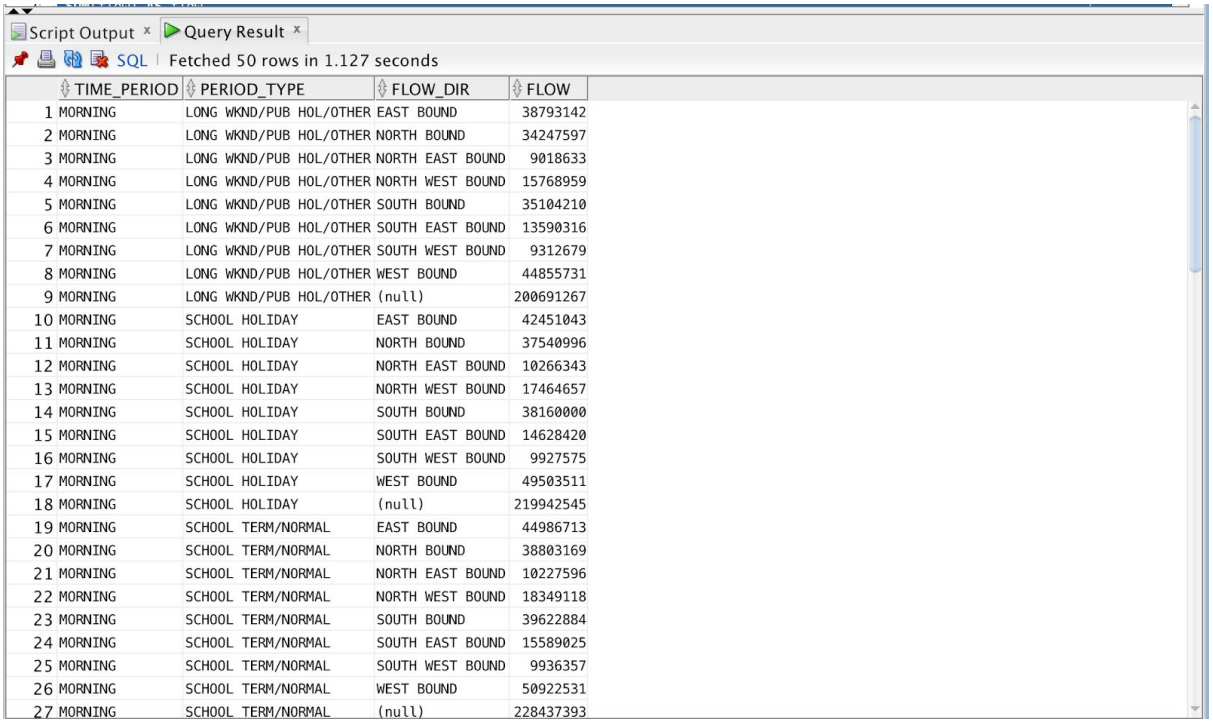


Script Output | Query Result

SQL | Fetched 50 rows in 1.177 seconds

	PERIOD_TYPE	H...	FLOW_DIR	FLOW
1	LONG WKND/PUB HOL/OTHER	0	EAST BOUND	1714908
2	LONG WKND/PUB HOL/OTHER	0	NORTH BOUND	1393512
3	LONG WKND/PUB HOL/OTHER	0	NORTH EAST BOUND	290753
4	LONG WKND/PUB HOL/OTHER	0	NORTH WEST BOUND	493581
5	LONG WKND/PUB HOL/OTHER	0	SOUTH BOUND	1383704
6	LONG WKND/PUB HOL/OTHER	0	SOUTH EAST BOUND	660037
7	LONG WKND/PUB HOL/OTHER	0	SOUTH WEST BOUND	280700
8	LONG WKND/PUB HOL/OTHER	0	WEST BOUND	1486728
9	LONG WKND/PUB HOL/OTHER	0	(null)	7703923
10	LONG WKND/PUB HOL/OTHER	1	EAST BOUND	1074162
11	LONG WKND/PUB HOL/OTHER	1	NORTH BOUND	892574
12	LONG WKND/PUB HOL/OTHER	1	NORTH EAST BOUND	188804
13	LONG WKND/PUB HOL/OTHER	1	NORTH WEST BOUND	317881
14	LONG WKND/PUB HOL/OTHER	1	SOUTH BOUND	856542
15	LONG WKND/PUB HOL/OTHER	1	SOUTH EAST BOUND	411908
16	LONG WKND/PUB HOL/OTHER	1	SOUTH WEST BOUND	179048
17	LONG WKND/PUB HOL/OTHER	1	WEST BOUND	961713
18	LONG WKND/PUB HOL/OTHER	1	(null)	4882632
19	LONG WKND/PUB HOL/OTHER	2	EAST BOUND	774981
20	LONG WKND/PUB HOL/OTHER	2	NORTH BOUND	642868
21	LONG WKND/PUB HOL/OTHER	2	NORTH EAST BOUND	144497
22	LONG WKND/PUB HOL/OTHER	2	NORTH WEST BOUND	229805
23	LONG WKND/PUB HOL/OTHER	2	SOUTH BOUND	610334
24	LONG WKND/PUB HOL/OTHER	2	SOUTH EAST BOUND	291229
25	LONG WKND/PUB HOL/OTHER	2	SOUTH WEST BOUND	135433
26	LONG WKND/PUB HOL/OTHER	2	WEST BOUND	711819
27	LONG WKND/PUB HOL/OTHER	2	(null)	3540966

