Office Use Only

# Monash University

## Semester Examination Period
## Faculty of Information Technology


**EXAM CODES:** Sample-3

**TITLE OF PAPER:** Sample-3

**EXAM DURATION:**

**READING TIME:**

***THIS PAPER IS FOR STUDENTS STUDYING AT:( tick where applicable)***

| | | | | |
|---|---|---|---|---|
| ☐ Berwick | ☒ Clayton | ☐ Malaysia | ☒ Off Campus Learning | ☐ Open Learning |
| ☒ Caulfield | ☐ Gippsland | ☐ Peninsula | ☐ Enhancement Studies | ☐ Sth Africa |
| ☐ Pharmacy | ☐ Other (specify) | | | |

**During an exam, you must not have in your possession, a book, notes, paper, calculator, pencil case, mobile phone or other material/item which has not been authorised for the exam or specifically permitted as noted below. Any material or item on your desk, chair or person will be deemed to be in your possession. You are reminded that possession of unauthorised materials in an exam is a discipline offence under Monash Statute 4.1.**

**No examination papers are to be removed from the room.**

Attempt all questions. All answers must be **printed neatly** on this paper. Answer questions with concisely expressed factual information. The backs of pages in this paper may be used for any rough work. **Any material written on the backs of pages will not normally be corrected**. If an answer needs to overflow from its designated answer space to a blank page, clearly indicate that this is the case and that the material on the blank page is for correction. This paper must be handed up at the end of the examination, even if no questions are attempted. **There are 8 questions, each of which is worth 10 marks. The total is 80 marks.** This exam counts as 60% of the final assessment for the unit.

**AUTHORISED MATERIALS**

| | | |
|---|---|---|
| **CALCULATORS** | ☐ **YES** | ☒ **NO** |
| **OPEN BOOK** | ☐ **YES** | ☒ **NO** |
| **SPECIFICALLY PERMITTED ITEMS** | ☐ **YES** | ☒ **NO** |

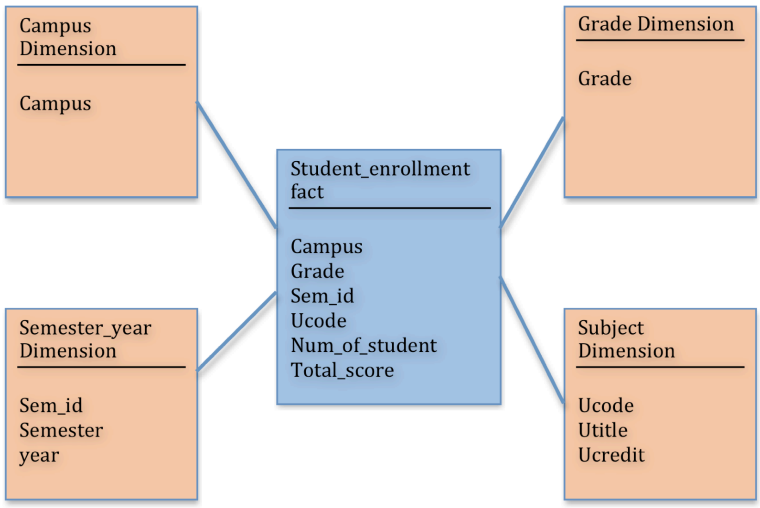
***Candidates must complete this section if required to write answers within this paper***

STUDENT ID __ __ __ __ __ __ __ __ __ DESK NUMBER __ __ __ __

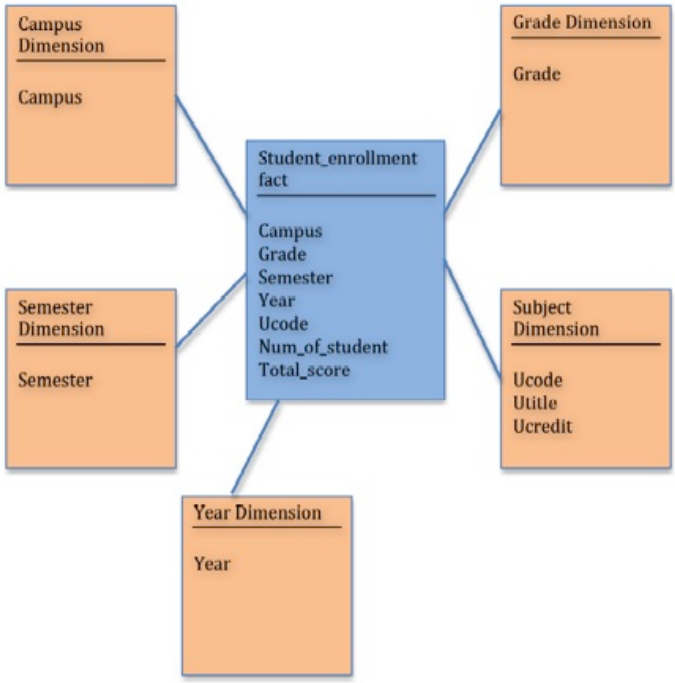
**Question 1:**

Consider the following two star schemas. Star schema-1 has a dimension called Semester_Year_Dimension, whereas in Star Schema-2, the Semester_Year_Dimension is split into two dimensions (e.g. Semester_Dimension and Year_Dimension)

*Star Schema-1 (one semester-year dimension)*




*Star Schema-2 (two separate dimensions for semester and year)*




*Questions*:

*(i)* Show the sample contents of tables Semester_Year_Dimension and Fact-1 (from Star Schema-1), and the sample contents of tables Semester_Dimension, Year_Dimension, and Fact-2 (from Star Schema-2)

*(ii)* Compare and contrast the two star schemas.

Write your answer here:

Star Schema-1:

```
SQL> select * from sem_year_dim;

SEM_ID     OYEAR      OSEM
---------- ---------- ----
20092      2009          2
20091      2009          1

SQL> select * from student_enrollment_fact;

OCAMPUS    SEM_ID     UCODE      GRADE NUM_OF_STUDENT TOTAL_SCORE
---------- ---------- ---------- ----- -------------- -----------
Main       20091      IT001      HD                3         253
Main       20091      IT001      D                 2         149
Main       20091      IT002      C                 1          65
Main       20091      IT002      N                 1          47
Main       20092      IT002      D                 2         151
City       20092      IT001      C                 1          64
Main       20091      IT001      N                 1          41
Main       20092      IT002      C                 1          64
DE         20092      IT004      P                 2         105
City       20091      IT003      C                 1          63
City       20092      IT001      N                 1          32

11 rows selected.
```

Star Schema-2:

```
SQL> select * from sem_dim;

    SEM_ID
----------
         1
         2

SQL> select * from year_dim;

   YEAR_ID
----------
      2009

SQL> select * from student_enrollment_fact_2;

OCAMPUS    SEM_ID  YEAR_ID UCODE      GR NUM_OF_STUDENT TOTAL_SCORE
---------- ------ -------- ---------- -- -------------- -----------
DE              2     2009 IT004      P                2         105
Main            1     2009 IT001      N                1          41
Main            1     2009 IT001      D                2         149
Main            2     2009 IT002      C                1          64
Main            1     2009 IT002      N                1          47
Main            2     2009 IT002      D                2         151
City            2     2009 IT001      N                1          32
Main            1     2009 IT002      C                1          65
City            1     2009 IT003      C                1          63
Main            1     2009 IT001      HD               3         253
City            2     2009 IT001      C                1          64

11 rows selected.
```

Continue your answer here:

**Star schema-1 (semester_year_dim):**
- Duplication. For example: S1 appears many times
- But if we need to access the dimension, we only need one join between the fact and semester_year_dim

**Star schema-2 (semester_dim and year_dim):**
- No duplication. For example, there is only one S1.
- If we need to access both semester and year dimensions, we need two join operations between the fact and the two dimension tables. Having more join operation is always costly.

**Conclusion**: From the FACT point of view, it does not really matter whether we have two separate dimensions or just one dimension. They are the same. However, if two information, like semester and year, is often seen as one entity or one piece of information, it would be easier if the two information is present in one dimension.

