Solutions

Tutorial 3 – Star Schemas

PART I: The USELOG Case Study

Description

University Computer Lab's director keeps track of the lab usage, measured by the number of students using the lab. This particular function is very important for budgeting purposes. The computer lab director assigns you the task of developing a small Data Warehouse in which to keep track of the lab usage statistics. The main requirements for this database are to:

a. Show the usage numbers by different time periods (e.g. morning, afternoon, night)

b. Show the usage numbers by time period (e.g. morning, afternoon, night), by major, and by student's class

c. Compare the usage numbers for different majors and semesters (e.g. semester 1, semester 2).

Use the provided database that includes the following tables: USELOG, STUDENT, MAJOR, and CLASS

USELOG contains the student access data
USELOG (Log_Date, Log_Time, Student_ID, Act)

STUDENT is a table containing student data
STUDENT (Student_ID, Sex, Full/Part, Type, Class_ID, Major_Code)

MAJOR is a table containing major data
MAJOR (Major_Name, Major_Code)

CLASS is a table containing class data
CLASS (Class_Description, Class_ID)

Tasks

The above case study has been discussed in the lecture this week. Given the requirements above, complete the following:

1. Create a star schema for the charter data.
2. Define the dimensions and attributes for the Uselog star schema.
3. Write the SQL statements for the implementation of the star schema. The following operational databases have been provided for you:

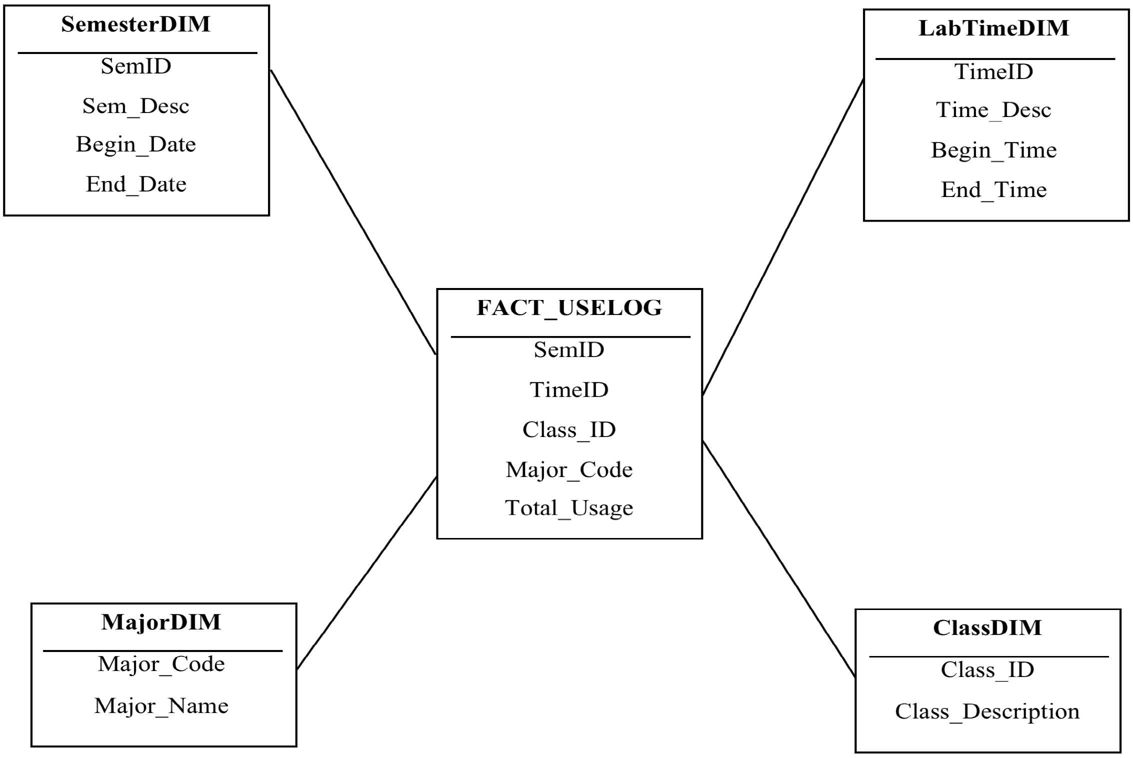
   `dw.Class`: table that stores information about classification ids and descriptions
   `dw.Major`: table that stores information about major codes and descriptions
   `dw.Student`: table that stores information about students as described above
   `dw.Uselog`: table that stores information about lab usage as described above

   You do not need to copy these four tables (`dw.Class`, `dw.Major`, `dw.Student`, and `dw.Uselog`) into your account. You can just simply use these tables.