3. The third OLAP query is used to check the traffic flow of each flow direction in each period type in each time period (morning/afternoon/evening). Part of the result is shown below.



Script Output | Query Result

SQL | Fetched 50 rows in 1.127 seconds

	TIME_PERIOD	PERIOD_TYPE	FLOW_DIR	FLOW
1	MORNING	LONG WKND/PUB HOL/OTHER	EAST BOUND	38793142
2	MORNING	LONG WKND/PUB HOL/OTHER	NORTH BOUND	34247597
3	MORNING	LONG WKND/PUB HOL/OTHER	NORTH EAST BOUND	9018633
4	MORNING	LONG WKND/PUB HOL/OTHER	NORTH WEST BOUND	15768959
5	MORNING	LONG WKND/PUB HOL/OTHER	SOUTH BOUND	35104210
6	MORNING	LONG WKND/PUB HOL/OTHER	SOUTH EAST BOUND	13590316
7	MORNING	LONG WKND/PUB HOL/OTHER	SOUTH WEST BOUND	9312679
8	MORNING	LONG WKND/PUB HOL/OTHER	WEST BOUND	44855731
9	MORNING	LONG WKND/PUB HOL/OTHER	(null)	200691267
10	MORNING	SCHOOL HOLIDAY	EAST BOUND	42451043
11	MORNING	SCHOOL HOLIDAY	NORTH BOUND	37540996
12	MORNING	SCHOOL HOLIDAY	NORTH EAST BOUND	10266343
13	MORNING	SCHOOL HOLIDAY	NORTH WEST BOUND	17464657
14	MORNING	SCHOOL HOLIDAY	SOUTH BOUND	38160000
15	MORNING	SCHOOL HOLIDAY	SOUTH EAST BOUND	14628420
16	MORNING	SCHOOL HOLIDAY	SOUTH WEST BOUND	9927575
17	MORNING	SCHOOL HOLIDAY	WEST BOUND	49503511
18	MORNING	SCHOOL HOLIDAY	(null)	219942545
19	MORNING	SCHOOL TERM/NORMAL	EAST BOUND	44986713
20	MORNING	SCHOOL TERM/NORMAL	NORTH BOUND	38803169
21	MORNING	SCHOOL TERM/NORMAL	NORTH EAST BOUND	10227596
22	MORNING	SCHOOL TERM/NORMAL	NORTH WEST BOUND	18349118
23	MORNING	SCHOOL TERM/NORMAL	SOUTH BOUND	39622884
24	MORNING	SCHOOL TERM/NORMAL	SOUTH EAST BOUND	15589025
25	MORNING	SCHOOL TERM/NORMAL	SOUTH WEST BOUND	9936357
26	MORNING	SCHOOL TERM/NORMAL	WEST BOUND	50922531
27	MORNING	SCHOOL TERM/NORMAL	(null)	228437393

3.0 User Interface Creation

For the interface design, we consider the information Vicroads business requirement and try our best to show all the related information. This interface uses interaction design, allowing users to select the information they want to see easily. As we know that VicRoads is mainly required the flow information in different period type, there is a period type filter that apply to all worksheet is shown on the top right concert. The whole dashboard is used in English, matching between system and real-world. There is no technical terms in the interface, all the words are easy to understand. Furthermore, users can filter something by clicking on the information area and redo it by clicking the area again or blank area of that chart, which follows the user control and freedom rule of Jakob Nielsen's principles. It is simple but shows the information that VicRoads mainly want with efficient and correctness.