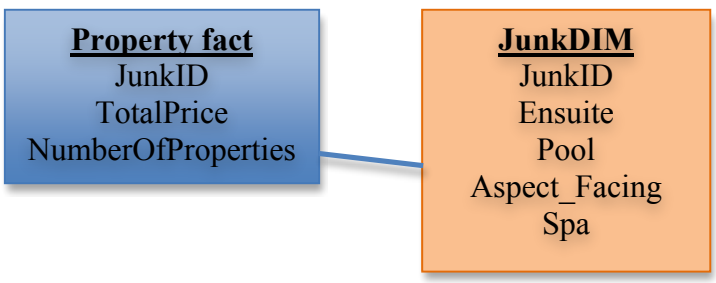
**Question 2:**

Using the real estate case study, assume we have a junk dimension as follows:

**Table JunkDIM:**

| JUNKID | ENSUITE | POOL | ASPECT_FACING | SPA |
|---|---|---|---|---|
| 1 | yes | no | East | yes |
| 2 | yes | no | South | yes |
| 3 | no | no | North | yes |
| 4 | no | no | East | no |
| 5 | yes | no | North | no |
| 6 | yes | yes | South | yes |
| 7 | no | no | West | no |
| 8 | yes | yes | East | yes |
| 9 | null | no | North | no |
| 10 | no | yes | East | no |
| 11 | yes | no | South | no |
| 12 | yes | no | North | yes |
| 13 | yes | no | East | no |
| 14 | no | no | North | no |
| 15 | no | yes | South | yes |
| 16 | yes | yes | North | no |
| 17 | no | no | East | yes |
| 18 | yes | yes | South | no |
| 19 | no | no | South | yes |
| 20 | no | yes | West | no |
| 21 | yes | yes | North | yes |
| 22 | no | yes | North | yes |
| 23 | null | no | East | no |
| 24 | no | no | South | no |
| 25 | nul | no | West | no |
| 26 | no | yes | West | yes |
| 27 | yes | no | West | no |
| 28 | no | no | West | yes |
| 29 | yes | yes | West | no |
| 30 | no | yes | North | no |
| 31 | no | yes | East | yes |
| 32 | nul | no | South | no |
| 33 | yes | yes | West | yes |
| 34 | yes | no | West | yes |
| 35 | yes | yes | East | no |
| 36 | no | yes | South | no |

The star schema using this junk dimension is as follows:



A tempfact table has been created as follows:

```
Create Table TempFactProperty As
Select Ensuite, Pool, Aspect_Facing, Spa,
       sum(Houseprice) as TotalPrice,
        count(*) as NumberofProperties
From dw.Property1
Group By Ensuite, Pool, Aspect_Facing, Spa;
```

A JunkId attribute is added to the TempFactProperty table using the following Later Table command:

```
Alter Table TempFactProperty
Add (JunkID Number(2));
```

The JunkID column in TempFactProperty is still empty. There are two ways to write an update SQL command to update the JunkID column in the TempFactProperty. Option-1 is to use a cursor in which the update junkid is placed in a loop. Option-2 is to use a subquery in the update TempFactProperty set JunkID.

*Question*: Write the SQL code to update JunkID column in TempFactProperty to match with the JunkID column in the JunkDim using the above two options.

Write your answer here:

OPTION-1:

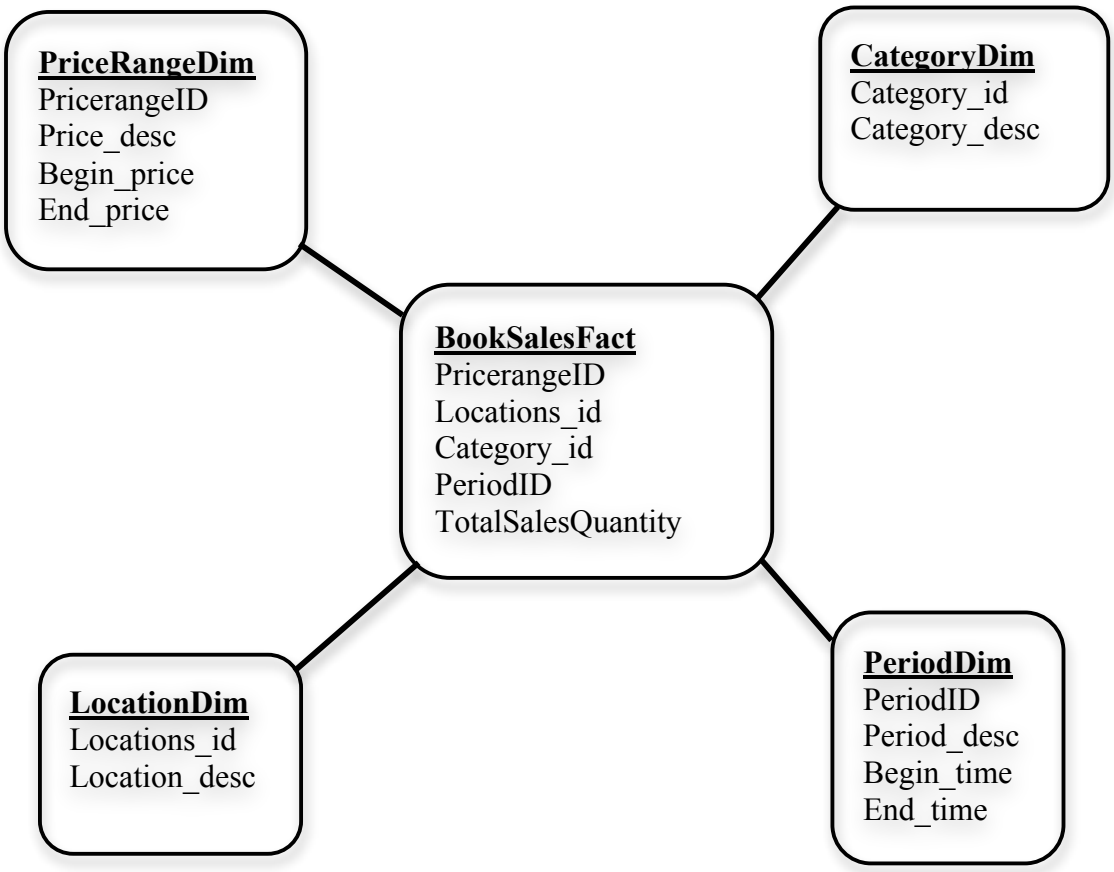
```
Declare
   cursor junkcursor is
        select * from JunkDim;
begin
    for junkcursorrec in junkcursor LOOP
       update TempFactProperty
       set JunkID = junkcursorrec.JunkID
       where Ensuite = junkcursorrec.Ensuite
       and Spa = junkcursorrec.Spa
       and Pool = junkcursorrec.Pool
       and Aspect_Facing = junkcursorrec.Aspect_Facing;
    end loop;
end;
/
```

OPTION-2:

```
update TempFactProperty tf
set tf.junkID = (select jd.junkID
                 from JunkDIM jd
                 where jd.ensuite = tf.ENSUITE
                 and jd.pool = tf.POOL
                 and jd.ASPECT_FACING = tf.ASPECT_FACING
                 and jd.spa = tf.spa );
```

**Question 3:**

Given the following star schema for a bookshop:




Write the SQL commands to answer the following queries:

*a.* Show total number (quantity) of books, including subtotals, by different transaction periods and customer locations
*b.* Show top 10% book categories from the sales
*c.* Show top 2 book price ranges which include the most number (quantity) of sold books