4. Write the SQL statements to produce the following reports:
   a. Show the usage numbers by different time periods (e.g. morning, afternoon, night)
   b. Show the usage numbers by time period (e.g. morning, afternoon, night), by major, and by student's class
   c. Show the usage numbers for different majors and semesters (e.g. semester 1, semester 2).

Solutions

Tasks: Q1 and Q2: Star Schema



Task: Q3 The SQL to Create the Dimensions and Fact

```
--first create the dimensions
--create semester dimension
create table semesterDIM
(SemID      varchar2(10),
 Sem_Desc   varchar2(20),
 begin_date date,
 end_date   date);

-- create time dimension (note do not use time as a
-- table name, it is a reserve keyword)
create table labtimeDIM
(TimeID     number,
Time_Desc   varchar2(15),
begin_time  date,
end_time    date);
```

```
-- create major and class dimensions
create table majorDIM as
select * from dw.major;

create table classDIM as
select * from dw.class;

-- populate semester dimension
-- (the begin and end date can be changed)

insert into semesterDIM values ('S1', 'Semester1',
to_date('01-JAN', 'DD-MON'), to_date('15-JUL', 'DD-MON'));

insert into semesterDIM values ('S2', 'Semester2',
to_date('16-JUL', 'DD-MON'), to_date('31-DEC', 'DD-MON'));

--populate labtime dimension
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'));


-- secondly, create a temp table to extract from uselog table
create table tempfact_uselog as
select U.log_date , U.log_time,
U.student_ID, S.class_id, S.major_code
from dw.uselog U, dw.student S
where U.student_id = S.student_id;

-- add a column in the tempfact table to store timeid
-- (cannot directly do this in the tempfact table because
-- log_time was of DATE type and timeid is of NUMBER type).

alter table tempfact_uselog
add (timeid number);

update tempfact_uselog
set timeid = 1
where  to_char(log_time, 'HH24:MI') >= '06:01'
and to_char(log_time, 'HH24:MI') <='12:00';

update tempfact_uselog
set timeid = 2
where  to_char(log_time, 'HH24:MI') >= '12:01'
and to_char(log_time, 'HH24:MI') <='18:00';

-- note that we use OR in the last update statement to
-- include the time between 18:01 and 06:00.
update tempfact_uselog
set timeid = 3
where to_char(log_time, 'HH24:MI') >= '18:01'
or to_char(log_time, 'HH24:MI') <='06:00';
```

```
-- alternatively, you may want to update timeid=3
-- for all other records where the time_id is still empty
      -- update tempfact_uselog
      -- set timeid = 3
      -- where timeid is NULL;

-- add a column in the tempfact_uselog table to store semid
-- (cannot directly do this in the test table because
-- log_date was of DATE type and semid is of VARCHAR type.)

alter table tempfact_uselog
add (semid varchar2(10));

-- populate the new attribute semid by summarizing
-- the date(log_date)

update tempfact_uselog
set semid = 'S1'
where to_char(log_date, 'MMDD') >= '0101'
and to_char(log_date, 'MMDD') <= '0715';

update tempfact_uselog
set semid = 'S2'
where to_char(log_date, 'MMDD') >= '0716'
and to_char(log_date, 'MMDD') <= '1231';


-- Now, create the fact table,
-- make sure to include the TOTAL aggregate.
-- This is an aggregate table of the earlier tempfact table.

create table fact_uselog as
select t.semid, t.timeid, t.class_id,
       t.major_code, count(t.student_id) as total_usage
from tempfact_uselog t
group by t.semid, t.timeid, t.class_id, t.major_code;
```

