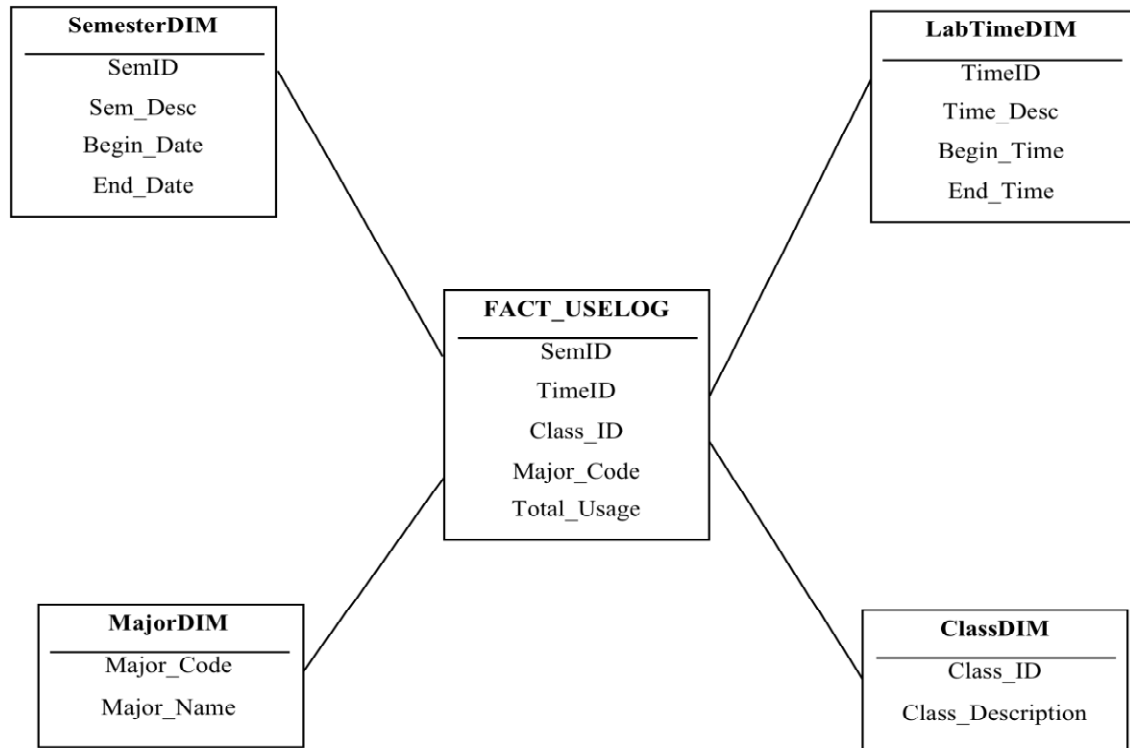


USELOG Case Study

1. USELOG star schema

The star schema of the second case study (USELOG system) is shown as follows.



The steps to implement this star schema in tables are:

- For each dimension, create a table. There are three methods to create a table for a dimension:
 - If the dimension is exactly identical to a table in the operational database, you can directly copy the table from the operational database to your data warehouse, using the following command:

```
CREATE TABLE <<dimension_table>>
AS
SELECT *
FROM <<account>>.<<table_name>>;
```

For example, assuming Class_dim Dimension is totally taken from the operational table called dtaniar.class:

```
CREATE TABLE class_dim
AS
SELECT *
FROM dtaniar.class;
```

- If the dimension consists of some attributes from an existing table in the operational database, then you can extract those attributes, using the following commands:

```
CREATE TABLE <<dimension_table>>
AS
SELECT DISTINCT <<attribute_names>>
FROM <<account>>.<<table_name>>;
```

For example, assuming that the operational table dtaniar.major has many attributes, and we only need to attributes, namely major_code and major_name, hence we can extract these attributes:

```
CREATE TABLE major_dim
AS
SELECT DISTINCT major_code, major_name
FROM dtaniar.major;
```

Using the previous case study on Student Enrolment, for the campus_dim dimension, we extract the attribute ocampus from dtaniar.offering:

```
CREATE TABLE campus_dim
AS
SELECT DISTINCT ocampus
FROM dtaniar.offering;
```

- If the dimension needs records that do not explicitly exist in the operational database, then you need to manually create the table and then insert the records.

For example, if we want to create a dimension table to store the time period of a day (e.g. day, afternoon, night), whereas the operational has a table that stores transactions with the actual transaction time, then we need to create the dimension table manually. It is not possible to either copy or extract the data from the operational database, because the information we would like to store in the dimension (e.g. morning, afternoon, night) does not exist in the operational database.

```
create table labtimeDIM
(TimeID      number,
Time_Desc   varchar2(15),
begin_time  date,
end_time    date);
```

```
insert into labtimeDIM
values(1, 'morning',
      to_date('06:01', 'HH24:MI'),
      to_date('12:00', 'HH24:MI'));
```

```
insert into labtimeDIM
values(2, 'afternoon',
      to_date('12:01', 'HH24:MI'),
      to_date('18:00', 'HH24:MI'));
```

```
insert into labtimeDIM
values(3, 'night',
      to_date('18:01', 'HH24:MI'),
      to_date('06:00', 'HH24:MI'));
```

- The fact requires semID and timeID, but the operational database (e.g. the Uselog table) does not have semID and timeID. The operational database has the actual date and the actual time in the uselog table. Hence, we need to create a temp_fact, which is an intermediate step before creating the final fact table.

Additionally, the fact requires total_usage (which is actually number of students in the lab). The operational database does not have number of students, but has the actual students. So, we need to create a temp_fact that contains attributes date and time from uselog (which are later to be converted into semester and time ID), class ID and major code, as well as student ID.

- Once the temp_fact is created, we need to convert from date to semID, and from time to timeID. We can do it by adding two new columns in the temp_fact – call these two attributes semID and timeID. At this point, temp_fact has 7 attributes: date, time, class ID, major code, student ID, and the two newly added columns (semID and timeID)
- For records that have dates between 1-Jan to 15-Jul, then fill in the semID column with 'S1' to indicate semester 1 (assuming that semester 1 is between 1-Jan and 15-Jul). Do the same for semester 2 records.
- For records that have time between 06:01 and 12:00, fill in the timeID column with 1 to indicate morning (assuming that morning/code 1, is for time between 6:01am and 12pm). Do the same for afternoon and night.
- At this point, temp_fact will have the two columns (semID and timeID) filled in with the correct codes.
- Then finally we are able to create a final fact table by taking this temp_fact and then aggregate and count the studentID (note that in the fact table, we do not need to keep the date and time anymore, because we already have semID and timeID).