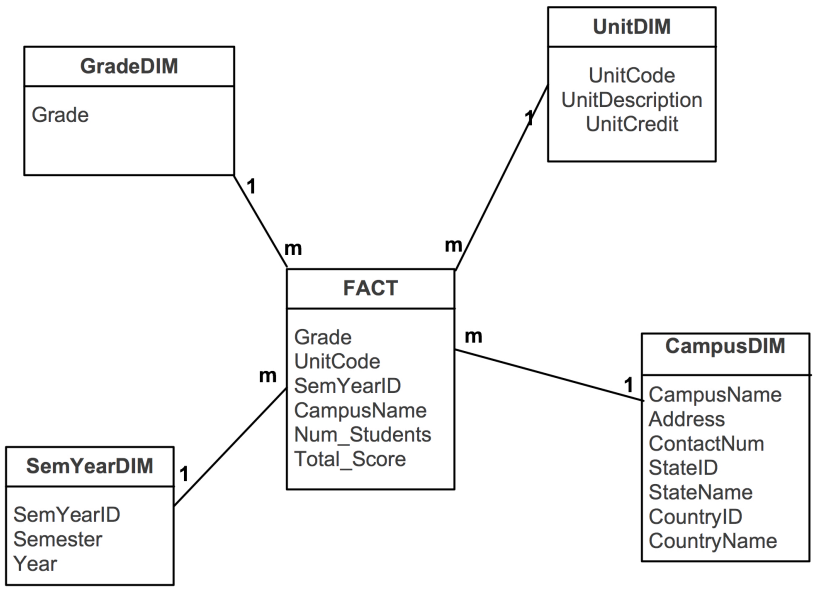
Write your answer here:

```
-- Qa (total number (quantity) by different periods and customer locations

SELECT
decode(grouping(l.location_desc),1,'All location',l.location_desc) as Location,
decode(grouping(p.period_desc),1,'All Quarter',p.period_desc),
SUM(f.TotalSalesQuantity) as total_number
FROM BookSalesFact f, perioddim p, locationdim l
WHERE f.locations_id = l.locations_id
AND f.periodID = p.periodID
GROUP BY rollup(l.location_desc, p.period_desc);

LOCATION                       DECODE(GROUPING(P.PERIOD_DESC) TOTAL_NUMBER
------------------------------ ------------------------------ ------------
Sydney                         1st quarter                               2
Sydney                         2nd quarter                               1
Sydney                         All Quarter                               3
Melbourne                      2nd quarter                               3
Melbourne                      3rd quarter                               2
Melbourne                      4th quarter                               5
Melbourne                      All Quarter                              10
All location                   All Quarter                              13

8 rows selected.


SQL> -- Qb (top10%)
SQL> SELECT *
  2  FROM (
  3  SELECT c.category_desc, SUM(p.TotalSalesQuantity) as total_sales,
  4  percent_rank() over (order by SUM(p.TotalSalesQuantity)) as percent
  5  FROM BookSalesFact p, categorydim c
  6  WHERE p.category_id = c.category_id
  7  GROUP BY c.category_desc)
  8  WHERE percent >= 0.9
  9  ORDER BY percent desc;

CATEGORY_DESC                  TOTAL_SALES    PERCENT
------------------------------ ----------- ----------
Science Fiction                          7          1



SQL> Qc (top2 rank)
SELECT *
  FROM (
  SELECT pricerangeID, SUM(p.TotalSalesQuantity),
  RANK() over (order by SUM(p.TotalSalesQuantity)desc) as ranking
  FROM BookSalesFact p
  GROUP BY pricerangeID
  ORDER BY ranking)
WHERE ranking <= 2;

PRICERANGEID SUM(P.TOTAL_NUMBER)    RANKING
------------ ------------------- ----------
           3                   7          1
           2                   3          2
           1                   3          2
```

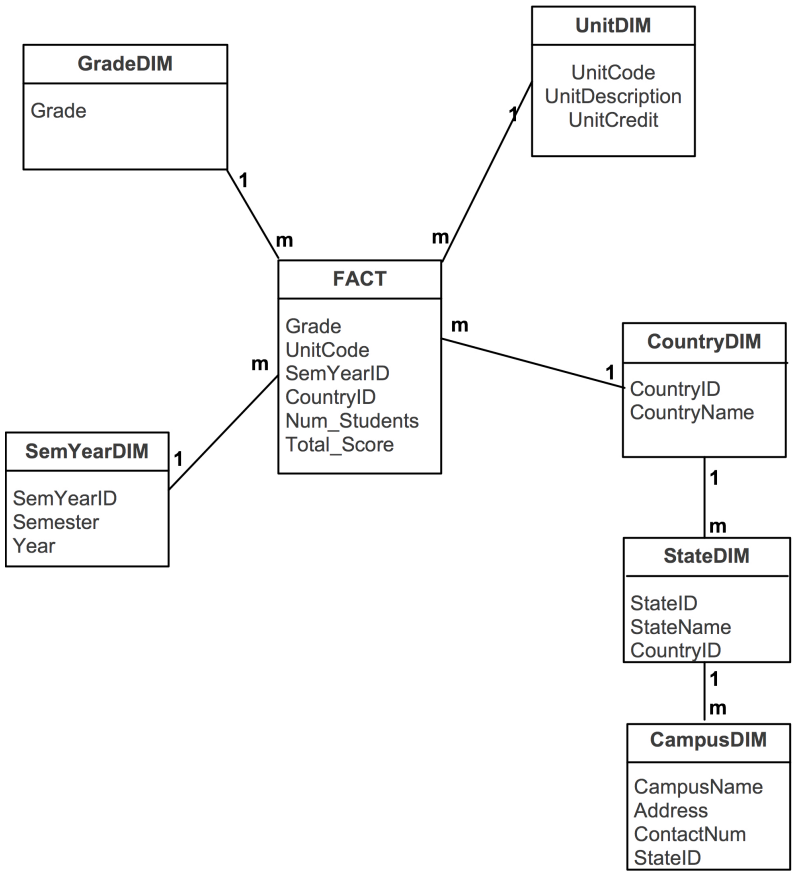
**Question 4:**

Consider the following Student Enrolment star schemas: Star Schema Version-1 does not have a dimension hierarchy, whereas Star Schema Version-2 has a dimension hierarchy: from country to state, and to campus.

*Star Schema Version-1*




*Star Schema Version-2*



*Questions:*

a. In contrasting both star schemas, is there any mistake in any of the two star schemas (Note that Star Schema Version-1 does not have a hierarchy, and Star Schema Version-2 does have)?
   - If yes, state which star schema, and explain your reason.
   - If no, also explain your reason.

b. Compare both star schemas.
   - If there are mistakes in any (or both) star schemas, you need to draw the correct schema(s) first before comparing between each other.
   - If there are no mistakes in both star schemas, you can immediately compare the two star schemas.

   Also, when you compare the two star schemas, you need to use some sample data (in the fact and in certain dimensions) to support your arguments

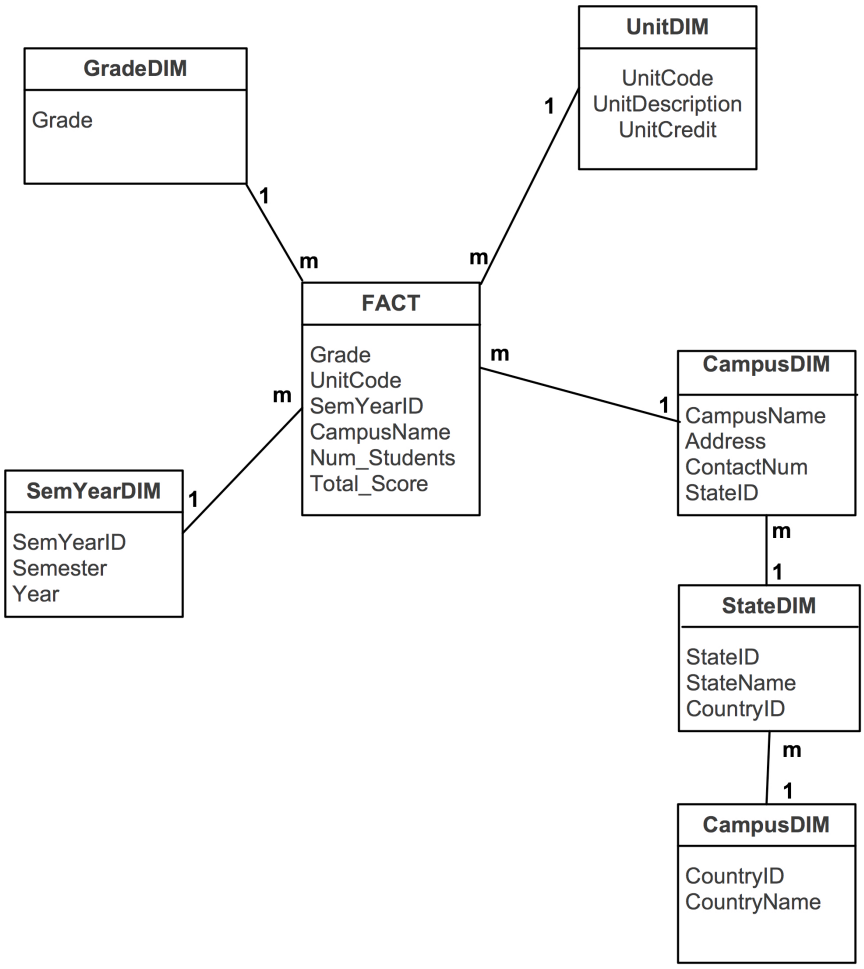
Write your answers here:

**(a)** There is a mistake in Star Schema Version-2; the mistake is in the hierarchy. The hierarchy should start from the most detail (e.g. Campus) to the most general (e.g. Country). Hence, the correct hierarchy should be CampusDIM➔StateDIM➔CountryDIM, and not in the opposite direction. Consequently, the fact should have CampusName, instead of CountryID.