In the upper left corner, there is a date of today which is automatically updated to inform the user what date is it today. As mentioned before, in the top right corner, there is a filter that allows Vicroads to select the part of data they mainly want to see quickly. For their business, they focus on the data based on different period types so that they could make better business decisions. Therefore, it is put on the top and it would affect the whole sheet if users select.

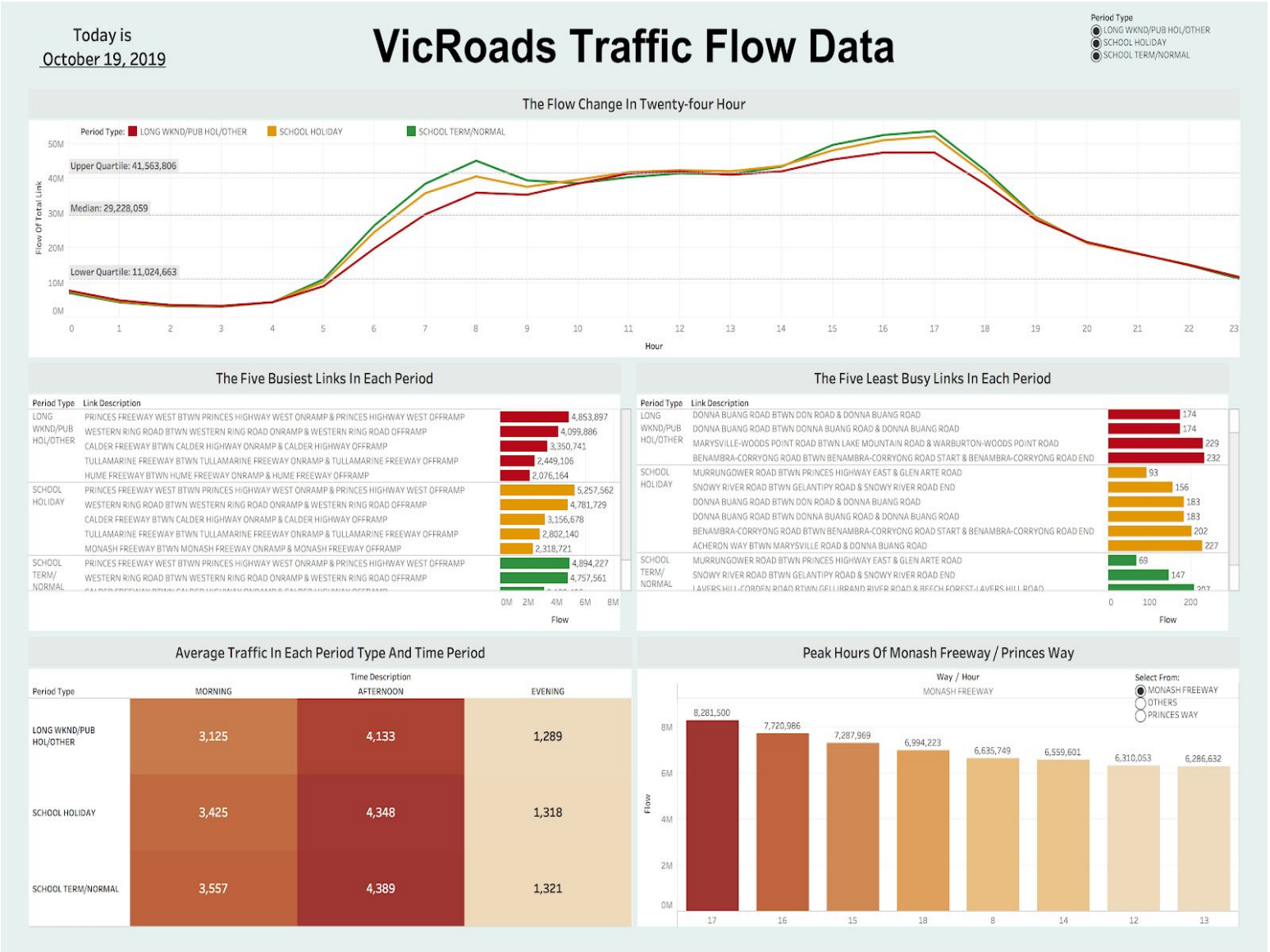
The following is a line chart that shows the trend of total traffic flow of total road links in a day that group by each hour. It would be automatically calculated the quartiles, and show the upper quartile, the median and the lower quartile to help the company to do the data analysis.