Task: Q4 The Reports

a) Show the usage numbers by different time periods (e.g. morning, afternoon, night)

```
select u.timeid, l.time_desc, sum(u.total_usage) as usage_numbers
from fact_uselog u, labtimedim l
where u.timeid = l.timeid
group by u.timeid, l.time_desc
order by u.timeid;
```

```
    TIMEID TIME_DESC       USAGE_NUMBERS
---------- --------------- -------------
         1 morning                 39921
         2 afternoon               48261
         3 night                   20079
```

b) Show the usage numbers by time period (e.g. morning, afternoon, night), by major, and by student's class)

```
select u.timeid, l.time_desc, u.major_code, m.major_name, u.class_id,
       c.class_description, sum(u.total_usage) as usage_numbers
from fact_uselog u, labtimedim l, majordim m, classdim c
where u.timeid = l.timeid
and u.major_code = m.major_code
and u.class_id = c.class_id
group by u.timeid, l.time_desc, u.major_code, m.major_name,
         u.class_id, c.class_description;
```

```
TIMEID TIME_DESC MAJOR_CO MAJOR_NAME                        CLASS_ID CLASS_DES USAGE_NUMBERS
------ --------- -------- -------------------------------- -------- --------- -------------
     3 night     ENVI     Environmental Science Technology JR       Junior    3
     2 afternoon SEd1     Special Education                FR       Freshman  3
     2 afternoon HEAL     Health Education                 JR       Junior    6
     2 afternoon SSTU     Social Studies                   SO       Sophomore 3
     3 night     RIPT     Recording Industry Production    SO       Sophomore 3

773 rows selected.
```

c) Show the usage numbers for different majors and semesters (e.g. semester 1, semester 2).

```
select u.major_code, m.major_name, u.semid, s.sem_desc,
       sum(u.total_usage) as usage_numbers
from fact_uselog u, majordim m, semesterdim s
where u.major_code = m.major_code
and u.semid = s.semid
group by u.major_code, m.major_name, u.semid, s.sem_desc
order by u.major_code;
```

```
MAJOR    MAJOR_NAME                      SEMID   SEM_DESC     USAGE_NUMBERS
-------  ------------------------------  ------- -----------  -------------
UNIL     Business Degree/ Liberal Arts   S1      Semester1           18
UNIL     Business Degree/ Liberal Arts   S2      Semester2           18
VTE      Vocational Tech Education       S1      Semester1           69
VTE      Vocational Tech Education       S2      Semester2          177
WELL     Wellness                        S1      Semester1           27
WELL     Wellness                        S2      Semester2           24


207 rows selected.
```

PART II: The ROBCOR Aviation Charters Case Study

Description

ROBCOR, Inc. provides “on demand” aviation charters, using a mix of different airplane and airplane types. Because ROBCOR, Inc., has grown rapidly, it has hired you to be its first database manager. Your first and critical assignment is to develop a decision support system to analyze the charter data.

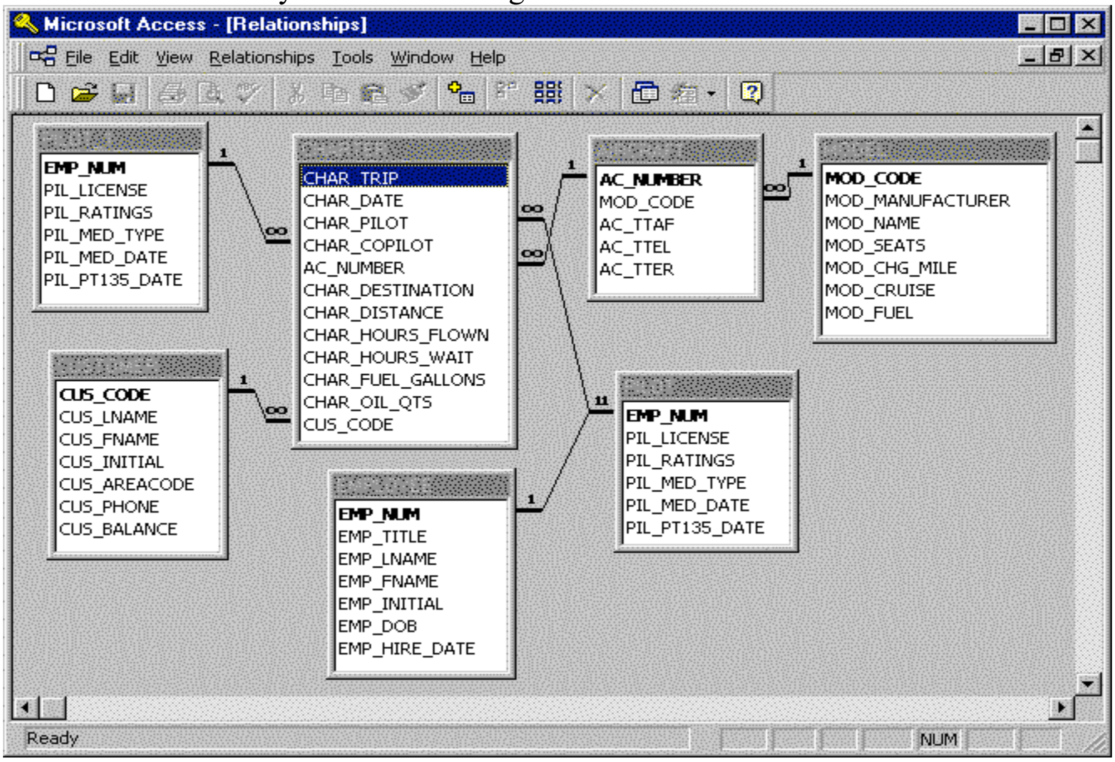
The charter operations manager wants to be able to analyze charter data such as total hours flown, total fuel used, and total revenue (charter distance x model charge per mile). She would also like to be able to drill-down by pilot, airplane model, and time periods. The main requirements for this database are to:

a. Show the total revenue each month/year

b. Show the total hours flown by each pilot

c. Show the total fuel used by each airplane model.

The database currently has the following tables:



EMP_NUM
PIL_LICENSE
PIL_RATINGS
PIL_MED_TYPE
PIL_MED_DATE
PIL_PT135_DATE

CHAR_TRIP
CHAR_DATE
CHAR_PILOT
CHAR_COPILOT
AC_NUMBER
CHAR_DESTINATION
CHAR_DISTANCE
CHAR_HOURS_FLOWN
CHAR_HOURS_WAIT
CHAR_FUEL_GALLONS
CHAR_OIL_QTS
CUS_CODE

AC_NUMBER
MOD_CODE
AC_TTAF
AC_TTEL
AC_TTER

MOD_CODE
MOD_MANUFACTURER
MOD_NAME
MOD_SEATS
MOD_CHG_MILE
MOD_CRUISE
MOD_FUEL

CUS_CODE
CUS_LNAME
CUS_FNAME
CUS_INITIAL
CUS_AREACODE
CUS_PHONE
CUS_BALANCE

EMP_NUM
EMP_TITLE
EMP_LNAME
EMP_FNAME
EMP_INITIAL
EMP_DOB
EMP_HIRE_DATE

EMP_NUM
PIL_LICENSE
PIL_RATINGS
PIL_MED_TYPE
PIL_MED_DATE
PIL_PT135_DATE

The tables can be copied from the dw account, using:

```
Select * from dw.<table_name>;
```

Or

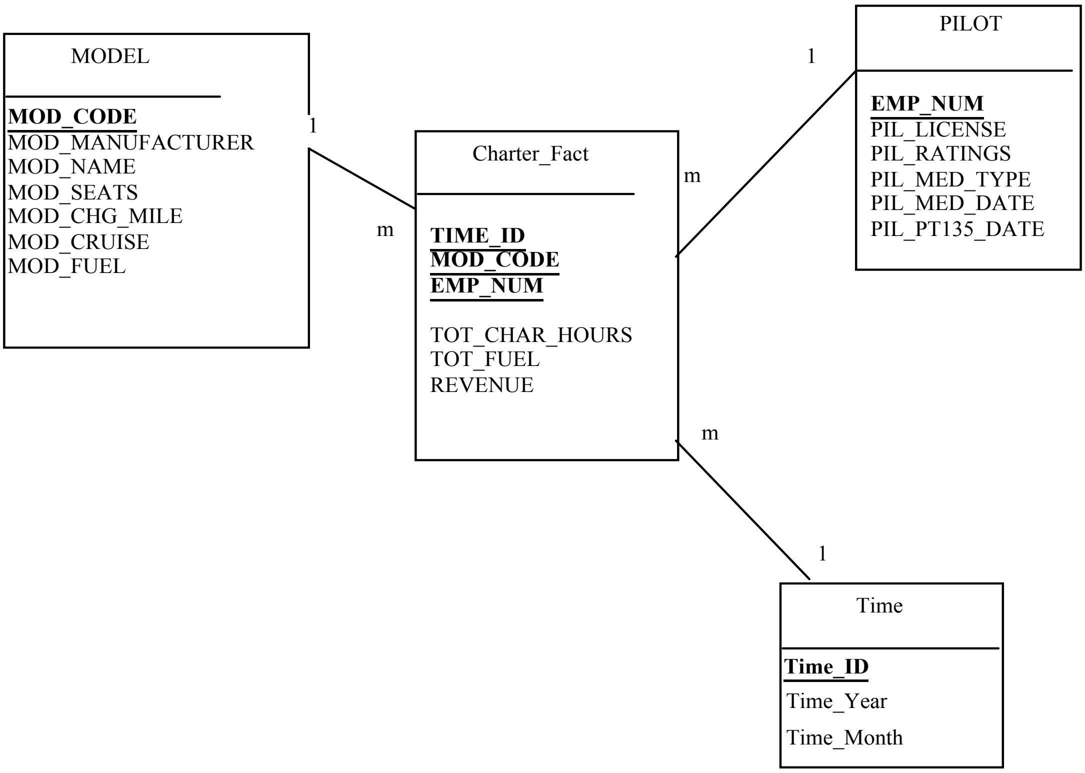
```
Create Table <your_table_name> As
Select …
From dw.<table_name>
Where …
```