There is no mistake in Star Schema Version-1.

Continue your answers here:

**(b)** The correct star schema for version-2 is as follows:
Note: the FK must also be correct.



Continue your answers here:

**Data duplication or Normalization**

Star Schema-1: unnormalized, has data duplication
The corrected (new) Star Schema-2: normalized, minimized data duplication

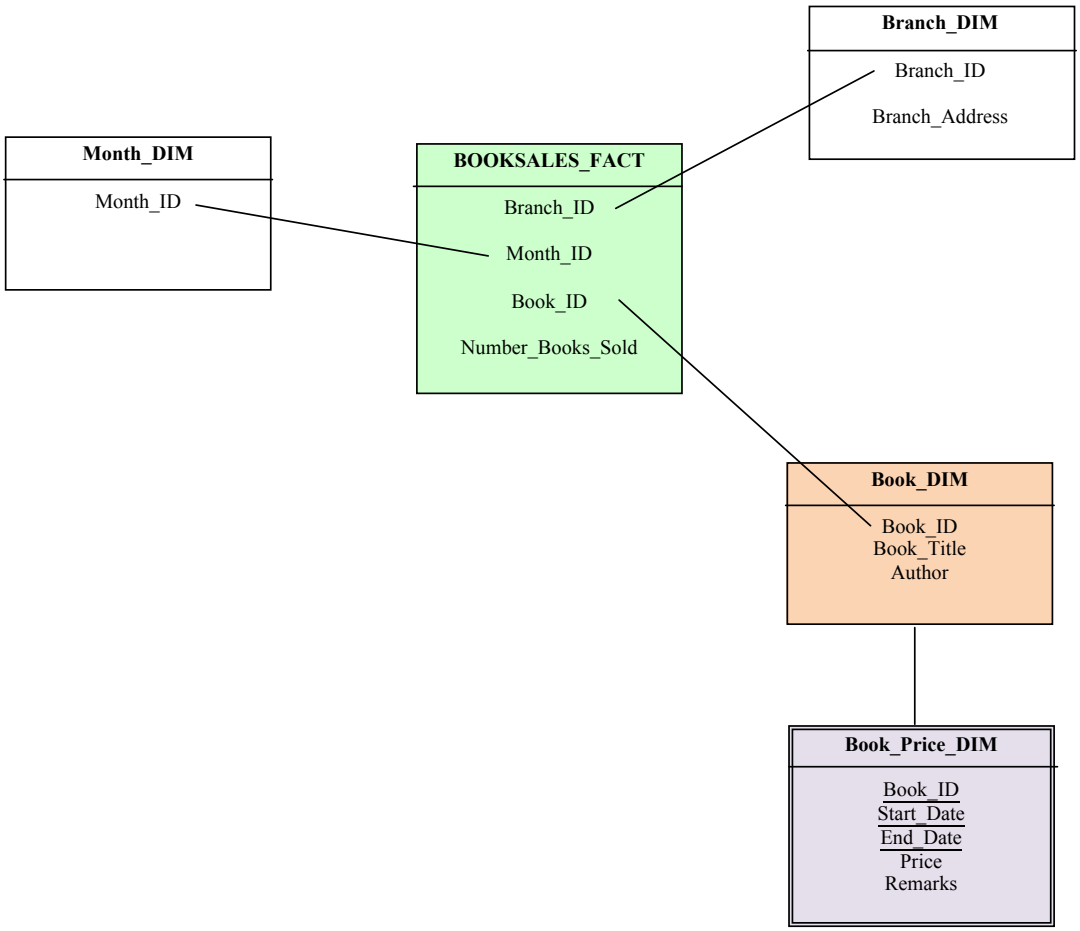
**Minimise Join**

Star Schema-1: need only one join between Fact and CampusDIM
The corrected (new) Star Schema-2: need three join operations between Fact, CampusDIM, StateDIM, and CountryDIM

For example: when we answer a query "how many students from campus in Australia", Star Schema-1 needs to join Fact with CampusDIM only, whereas Star Schema-2 needs to join tables Fact, CampusDIM, StateDIM, and CountryDIM.

**Question 5:**

This question is taken from the *Bookshop* Case Study on Temporal Data Warehousing. The following shows a star schema shows a fact table (number of books sold) and three dimensions (e.g. Month, Branch, and Book). The Book dimension is temporal dimension, which contains a temporal attribute, called Price, which is book price.




The tables for this star schema have been created and populated from the operational database. The sample data is as follows:

**Month_DIM Table**

| Month_ID |
|---|
| 201503 |
| 201502 |
| 201501 |
| 201412 |
| etc |

**Branch_DIM Table**

| Branch_ID | Branch_Address |
|---|---|
| City | Melbourne Central Shopping Centre, Melbourne |
| Chadstone | 285 Dandenong Road, Chadstone |
| Camberwell | 199 Burke Road, Camberwell |
| etc | |

**Book_DIM Table**

| Book_ID | Book_Title | Author |
|---|---|---|
| C1 | CSIRO Diet | CSIRO Team |
| H6 | Harry Potter 6 | Rowling |
| DV | Da Vinci Code | Dan Brown |
| … | … | … |

**Book_Price_DIM Table**

| Book_ID | Start_Date | End_Date | Price | Remarks |
|---|---|---|---|---|
| C1 | 201401 | 201407 | $45.95 | Full Price |
| C1 | 201408 | 201410 | $36.75 | 20% Discount |
| C1 | 201411 | 201501 | $23.00 | Half Price |
| C1 | 201502 | 201512 | $45.95 | Full Price |
| H6 | 201401 | 201403 | $21.95 | Launching |
| H6 | 201404 | 201501 | $30.95 | Full Price |
| H6 | 201502 | 201512 | $10.00 | End of Product Sale |
| DV | 201401 | 201512 | $27.95 | Full Price |
| … | … | … | … | |

**BookSales_Fact Table**

| Month_ID | Branch_ID | Book ID | Number_Books_Sold |
|---|---|---|---|
| 201503 | City | C1 | 5 |
| 201503 | City | H6 | 15 |
| 201503 | City | DV | 23 |
| 201503 | City | … | |
| 201503 | Chadstone | C1 | 15 |
| 201503 | Chadstone | H6 | 3 |
| 201503 | Chadstone | DV | 2 |
| 201503 | Chadstone | … | |
| 201503 | Camberwell | C1 | 1 |
| 201503 | Camberwell | H6 | 1 |
| 201503 | Camberwell | DV | 2 |
| 201503 | Camberwell | … | |
| 201503 | … | … | |
| … | … | … | |
| 201412 | City | C1 | 15 |
| 201412 | City | H6 | 6 |
| 201412 | City | DV | 6 |
| 201412 | City | … | |
| 201412 | Chadstone | C1 | 10 |
| 201412 | Chadstone | H6 | 8 |
| 201412 | Chadstone | DV | 1 |
| 201412 | Chadstone | … | |
| 201412 | Camberwell | C1 | 18 |
| 201412 | Camberwell | H6 | 3 |
| 201412 | Camberwell | DV | 2 |
| 201412 | Camberwell | … | |
| 201412 | … | … | |
| … | … | … | |

*Question:*
Write the SQL command to produce the following report (**10 marks**):

| Month_ID | Branch_ID | Book_ID | Book_Title | Author | Price | Number_Books_Sold |
|---|---|---|---|---|---|---|
| 201503 | City | C1 | CSIRO Diet | CSIRO Team | $45.95 | 5 |
| 201503 | City | H6 | Harry Potter 6 | Rowling | $10.00 | 15 |
| 201503 | City | DV | Da Vinci Code | Dan Brown | $27.95 | 23 |
| 201503 | City | | … | | | |
| 201503 | Chadstone | C1 | CSIRO Diet | CSIRO Team | $45.95 | 15 |
| 201503 | Chadstone | H6 | Harry Potter 6 | Rowling | $10.00 | 3 |
| 201503 | Chadstone | DV | Da Vinci Code | Dan Brown | $27.95 | 2 |
| 201503 | Chadstone | | … | | | |
| 201503 | Camberwell | C1 | CSIRO Diet | CSIRO Team | $45.95 | 1 |
| 201503 | Camberwell | H6 | Harry Potter 6 | Rowling | $10.00 | 1 |
| 201503 | Camberwell | DV | Da Vinci Code | Dan Brown | $27.95 | 2 |
| 201503 | Camberwell | | … | | | |
| 201503 | … | | … | | | |
| … | … | | … | | | |
| … | … | | … | | | |
| 201412 | City | C1 | CSIRO Diet | CSIRO Team | $23.00 | 15 |
| 201412 | City | H6 | Harry Potter 6 | Rowling | $30.95 | 6 |
| 201412 | City | DV | Da Vinci Code | Dan Brown | $27.95 | 6 |
| 201412 | City | | … | | | |
| 201412 | Chadstone | C1 | CSIRO Diet | CSIRO Team | $23.00 | 10 |
| 201412 | Chadstone | H6 | Harry Potter 6 | Rowling | $30.95 | 8 |
| 201412 | Chadstone | DV | Da Vinci Code | Dan Brown | $27.95 | 1 |
| 201412 | Chadstone | | … | | | |
| 201412 | Camberwell | C1 | CSIRO Diet | CSIRO Team | $23.00 | 18 |
| 201412 | Camberwell | H6 | Harry Potter 6 | Rowling | $30.95 | 3 |
| 201412 | Camberwell | DV | Da Vinci Code | Dan Brown | $27.95 | 2 |
| 201412 | Camberwell | | … | | | |
| 201412 | … | | … | | | |
| … | … | | … | | | |

