Tutorial 2
Student Enrollment Case Study

Description

This tutorial will use the Student Case Study consisting of STUDENT2, SUBJECT2, OFFERING2 and ENROLLMENT2 tables.

The STUDENT2 table contains the personal details of students. The SUBJECT2 table keeps all the records of the available subjects. However, a single subject may have many offerings. This means that the same subject may be offered several times in a year as well as on different modes and campuses. Therefore, the OFFERING2 table stores the information of subjects being offered in particular semesters, etc. Basically, after a student selects a subject she/he is interested, she/he then needs to choose which semester or which campus she/he wants to take that subject. Once this is done, the details are stored in the ENROLLMENT2 table that will also store the Student ID and Offering ID as well as the score obtained by the student at the end of the semester.

Figure 1 shows an E/R diagram with four entities as well as their associated attributes. The sample data is also shown in Figure 2 below. Your first task is to create the four tables, and populate them with some records as shown Figure 2.

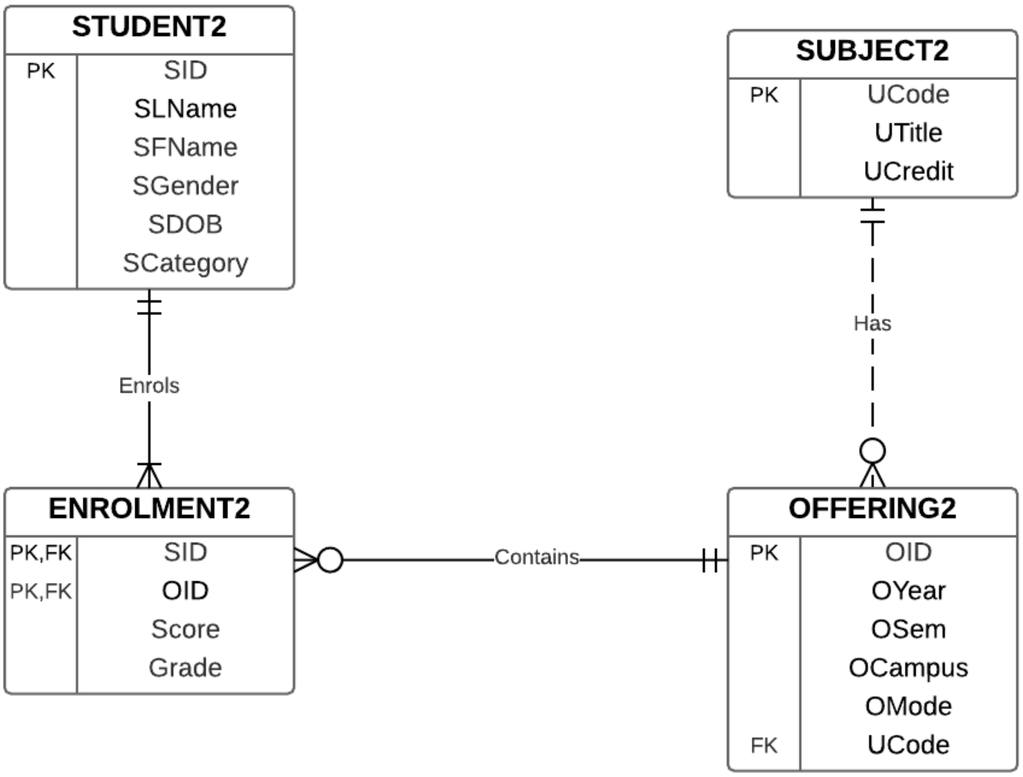


Figure 1: ER-Diagram for Student Case Study

Table Student2

SID	Slname	Sfname	Sgender	Sdob	Scategory
10001	Tan	Mirriam	F	19-Jul-81	112
10002	Murray	Juan	M	10-Jun-83	211
10003	Lay	Andy	M	19-Feb-86	211
10004	Wright	Allan	F	29-Jan-83	211
10005	Simon	Ally	F	24-Aug-83	112
10006	Smith	Ben	M	9-Jul-87	211
10007	Brown	Kate	F	19-Oct-72	112
10008	Miller	Larry	M	22-Jul-73	211
10009	Smith	Leonard	M	26-May-85	211
10010	Brown	Menson	M	12-Jul-83	112

Table Subject2

Ucode	Utitle	Ucredit
IT001	Database	5
IT002	Java	5
IT003	SAP	10
IT004	Network	5
IT005	ASP.net	5