Next, there are two charts present the information about the five busiest links in each period and the five least busy links in each period. All information related to period type is used the same colour legends.

The chart in the lower-left corner shows the information classified according to two dimensions, the three period types the company defined and the time period

morning, afternoon and evening. It presents clear by a table and uses the brightness of a single colour to show the number of the flow.

The chart in the lower-right corner shows the two road links that VicRoads most care about, which are Monash Freeway and Princess way. The interface allows users to select the way they want and show information about some busy flows, like in which hour and the number of the flow.



Conclusion

This report summarises the overall design for the traffic flow data analysis and how they will be used in practice. The objective of the traffic data design is to develop a multi-dimensional model and a user interface for data visualisation which could help the VicRoads department provide a better usage for the road. We have designed an entity-relationship diagram which contains three tables, a star schema with four dimension tables and one fact table, a data dictionary which defines all of the entities and attributes necessary. An organized data warehouse has been created after the ETL process, last but not least, an interactive user interface has been designed to present the data in terms of the chart and colour. A sequence of SQL source code has been appended to demonstrate how these data will be processed in the design.

Appendix

-----IMPORT RAW DATA:

```
create table Ass3
(
FLOW_ID int,
LINK_ID int,
LINK_DESC VARCHAR2(100),
FLOW_DIR VARCHAR2(40),
PERIOD_TYPE VARCHAR2(40),
F00     int,
F01     int,
F02     int,
F03     int,
F04     int,
F05     int,
F06     int,
F07     int,
F08     int,
F09     int,
F10     int,
F11     int,
F12     int,
F13     int,
F14     int,
F15     int,
F16     int,
F17     int,
F18     int,
F19     int,
F20     int,
F21     int,
F22     int,
F23 int
);
```

-----F_LINK_INFO TABLE CREATE:

```
DROP TABLE F_LINK_INFO;

CREATE TABLE F_LINK_INFO(
link_id INT PRIMARY KEY,
link_desc VARCHAR2(100));
```

```
insert into F_LINK_INFO(link_id,link_desc)
SELECT DISTINCT(link_id),LINK_DESC FROM Ass3;
```

-----F_INFO TABLE CREATE:

```
DROP TABLE  F_INFO;

CREATE TABLE F_INFO(
flow_id INT PRIMARY KEY,
link_id INT,
flow_dir VARCHAR2(100)
);

insert into F_INFO(FLOW_ID,FLOW_DIR,LINK_ID)
SELECT DISTINCT(FLOW_ID),FLOW_DIR,LINK_ID FROM Ass3;

ALTER TABLE F_INFO
ADD FOREIGN KEY (LINK_id) REFERENCES F_LINK_INFO(LINK_id);
```

-----F_HOUR_INFO TABLE CREATE:

```
DROP TABLE F_HOUR_INFO;

CREATE TABLE F_HOUR_INFO
(
FLOW_ID int ,
PERIOD_TYPE VARCHAR2(100) ,

F00     int,
F01     int,
F02     int,
F03     int,
F04     int,
F05     int,
F06     int,
F07     int,
F08     int,
F09     int,
F10     int,
F11     int,
F12     int,
F13     int,
F14     int,
F15     int,
F16     int,
F17     int,
```

```
F18     int,
F19     int,
F20     int,
F21     int,
F22     int,
F23 int
);


insert into F_HOUR_INFO(FLOW_ID,PERIOD_TYPE,F00,
F01     ,
F02     ,
F03     ,
F04     ,
F05     ,
F06     ,
F07     ,
F08     ,
F09     ,
F10     ,
F11     ,
F12     ,
F13     ,
F14     ,
F15     ,
F16     ,
F17     ,
F18     ,
F19     ,
F20     ,
F21     ,
F22     ,
F23
)
SELECT FLOW_ID,PERIOD_TYPE, F00,
F01     ,
F02     ,
F03     ,
F04     ,
F05     ,
F06     ,
F07     ,
F08     ,
F09     ,
F10     ,
F11     ,
```

```
F12     ,
F13     ,
F14     ,
F15     ,
F16     ,
F17     ,
F18     ,
F19     ,
F20     ,
F21     ,
F22     ,
F23
FROM Ass3;

drop sequence record_id_seq;

create sequence record_id_seq
start with 1
INCREMENT by 1;

alter table F_HOUR_INFO add RECORD_ID INT;

update F_HOUR_INFO set RECORD_ID=record_id_seq.nextval;

ALTER TABLE F_HOUR_INFO ADD PRIMARY KEY (record_id);

ALTER TABLE F_HOUR_INFO
ADD FOREIGN KEY (FLOW_id) REFERENCES F_INFO(FLOW_id);
```