The structures of the above tables are as follows:

```
SQL> desc Month_DIM;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 MONTH_ID                                           VARCHAR2(6)

SQL> desc Branch_DIM;

 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 BRANCH_ID                                          VARCHAR2(15)
 BRANCH_ADDRESS                                     VARCHAR2(50)
```

```
SQL> desc Book_DIM;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 BOOK_ID                                            VARCHAR2(5)
 BOOK_TITLE                                         VARCHAR2(20)
 AUTHOR                                             VARCHAR2(20)

SQL> desc Book_Price_DIM;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 BOOK_ID                                            VARCHAR2(5)
 START_DATE                                         VARCHAR2(6)
 END_DATE                                           VARCHAR2(6)
 PRICE                                              NUMBER(6,2)
 REMARKS                                            VARCHAR2(20)

SQL> desc BookSales_Fact;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 MONTH_ID                                           VARCHAR2(6)
 BRANCH_ID                                          VARCHAR2(15)
 BOOK_ID                                            VARCHAR2(5)
 NUMBER_BOOKS_SOLD                                  NUMBER
```

Write your answer here:

```
Select
   F.Month_ID,
   F.Branch_ID,
   F.Book_ID,
   B.Book_Title,
   B.Author,
   P.Price,
   F.Number_Books_Sold
From BookSales_Fact F, Book_DIM B, Book_Price_DIM P
Where F.Book_ID = B.Book_ID
And B.Book_ID = P.Book_ID
And F.Month_ID >= P.Start_Date
And F.Month_ID <= P.End_Date;
```

| MONTH_ID | BRANCH_ID | BOOK_ | AUTHOR | PRICE | NUMBER_BOOKS_SOLD |
|---|---|---|---|---|---|
| 201503 | City | C1 | CSIRO Team | 45.95 | 5 |
| 201503 | City | H6 | Rowling | 10 | 15 |
| 201503 | City | DV | Dan Brown | 27.95 | 23 |
| 201503 | Chadstone | C1 | CSIRO Team | 45.95 | 15 |
| 201503 | Chadstone | H6 | Rowling | 10 | 3 |
| 201503 | Chadstone | DV | Dan Brown | 27.95 | 2 |
| 201503 | Camberwell | C1 | CSIRO Team | 45.95 | 1 |
| 201503 | Camberwell | H6 | Rowling | 10 | 1 |
| 201503 | Camberwell | DV | Dan Brown | 27.95 | 2 |
| 201412 | City | C1 | CSIRO Team | 23 | 15 |
| 201412 | City | H6 | Rowling | 30.95 | 6 |

| MONTH_ID | BRANCH_ID | BOOK_ | AUTHOR | PRICE | NUMBER_BOOKS_SOLD |
|---|---|---|---|---|---|
| 201412 | City | DV | Dan Brown | 27.95 | 6 |
| 201412 | Chadstone | C1 | CSIRO Team | 23 | 10 |
| 201412 | Chadstone | H6 | Rowling | 30.95 | 8 |
| 201412 | Chadstone | DV | Dan Brown | 27.95 | 1 |
| 201412 | Camberwell | C1 | CSIRO Team | 23 | 18 |
| 201412 | Camberwell | H6 | Rowling | 30.95 | 3 |
| 201412 | Camberwell | DV | Dan Brown | 27.95 | 2 |

18 rows selected.

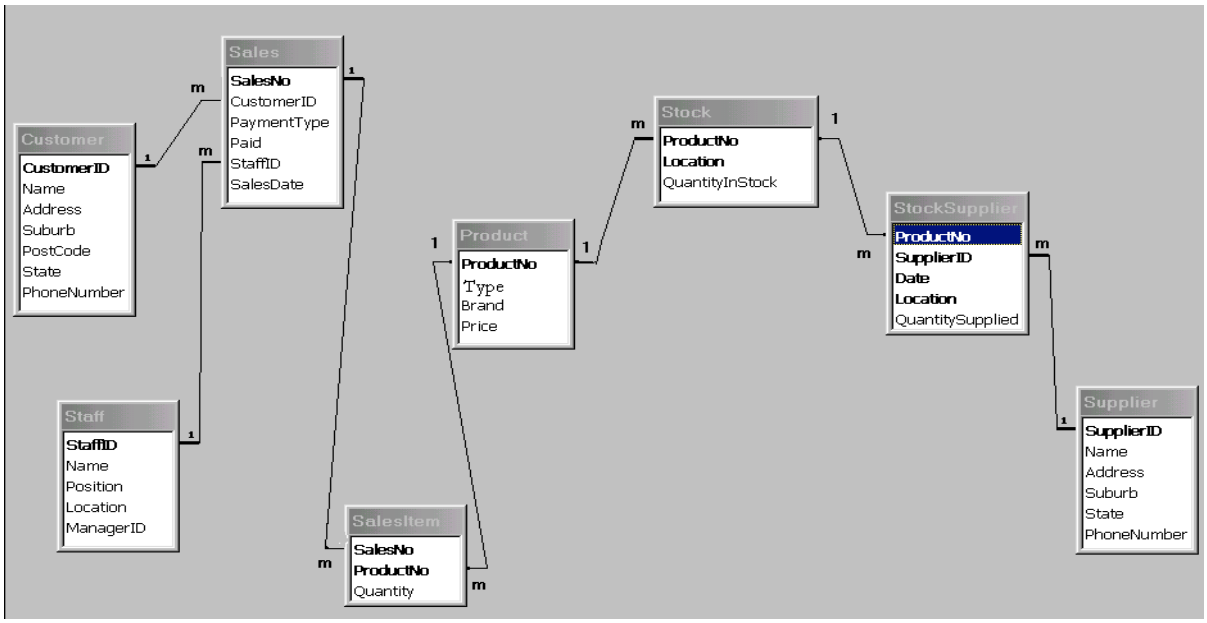
## Question 6:

This question is taken from the *Product-Sales-Supplier* Case Study.