Table Offering2

OID	Oyear	Osem	Ocampus	Omode	Ucode
1	2009	1	Main	D	IT001
2	2009	2	City	E	IT001
3	2009	2	DE	E	IT004
4	2009	2	Main	D	IT002
5	2009	1	City	E	IT003
6	2009	1	Main	E	IT002
7	2010	1	Main	D	IT001
8	2010	2	City	E	IT001
9	2010	2	DE	E	IT004
10	2010	2	Main	D	IT002
11	2010	1	City	E	IT003
12	2010	1	Main	E	IT002

Table Enrollment2

SID	OID	Score	Grade
10001	1	81	HD
10001	4	78	D
10002	2	64	C
10002	3	53	P
10003	2	32	N
10004	1	41	N
10005	5	63	C
10006	4	73	D
10006	1	74	D
10007	1	85	HD
10008	1	87	HD
10008	4	64	C
10009	1	75	D
10010	3	52	P
10005	6	65	C
10010	6	47	N

Figure 2: Tables and Records for the Student Enrollment Case Study

Tasks

a) Create table SUBJECT2 and insert the above 5 records.

```
Create Table SUBJECT2 (
 UCode   Varchar2(10) NOT NULL,
 UTitle  Varchar2(20) NOT NULL,
 UCredit Number(2),
 PRIMARY KEY (Ucode)
);

Insert Into SUBJECT2 Values ('IT001', 'Database', 5);
Insert Into SUBJECT2 Values ('IT002', 'Java', 5);
Insert Into SUBJECT2 Values ('IT003', 'SAP', 10);
Insert Into SUBJECT2 Values ('IT004', 'Network', 5);
Insert Into SUBJECT2 Values ('IT005', 'ASP.NET', 5);
```

b) Table STUDENT2 has been created in the `dtaniar` account. Several records have been inserted to this table. You can now import table STUDENT2 to your account using the following SQL statement:

```
Create Table STUDENT2
As
Select *
From dtaniar.STUDENT2;
```

c) Describe the structure of table STUDENT2.

```
desc student2;
```

OR

```
describe student2;
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------
 SID                                       NOT NULL VARCHAR2(10)
 SLNAME                                    NOT NULL VARCHAR2(20)
 SFNAME                                    NOT NULL VARCHAR2(20)
 SGENDER                                            CHAR(1)
 SDOB                                               DATE
 SCATEGORY                                          NUMBER(5)
```

d) Display all records from table STUDENT2.

```
select * from student2;

SID        SLNAME     SFNAME     SGENDER SDOB        SCATEGORY
---------- ---------- ---------- ------- ----------- ----------
10001      Tan        Mirriam    F       19/JUL/81   112
10002      Murray     Juan       M       10/JUN/83   211
10003      Lay        Andy       M       19/JUL/86   211
10004      Wright     Allan      F       29/JAN/83   211
10005      Simon      Ally       F       24/AUG/83   112
10006      Smith      Ben        M       09/JUL/87   211
10007      Brown      Kate       F       19/OCT/72   112

7 rows selected.
```

e) Insert the missing records to table STUDENT2.

```
Insert Into STUDENT2 Values ('10008', 'Miller', 'Larry', 'M',
To_date('22-07-1973', 'DD-MM-YYYY'), 211);
Insert Into STUDENT2 Values ('10009', 'Smith', 'Leonard', 'M',
To_date('26-05-1985', 'DD-MM-YYYY'), 211);
Insert Into STUDENT2 Values ('10010', 'Brown', 'Menson', 'M',
To_date('12-07-1983', 'DD-MM-YYYY'), 112);
```

f) Import Tables OFFERING2 and ENROLLMENT2 from `dtaniar` account. The method is similar to question (*b*) above.

```
CREATE TABLE Offering2
AS
SELECT *
FROM dtaniar.Offering2;

CREATE TABLE Enrollment2
AS
SELECT *
FROM dtaniar.Enrollment2;
```

g) Using SQL to answer the questions:

1) How many students enrolled in the Database unit offered in Main campus?

```
select count(*) as Number_of_student
from Offering2 o, Enrollment2 e, Subject2 s
where e.OID = o.OID
and s.Ucode = o.Ucode
and o.Ocampus = 'Main'
and s.Utitle = 'Database';
```

OR

```
select count(st.sid) as Number_of_student
from Student2 st, Offering2 o, Enrollment2 e, Subject2 s
where st.SID = e.SID
and e.OID = o.OID
and s.Ucode = o.Ucode
and o.Ocampus = 'Main'
and s.Utitle = 'Database';

ANSWER:

NUMBER_OF_STUDENT
-----------------
                6
```