-----ETL:

-----1.clean null

```
delete from f_hour_info where flow_id in
(select flow_id from f_info where link_id in
(select link_id from f_link_info where link_desc is null));

delete from f_info where link_id in
(select link_id from f_link_info where link_desc is null);

delete from f_link_info where link_desc is null;
```

-----2. Change all text to upper case

```
update F_HOUR_INFO set PERIOD_TYPE=upper(PERIOD_TYPE);
update F_INFO set FLOW_DIR=upper(FLOW_DIR);
update F_LINK_INFO set LINK_DESC=upper(LINK_DESC);
```

-----drop table ass3 incase no enough space

```
drop table ass3;
```

-----CREATE AGGRTABLE TABLE:

```
DROP Table AGGRTABLE ;

create table aggrTable AS SELECT
distinct(f_info.LINK_ID),f_hour_info.PERIOD_type,f_info.flow_dir,
SUM(F00) AS "S00",
SUM(F01) AS "S01",
SUM(F02)AS "S02",
SUM(F03)AS "S03",
SUM(F04)AS "S04",
SUM(F05)AS "S05",
SUM(F06)AS "S06",
SUM(F07)AS "S07",
SUM(F08)AS "S08",
SUM(F09)AS "S09",
SUM(F10)AS "S10",
SUM(F11)AS "S11",
SUM(F12)AS "S12",
SUM(F13)AS "S13",
SUM(F14)AS "S14",
SUM(F15)AS "S15",
SUM(F16)AS "S16",
SUM(F17)AS "S17",
SUM(F18)AS "S18",
SUM(F19)AS "S19",
SUM(F20)AS "S20",
SUM(F21)AS "S21",
SUM(F22)AS "S22",
SUM(F23)AS "S23" FROM f_hour_info FULL JOIN f_info on
f_hour_info.FLOW_ID=f_info.FLOW_ID
group by f_info.LINK_ID,PERIOD_TYPE,f_info.flow_dir ;
```

-----CREATE F_PERIOD TABLE:

```
DROP TABLE F_PERIOD;
```

```
CREATE TABLE F_PERIOD(
PERIOD_ID INT,
PERIOD_type VARCHAR2(40));

INSERT INTO F_PERIOD (PERIOD_type)
SELECT DISTINCT(PERIOD_TYPE) FROM F_HOUR_INFO;

DROP SEQUENCE PERIOD_ID_SEQ;

create sequence PERIOD_ID_SEQ
start with 1
INCREMENT by 1;

UPDATE F_PERIOD SET period_id=PERIOD_ID_SEQ.NEXTVAL;

ALTER TABLE F_PERIOD
ADD PRIMARY KEY (PERIOD_ID);
```

-----CREATE F_LINK TABLE:

```
DROP TABLE F_LINK;

CREATE TABLE F_LINK AS SELECT * FROM f_LINK_info;

ALTER TABLE F_LINK ADD PRIMARY KEY(LINK_ID);
```

-----CREATE F_DIR TABLE:

```
DROP TABLE F_DIR;

create table F_DIR AS SELECT DISTINCT(FLOW_DIR) FROM F_INFO;

ALTER TABLE F_DIR ADD DIR_ID INT;
DROP SEQUENCE DIR_ID_SEQ;
create sequence DIR_ID_SEQ
start with 1
INCREMENT by 1;
update F_DIR set DIR_ID=DIR_ID_SEQ.nextval;
ALTER TABLE F_DIR ADD PRIMARY KEY(DIR_ID);
```

-----CREATE F_TIME TABLE:
-----1. CREATE UNPIVOT TABLE,
-----2. THEN SELECT DISTINCT(HOUR_ID)