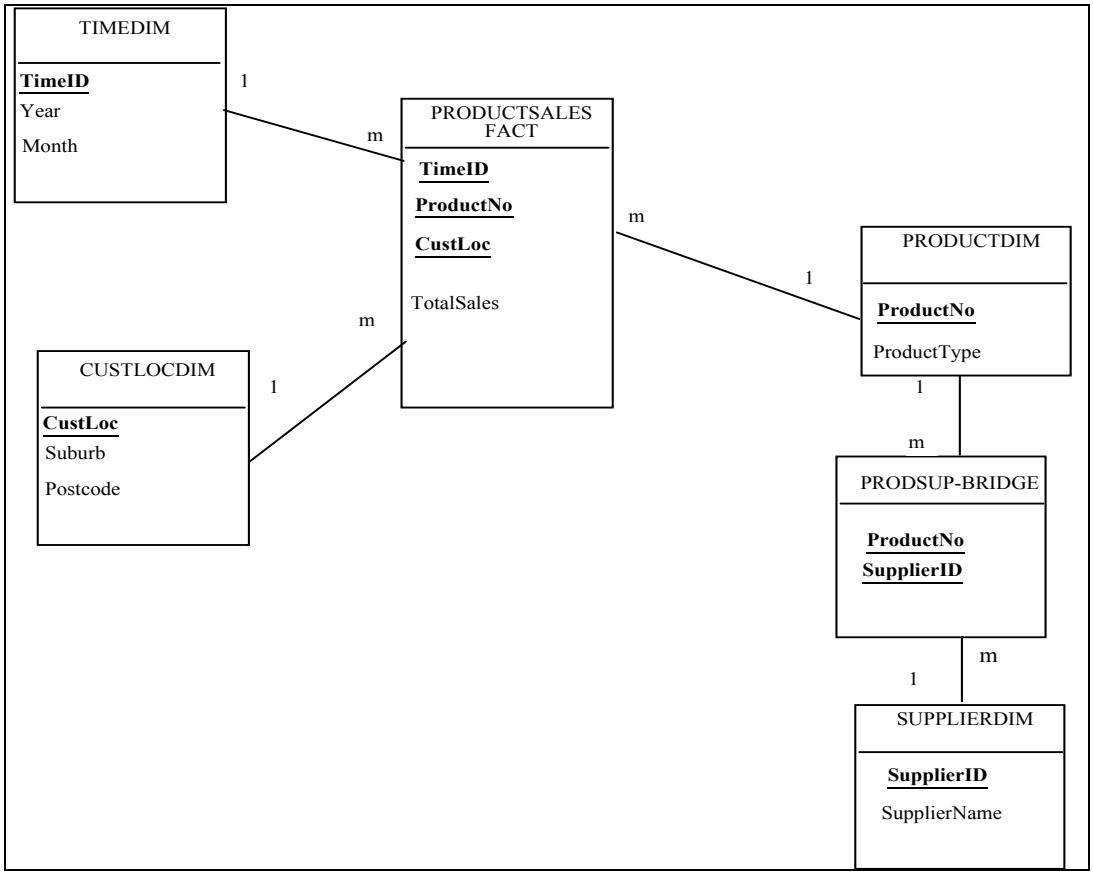
The director of a company is interested in analyzing the statistics of its product sales history. The analysis is needed for identifying which products are popular, which suppliers supply those products, when is the best time to purchase more stock, etc. You are required to design a small Data Warehouse to keep track of the statistics.

The director is particularly interested in analyzing the *total sales* (Quantity * Price) by *product*, *customer locations (suburbs and postcodes)*, *sales time periods* (monthly and yearly), and *supplier*.

The operational database currently has the following tables:




Your snowflake schema will have a Bridge Table connecting Product Dimension and Supplier Dimension. A snowflake schema with a Bridge Table as shown below:




The above snowflake schema is missing two attributes: **WeightFactor** attribute, and **ListAGG** attribute.

*Questions:*

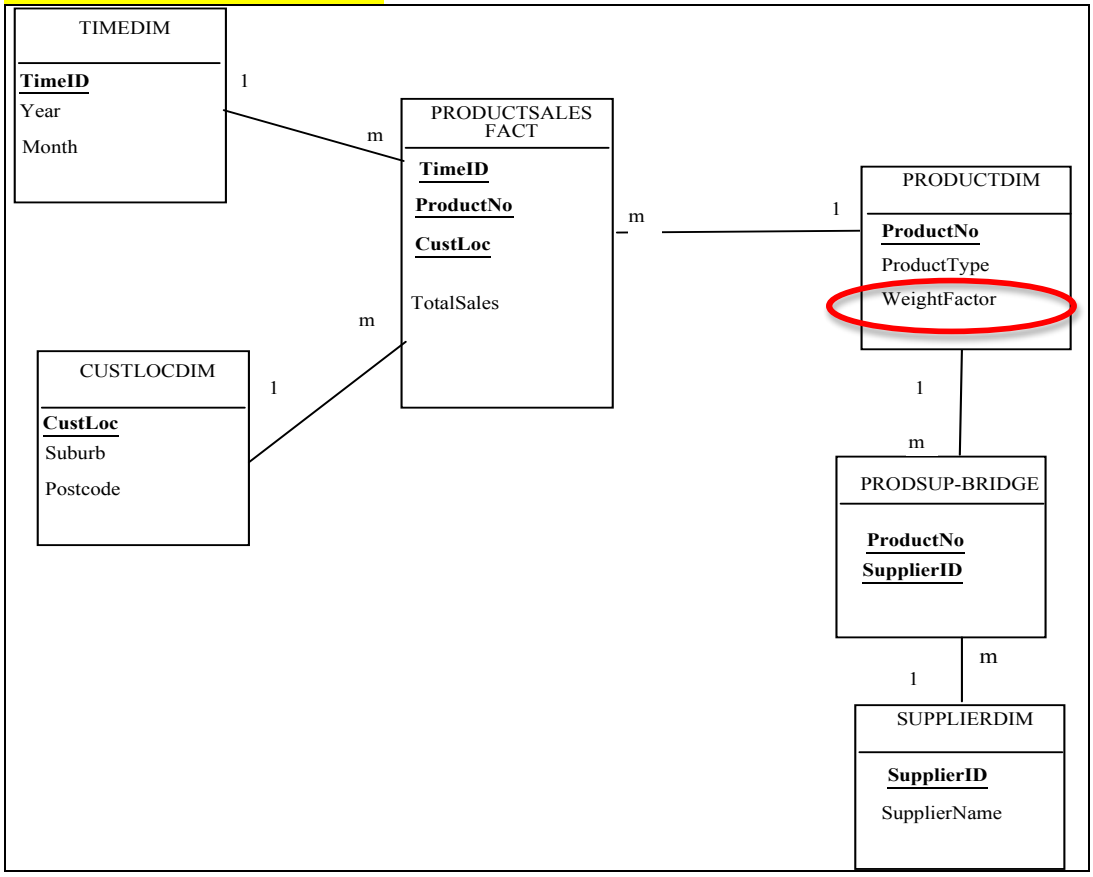
a. **Draw a new snowflake schema** (call it Snowflake Schema version 2) for the above case study, but this new snowflake schema must **use a WeightFactor attribute** (**without ListAGG attribute**). You also need to **show sample records** in the Product Dimension, the Bridge Table, and the Supplier Dimension. The sample data must show the correct values for the Weight attribute. Make sure that in your snowflake schema, the attributes are clearly shown.

b. **Draw another snowflake schema** (call it Star Schema version 3), which also has a Bridge Table and a WeightFactor attribute. But version-3 snowflake schema has the **ListAGG** attribute. You also need to **show sample records** in the Product Dimension, the Bridge Table, and the Supplier Dimension. The sample data must show the correct values for the Weight and ListAGG attributes.

c. Write the **SQL query** to create the ProductDim table for the Star Schema version 3.

Write your answer here:

Snowflake Schema Version 2




**ProductDIM Table**

| ProductNo | ProductType | WeightFactor |
|---|---|---|
| P1 | Shoes | 0.5 |
| P2 | Jeans | 0.33 |
| etc | | |