Given these requirements, complete the following:

1. Create a star schema for the charter data.
2. Define the dimensions and attributes for the charter operation's star schema.
3. Define the SQL statements for the implementation of the star schema.
4. Write the SQL statements to produce the following reports:
    a. Show the total revenue each year
    b. Show the total hours flown by each pilot
    c. Show the total fuel used by each aircraft model

Solutions

Tasks: Q1 and Q2: Star Schema



Task: Q3 The SQL to Create the Dimensions and Fact

```
--First create the dimensions
--create time dimension
Create table Time As
Select Distinct to_char(char_date, 'YYYYMM') as Time_ID,
                to_char(char_date, 'Month') as Time_Month,
                to_char(char_date, 'YYYY') as Time_Year
from dw.Charter;

--copy model and pilot dimensions
create table model as
select * from dw.model;

create table pilot as
select * from dw.pilot;
```

```
--Second, create the Charter_fact (the fact table) table
create table charter_fact as
select C.Char_Pilot as EMP_Num,
       M.Mod_Code,
       to_char(C.Char_Date, 'YYYYMM') as Time_ID,
       sum(C.Char_Hours_Flown) as Tot_Char_Hours,
       sum(C.Char_Fuel_Gallons) as Tot_Fuel,
       sum(C.Char_Distance * M.Mod_chg_mile) as Revenue
from   dw.Charter C, dw.Model M, dw.Aircraft A
where  C.AC_Number=A.AC_Number and A.Mod_Code=M.Mod_Code
group by C.Char_Pilot, M.Mod_Code, to_char(C.Char_Date, 'YYYYMM');
```

Task: Q4 The Reports

a. Show the total revenue each year

```
select t.time_year as Year, sum(c.revenue) as total_revenue
from charter_fact c, time t
where c.time_id = t.time_id
group by t.time_year
order by t.time_year;
```

```
YEAR TOTAL_REVENUE
---- -------------
1994     452579.08
1995     472881.69
1996     450564.01
1997     217264.37
```

b. Show the total hours flown by each pilot

```
select EMP_NUM, sum(TOT_CHAR_HOURS) as total_hours_flown
from  charter_fact
group by EMP_NUM
order by EMP_NUM;
```

```
EMP_NUM    TOTAL_HOURS_FLOWN
---------- -----------------
       101             672.7
       104             716.5
       105               739
       106             965.1
       109             721.5
```

c. Show the total fuel used by each aircraft model.

```
select m.mod_code as model_code, m.mod_name as model_name,
      sum(c.tot_fuel) as total_fuel_used
from  charter_fact c, model m
where m.mod_code = c.mod_code
group by m.mod_code, m.mod_name;
```

```
MODEL_CODE MODEL_NAME           TOTAL_FUEL_USED
---------- -------------------- ---------------
PA23-250   Aztec                          31265
PA31-350   Navajo Chieftain             83790.5
C-90A      KingAir                      61708.4
```