2) What is the total score of students taking the Database unit in Main campus?

```
select SUM(e.score) as Total_score
from Offering2 o, Enrollment2 e, Subject2 s
where e.OID = o.OID
and s.Ucode = o.Ucode
and o.Ocampus = 'Main'
and s.Utitle = 'Database';

ANSWER:

TOTAL_SCORE
-----------
        443
```

3) How many students enrolled in the Java unit offered in Semester 2, 2009?

```
select count(*) as Number_of_student
from Offering2 o, Enrollment2 e, Subject2 s
where e.OID = o.OID
and s.Ucode = o.Ucode
and s.Utitle = 'Java'
and o.Osem = 2
and o.Oyear = 2009;
```

OR

```
select count(st.sid) as Number_of_student
from Student2 st, Offering2 o, Enrollment2 e, Subject2 s
where st.SID = e.SID
and e.OID = o.OID
and s.Ucode = o.Ucode
and s.Utitle = 'Java'
and o.Osem = 2
and o.Oyear = 2009;
```

```
ANSWER:

NUMBER_OF_STUDENT
-----------------
                3
```

4) What is the total score of students taking the Java unit in Semester 2, 2009?

```
select SUM(e.score) as Total_score
from Offering2 o, Enrollment2 e, Subject2 s
where e.OID = o.OID
and s.Ucode = o.Ucode
and s.Utitle = 'Java'
and o.Osem = 2
and o.Oyear = 2009;
```

```
ANSWER:

TOTAL_SCORE
-----------
        215
```

5) How many students received HD in the SAP unit offered in Semester 1, 2009?

```
select count(*) as Number_of_student
from Offering2 o, Enrollment2 e, Subject2 s
where e.OID = o.OID
and s.Ucode = o.Ucode
and e.Grade = 'HD'
and s.Utitle = 'SAP'
and o.Osem = 1
and o.Oyear = 2009;
```

OR

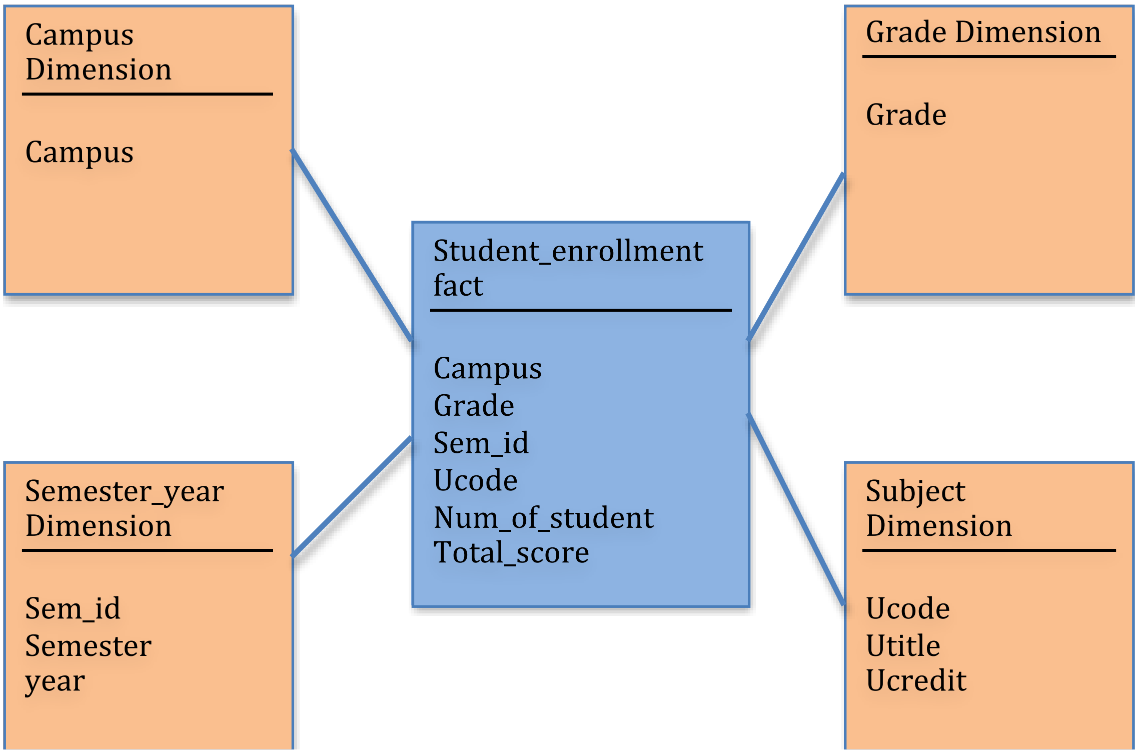
```
select count(st.sid) as Number_of_student
from Student2 st, Offering2 o, Enrollment2 e, Subject2 s
where st.SID = e.SID
and e.OID = o.OID
and s.Ucode = o.Ucode
and e.Grade = 'HD'
and s.Utitle = 'SAP'
and o.Osem = 1
and o.Oyear = 2009;


ANSWER:

NUMBER_OF_STUDENT
-----------------
                0
```