**ProdSup_Bridge Table**

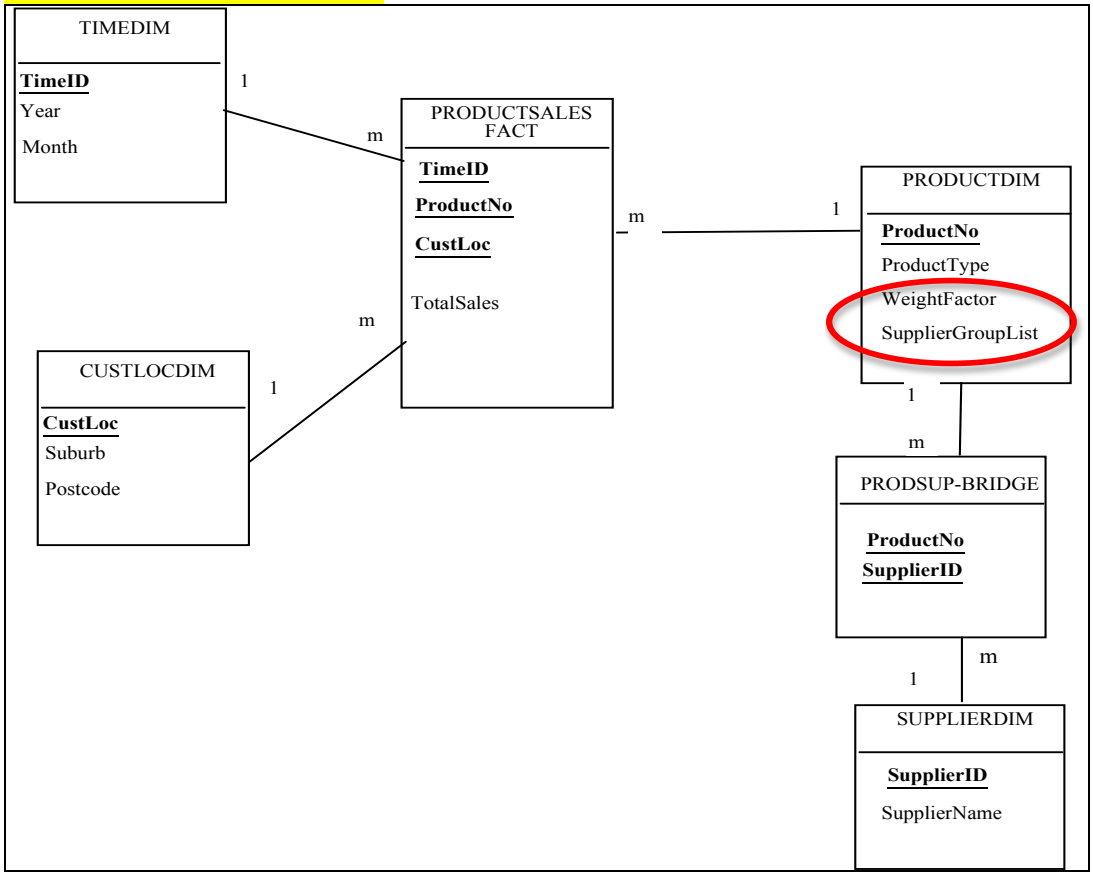
| ProductNo | SupplierID |
|---|---|
| P1 | S1 |
| P1 | S2 |
| P2 | S2 |
| P2 | S3 |
| P2 | S4 |
| etc | |

**SupplierDIM Table**

| SupplierID | SupplierName |
|---|---|
| S1 | Supplier-1 |
| S2 | Supplier-2 |
| S3 | Supplier-3 |
| S4 | Supplier-4 |
| etc | |

Continue your answer here:

Snowflake Schema Version 3




**ProductDIM Table**

| ProductNo | ProductType | WeightFactor | SupplierGroupList |
|---|---|---|---|
| P1 | Shoes | 0.5 | S1_S2 |
| P2 | Jeans | 0.33 | S2_S3_S4 |
| etc | | | |

**ProdSup_Bridge Table**

| ProductNo | SupplierID |
|---|---|
| P1 | S1 |
| P1 | S2 |
| P2 | S2 |
| P2 | S3 |
| P2 | S4 |
| etc | |

**SupplierDIM Table**

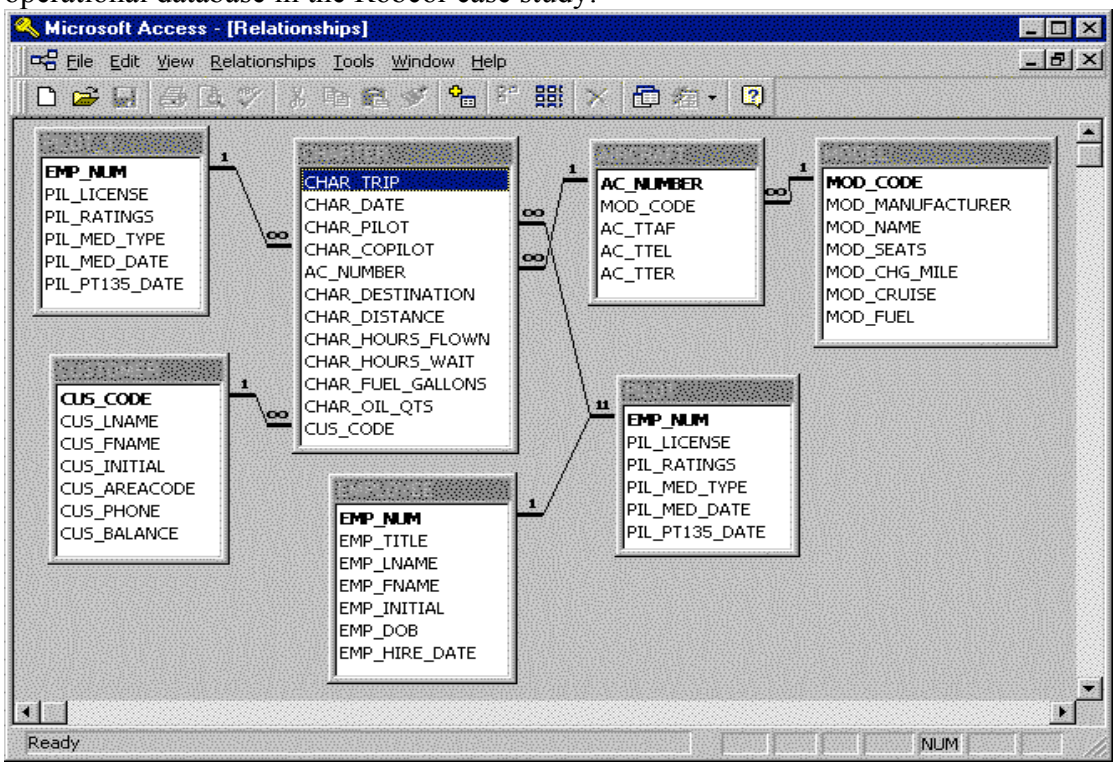
| SupplierID | SupplierName |
|---|---|
| S1 | Supplier-1 |
| S2 | Supplier-2 |
| S3 | Supplier-3 |
| S4 | Supplier-4 |
| etc | |

Continue your answer here:

```
Create Table ProductDim As
Select
   P.ProductNo,
   P.ProductType,
   1.0/count(SS.SupplierID) as WeightFactor,
   LISTAGG (SS.SupplierID, '_') Within Group
           (Order By SS.SupplierID) As SupplierGroupList
From Product P, Stock S, StockSupplier SS
Where P.ProductNo = S.ProductNo
And S.ProductNo = SS.ProductNo
And S.Location = SS.Location
Group By P.ProductNo, P.ProductType;
```

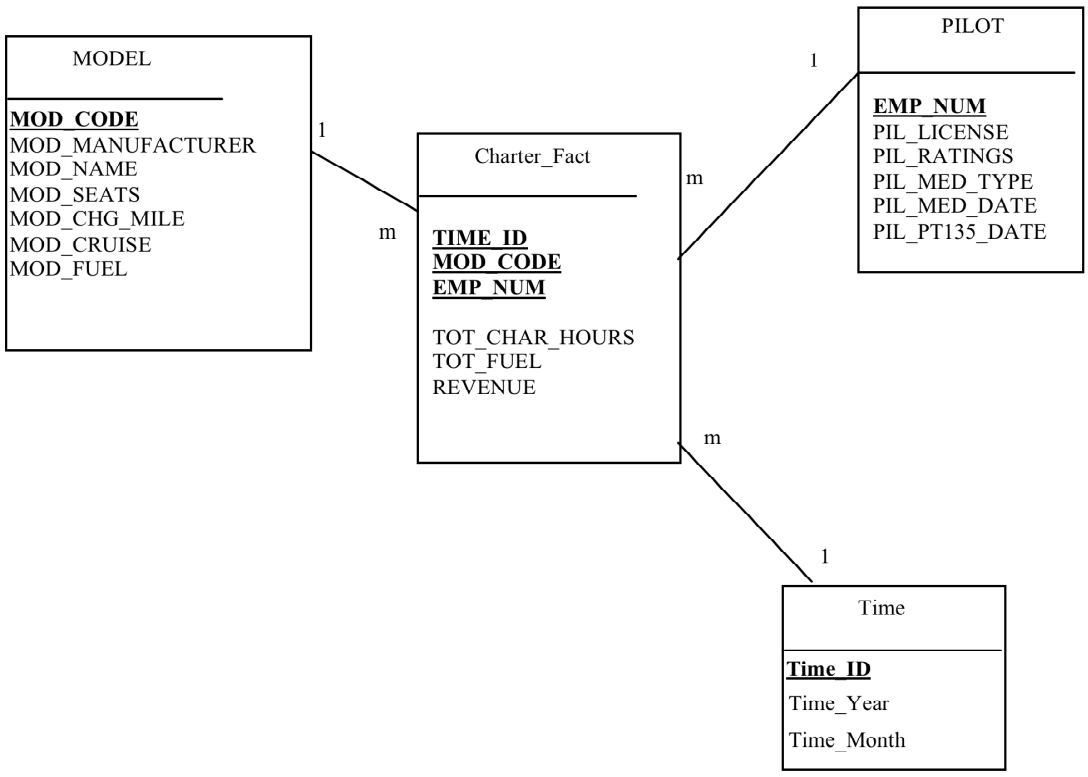
## Question 7:

This question is based on the Robcor case study. The following is the E/R diagram of the operational database in the Robcor case study:




A star schema for the above operational database is shown as follows:

*Robcor Star Schema-1*

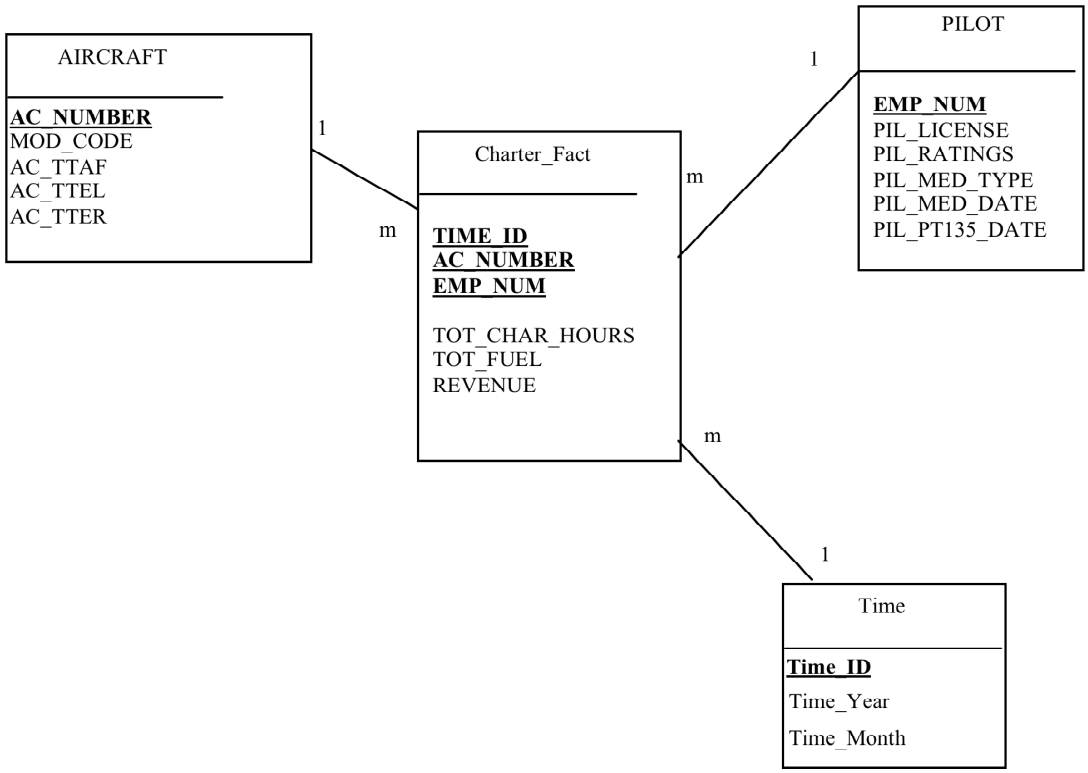


*Questions:*

a. Is it possible to determine which level Robcor Star Schema-1 is? If it is possible, state the level and also give the reason. If it is not possible to state the level, then give the reason.

b. Let's have a look at the following star schema (Robcor Star Schema-2). Between the two star schemas (Robcor Star Schema-1 and Robcor Star Schema-2), which one has a higher level of aggregation? State the name of the star schema, whether it is Robcor Star Schema-1 or Robcor Star Schema-2, and explain the reason.

*Robcor Star Schema-2*



Write your answers here:

(a)

It is not possible to determine whether this star schema is level-1 or level-2 or a higher level. What we know is that Robcor Star Schema-1 is not level-0 (not the lowest level).

Reason: Robcor Star Schema-1 is not the lowest level, because some of the dimensions have a higher level of aggregation (e.g. time id which is based on month, instead of the actual charter date).

However, it is not possible to name whether this is level 1 or level 2, because there can be any schemas in between Robcor star-schema-1 and level-0.

If there is **no star** schema in between level-0 and Robcor star schema-1, then Robcor star schema-1 becomes level 1.
If there is **one star schema** in between level-0 and Robcor star schema-1, then Robcor star schema-1 is level 2.
If there are **two star schemas** in between level-0 and Robcor star schema-1, then obviously Robcor star schema-1 becomes level 3.

(b)

Robcor star schema-1 has a higher level of aggregation than Robcor star schema-2.

Reason: one Model can have multiple Aircrafts. Hence a star schema using Model as a dimension has a higher level of aggregation than a star schema using Aircraft as a dimension

**Question 8:**

Suppose an operational database contains one table, called *Student*.

Table *Student*

| StudentID | StudentName | Suburb | Postcode | Sex |
|---|---|---|---|---|
| 21001 | Adam | Caulfield | 3162 | M |
| 21003 | Ben | Caulfield | 3162 | M |
| 21008 | Christine | Chadstone | 3148 | F |
| 21019 | Daisy | Caulfield | 3162 | F |
| 21033 | Edward | Clayton | 3168 | M |
| 21122 | Fred | Caulfield | 3162 | M |
| 21123 | Greg | Chadstone | 3148 | M |

Suppose the star schema contains only two dimensions: SuburbDIM and SexDIM. The fact table has one fact measure, called TotalStudents.

SuburbDIM table can be created using the following SQL command (assume that the operational database is located at `dtaniar` account):

```
CREATE TABLE SuburbDIM as
       SELECT distinct Suburb, Postcode
       FROM dtaniar.Student1;
```

Table *SuburbDIM*

| Suburb | Postcode |
|---|---|
| Caulfield | 3162 |
| Chadstone | 3148 |
| Clayton | 3168 |

SexDIM table can be created the same way:

```
CREATE TABLE SexDIM as
       SELECT distinct Sex
       FROM dtaniar.Student1;
```

Table *SexDIM*

| Sex |
|---|
| M |
| F |

And the Fact table can be created using the following SQL command:

```
CREATE TABLE Fact as
  SELECT Suburb, Sex, Count(*) as TotalStudents
  FROM dtaniar.Student1
  GROUP BY Suburb, Sex;
```

Table *Fact*

| Suburb | Sex | TotalStudents |
|---|---|---|
| Caulfield | M | 3 |
| Caulfield | F | 1 |
| Chadstone | M | 1 |
| Chadstone | F | 1 |
| Clayton | M | 1 |

Suppose we want to implement a surrogate key in the SuburbDIM, which we call SuburbID, for example.

Table *SuburbDIM*

| SuburbID | Suburb | Postcode |
|---|---|---|
| 1 | Caulfield | 3162 |
| 2 | Chadstone | 3148 |
| 3 | Clayton | 3168 |

*Questions*:
(a) Write the SQL query to create SuburbDIM to include a surrogate key called SuburbID as shown in the same data above.

(b) Supposed a tempfact table has been created using the following SQL query:

```
CREATE TABLE TempFact as
  SELECT Suburb, Sex FROM dtaniar.Student1;
```

Write an alter and an update command to alter the Tempfact to include a surrogate key (SuburbID).

(c) Write an SQL query to create a fact table so that the fact table will have the following contents:

Table *Fact*

| SuburbID | Sex | TotalStudents |
|---|---|---|
| 1 | M | 3 |
| 1 | F | 1 |
| 2 | M | 1 |
| 2 | F | 1 |
| 3 | M | 1 |

Write your answer here:

(a)
The SQL to create SuburbDIM is as follows:

```
CREATE TABLE SuburbDIM as
       SELECT distinct Suburb, Postcode
       FROM dtaniar.Student1;

Alter Table SuburbDIM add (SuburbID number(2));

Drop Sequence Suburb_seq_ID;

Create Sequence Suburb_seq_ID
  start with 1
  increment by 1
  maxvalue 99999999
  minvalue 1
  nocycle;

Update SuburbDIM SET SuburbID = suburb_seq_ID.nextval;
```

(b)

```
Alter Table TempFact
Add (SuburbID Number(2));


update TempFact tf
set tf.SuburbID = (select s.SuburbID
                   from SuburbDIM s
                   where s.Suburb = tf.Suburb);
```

(c)

```
Create Table Fact as
Select SuburbID, Sex, Count(*) as TotalStudents
From TempFact
GROUP BY SuburbID, Sex;
```

**THE END**