-----1.

```
DROP TABLE UNPIVOTTABLE;

CREATE TABLE UNPIVOTTABLE AS select *
  from AGGRTABLE
unpivot
(
  FLOW
    for TIME_ID in (S00 AS 00,
    S01 AS 01,
    S02 AS 02,
S03     AS 03,
S04     AS 04,
S05     AS 05,
S06     AS 06,
S07     AS 07,
S08     AS 08,
S09     AS 09,
S10     AS 10,
S11     AS 11,
S12     AS 12,
S13     AS 13,
S14     AS 14,
S15     AS 15,
S16     AS 16,
S17     AS 17,
S18     AS 18,
S19     AS 19,
S20     AS 20,
S21     AS 21,
S22     AS 22,
S23 AS 23
)

)
order by LINK_ID,PERIOD_TYPE,TIME_ID,FLOW_DIR;
```

-----2.

```
DROP TABLE F_TIME;
```

```
CREATE TABLE F_TIME AS SELECT DISTINCT(TIME_ID),TIME_ID as TIME_HOUR
FROM UNPIVOTTABLE ORDER BY TIME_ID;

ALTER TABLE F_TIME ADD PRIMARY KEY(TIME_ID);

ALTER TABLE F_TIME ADD TIME_DESC VARCHAR2(40);

UPDATE F_TIME SET TIME_DESC='EVENING' WHERE time_id BETWEEN 18 AND 23;

UPDATE F_TIME SET TIME_DESC='EVENING' WHERE time_id BETWEEN 0 AND 5;

UPDATE F_TIME SET TIME_DESC='MORNING' WHERE time_id BETWEEN 6 AND 11;

UPDATE F_TIME SET TIME_DESC='AFTERNOON' WHERE time_id BETWEEN 12 AND
17;
```

-----CREATE F_FACT TABLE:

-----BY CLEANING AND CHANGING THE UNPIVOT TABLE

```
DROP TABLE F_FACT;

ALTER TABLE UNPIVOTTABLE ADD PERIOD_ID INT;

UPDATE UNPIVOTTABLE SET period_id=(SELECT PERIOD_ID FROM f_period WHERE
f_period.PERIOD_TYPE=UNPIVOTTABLE.PERIOD_TYPE);

ALTER TABLE UNPIVOTTABLE ADD DIR_ID INT;

UPDATE UNPIVOTTABLE SET DIR_ID=(SELECT DIR_ID FROM F_DIR WHERE
F_DIR.FLOW_DIR=UNPIVOTTABLE.FLOW_DIR);

ALTER TABLE UNPIVOTTABLE DROP COLUMN PERIOD_TYPE;

ALTER TABLE UNPIVOTTABLE DROP COLUMN FLOW_DIR;

ALTER TABLE UNPIVOTTABLE
ADD FOREIGN KEY (LINK_ID) REFERENCES F_LINK(LINK_ID);

ALTER TABLE UNPIVOTTABLE
ADD FOREIGN KEY (PERIOD_ID) REFERENCES F_PERIOD(PERIOD_ID);

ALTER TABLE UNPIVOTTABLE
ADD FOREIGN KEY (TIME_ID) REFERENCES F_TIME(TIME_ID);
```

```
ALTER TABLE UNPIVOTTABLE
ADD FOREIGN KEY (DIR_ID) REFERENCES F_DIR(DIR_ID);


ALTER TABLE UNPIVOTTABLE RENAME TO F_FACT;
```

-----OLAP:

-----1.

```
SELECT link_desc, period_type ,flow_dir,
 SUM(flow) AS daily_flow, round(avg(flow),2) AS avg_daily_flow
FROM F_fact f join f_link l on f.link_id=l.link_id
join f_period p on f. period_id=p.period_id
join f_dir d on f.dir_id=d.dir_id
GROUP BY link_desc , ROLLUP(period_type,flow_dir)
ORDER BY link_desc,period_type,flow_dir;
```

-----2.

```
SELECT  period_type ,TIME_HOUR as HOUR,flow_dir,
 SUM(flow) AS flow
FROM F_fact f
join f_period p on f. period_id=p.period_id
join f_dir d on f.dir_id=d.dir_id
JOIN f_TIME t on f.time_id = t.time_id
GROUP BY  period_type, TIME_HOUR, CUBE(flow_dir)
ORDER BY period_type,TIME_HOUR,flow_dir;
```

----3.

```
SELECT  time_desc as time_period, period_type ,flow_dir,
 SUM(flow) AS flow
FROM F_fact f
join f_period p on f.period_id=p.period_id
join f_dir d on f.dir_id=d.dir_id
join f_time t on f.time_id=t.time_id
GROUP BY  time_desc, ROLLUP(period_type,flow_dir)
ORDER BY DECODE（
time_desc,'MORNING','1','AFTERNOON','2','EVENING','3'),period_type ,flow_dir;
```