Implementing the star schema:

h) Draw a star schema based on the above case study? First identify the dimensions (and their attributes), and the fact measurements for the fact table?
Notes: Refer to Lecture Notes Week-1c-CaseStudy1-StudentEnrollment.ppt. The requirements for the star schema are also stated in question (g) above.



i) Use the SQL command to create and populate the dimension tables.

```
--Campus dimension

Create table campus_dim as
SELECT distinct Ocampus
FROM Offering2;
```

```
--Semester_year dimension

Create table sem_year_dim as
SELECT distinct Oyear||Osem as sem_id, Oyear, Osem
FROM Offering2;

--Subject Dimension

Create table subject_dim as
SELECT *
FROM subject2;

--Grade Dimension

Create table grade_dim as
SELECT distinct Grade
FROM Enrollment2;
```

j) Use the SQL command to create the fact table.

```
create table student_enrollment_fact as
SELECT o.Ocampus, o.Oyear||o.Osem as sem_id, s.Ucode, e.Grade,
count(st.sid) as num_of_student, sum(e.score) as Total_score
FROM subject2 s, enrollment2 e, offering2 o, student2 st
WHERE e.OID = o.OID
and s.Ucode = o.Ucode
and st.SID = e.SID
GROUP BY o.Ocampus, o.Oyear||o.Osem, s.Ucode, e.Grade;
```

Alternatively, we can avoid using table student2:

```
create table student_enrollment_fact as
SELECT o.Ocampus, o.Oyear||o.Osem as sem_id, s.Ucode, e.Grade,
  count(e.sid) as num_of_student, sum(e.score) as Total_score
FROM subject2 s, enrollment2 e, offering2 o
WHERE e.OID = o.OID
and s.Ucode = o.Ucode
GROUP BY o.Ocampus, o.Oyear||o.Osem, s.Ucode, e.Grade;
```

Further, we can also avoid using table subject2:

```
create table student_enrollment_fact as
SELECT o.Ocampus, o.Oyear||o.Osem as sem_id, o.Ucode, e.Grade,
  count(e.sid) as num_of_student, sum(e.score) as Total_score
FROM enrollment2 e, offering2 o
WHERE e.OID = o.OID
GROUP BY o.Ocampus, o.Oyear||o.Osem, o.Ucode, e.Grade;
```

k) Use the star schema that you have created, display the average score of each unit offered in 2009.

```
SELECT s.utitle, sum(f.Total_score)/sum(f.num_of_student) as Avg_score
from student_enrollment_fact f, subject_dim s, sem_year_dim y
WHERE f.ucode = s.ucode
AND f.sem_id = y.sem_id
AND y.oyear = 2009
Group by s.utitle;
```

```
UTITLE                AVG_SCORE
-------------------- ----------
SAP                          63
Java                       65.4
Network                    52.5
Database                 67.375
```

l) Use the star schema that you have created, display the average score of each unit offered in main campus.

```
SELECT s.utitle, sum(f.Total_score)/sum(f.num_of_student) as Avg_score
from student_enrollment_fact f, subject_dim s, campus_dim c
WHERE f.ucode = s.ucode
AND f.ocampus = c.ocampus
AND c.ocampus = 'Main'
Group by s.utitle;
```

```
UTITLE                AVG_SCORE
-------------------- ----------
Java                       65.4
Database             73.8333333
```

m) Use the star schema that you have created, display the average score of Database unit with the grade N.

```
SELECT s.utitle, sum(f.Total_score)/sum(f.num_of_student) as Avg_score
from student_enrollment_fact f, subject_dim s, grade_dim g
WHERE f.ucode = s.ucode
AND f.grade = g.grade
AND s.utitle = 'Database'
AND g.grade = 'N'
Group by s.utitle;
```

```
UTITLE                AVG_SCORE
-------------------- ----------
Database                   36.5
```

THE END