Office Use Only

MONASH University

Semester Two 2022
Examination Period

Faculty of Information Technology

Sample Exam

EXAM CODES: **FIT3003**

TITLE OF PAPER: Business Intelligence and Data Warehousing - SAMPLE 2

EXAM DURATION: 2 hours 10 minutes or 130 minutes

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

☐ Caulfield ☒ Clayton ☐ Parkville ☐ Peninsula
☐ Monash Extension ☐ Off Campus Learning ☒ Malaysia ☐ Sth Africa
☐ Other (specify)

During an exam, you must not have in your possession any item/material that has not been authorised for your exam. This includes books, notes, paper, electronic device/s, mobile phone, smart watch/device, calculator, pencil case, or writing on any part of your body. Any authorised items are listed below. Items/materials on your desk, chair, in your clothing or otherwise on your person will be deemed to be in your possession.

No examination materials are to be removed from the room. This includes retaining, copying, memorising or noting down content of exam material for personal use or to share with any other person by any means following your exam.
Failure to comply with the above instructions, or attempting to cheat or cheating in an exam is a discipline offence under Part 7 of the Monash University (Council) Regulations, or a breach of instructions under Part 3 of the Monash University (Academic Board) Regulations.

AUTHORISED MATERIALS

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

if yes, items permitted are:

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

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

Question 1:

The table below shows some sample records of a fact table named "Pet_Fact". The fact table records the sales of pet products and the total transactions for a retail chain that has stores across Victoria, Australia.

TimePeriod	City	Product	Price_Range	Total_Sales_KG	Total_Transactions
Q1 2020	Geelong	Dog Dry Food	High	200	50
Q1 2020	Melbourne	Dog Dry Food	Medium	300	100
Q1 2020	Melbourne	Cat Dry Food	High	250	40
Q1 2020	Ballarat	Cat Wet Food	Medium	100	50
Q1 2020	Ballarat	Cat Dry Food	Low	120	80
Q1 2020	Geelong	Dog Dry Food	Medium	300	100
Q1 2020	Melbourne	Dog Wet Food	Low	200	100
Q2 2020	Melbourne	Dog Dry Food	Medium	400	250
Q2 2020	Ballarat	Cat Dry Food	Medium	500	300
Q2 2020	Melbourne	Cat Wet Food	High	300	60
Q2 2020	Ballarat	Dog Dry Food	Low	350	180
Q2 2020	Melbourne	Dog Dry Food	High	300	50
Q2 2020	Geelong	Dog Wet Food	High	150	30
Q2 2020	Geelong	Dog Dry Food	High	350	50

a) Given the sample fact table above, write a SQL statement that can produce the following result:

TIME_PERIOD	CITY	PRODUCT	AVERAGE_TOTAL_SALES_KG
Q1 2020	Geelong	Dog Dry Food	250
Q1 2020	Geelong	(null)	250
Q1 2020	Melbourne	Dog Dry Food	300
Q1 2020	Melbourne	Dog Wet Food	200
Q1 2020	Melbourne	(null)	250
Q1 2020	(null)	(null)	250
Q2 2020	Ballarat	Dog Dry Food	350
Q2 2020	Ballarat	(null)	350
Q2 2020	Geelong	Dog Dry Food	350
Q2 2020	Geelong	Dog Wet Food	150
Q2 2020	Geelong	(null)	250
Q2 2020	Melbourne	Dog Dry Food	350
Q2 2020	Melbourne	(null)	350
Q2 2020	(null)	(null)	310
(null)	(null)	(null)	283.33

b) Write a SQL statement that generates the following table, which shows the total transactions of every product in each city. The table also contains the sum of total transactions by products and by cities.

PRODUCT	CITY	TOTAL_TRANSACTIONS
All Products	All Cities	1440
All Products	Ballarat	610
All Products	Geelong	230
All Products	Melbourne	600
Cat Dry Food	All Cities	420
Cat Dry Food	Ballarat	380
Cat Dry Food	Melbourne	40

Cat Wet Food	All Cities	110
Cat Wet Food	Ballarat	50
Cat Wet Food	Melbourne	60
Dog Dry Food	All Cities	780
Dog Dry Food	Ballarat	180
Dog Dry Food	Geelong	200
Dog Dry Food	Melbourne	400
Dog Wet Food	All Cities	130
Dog Wet Food	Geelong	30
Dog Wet Food	Melbourne	100

c) The management would like to see a report that shows the top two products by sales in KG, in each city regardless of time period. Write a SQL statement that returns the following table:

CITY	PRODUCT	TOTAL_SALES	RANK
Ballarat	Cat Dry Food	620	1
Ballarat	Dog Dry Food	350	2
Geelong	Dog Dry Food	850	1
Geelong	Dog Wet Food	150	2
Melbourne	Dog Dry Food	1000	1
Melbourne	Cat Wet Food	300	2

d) Write a SQL statement that returns the following table, which shows the cumulative total transactions:

TIME_PERIOD	CITY	TOTAL TRANSACTIONS	CUMULATIVE TRANSACTIONS
Q1 2020	Ballarat	130	130
Q1 2020	Geelong	150	280
Q1 2020	Melbourne	240	520
Q2 2020	Ballarat	480	1,000
Q2 2020	Geelong	80	1,080
Q2 2020	Melbourne	360	1,440

Write your answer here:

a)

```
SELECT Time_Period, City, Product,
     ROUND(AVG(Total_Sales_KG),2) AS "Average_Sales_Total_KG"
FROM PET_FACT
WHERE Product LIKE 'Dog%'
GROUP BY ROLLUP (TIME_PERIOD, CITY, product)
```

b)

```
SELECT
 DECODE(GROUPING(Product), 1, 'All Products', Product) AS Product,
 DECODE(GROUPING(City), 1, 'All Cities', City) AS City,
 SUM(Total_Transactions) AS "TOTAL_TRANSACTIONS"
FROM PET_FACT
GROUP BY CUBE(PRODUCT, CITY)
ORDER BY PRODUCT, CITY;
```

c)

```
SELECT *
FROM (
SELECT CITY, PRODUCT,
SUM(Total_Sales_KG) AS "TOTAL_SALES",
RANK() OVER (PARTITION BY CITY ORDER BY SUM(total_sales_kg) DESC) AS Rank
FROM PET_FACT
GROUP BY (CITY, PRODUCT)
ORDER BY City, SUM(Total_Sales_KG) Desc)
WHERE T.Rank <=2;
```

Continue your answer here:

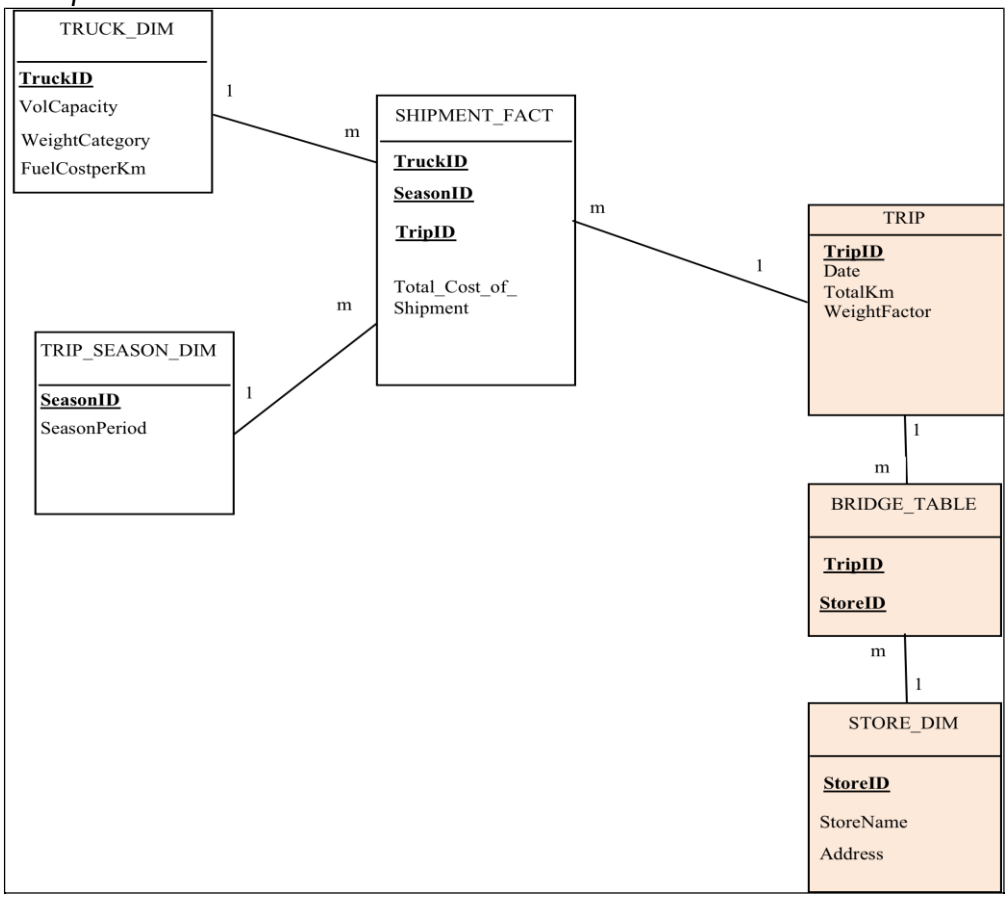
d)

```
Select time_period, city,
      SUM(total_transactions) As "Total Transactions",
   TO_CHAR(SUM(SUM(total_transactions))
   OVER(ORDER BY time_period ROWS UNBOUNDED PRECEDING),
   '9,999,999') AS "Cummulative Transactions"
From pet_fact
Group By time_period, city
Order by time_period, city;
```

Question 2:

A data warehouse for this Truck Delivery case study has been created, and the snowflake schema is shown as follows:

Snowflake schema



The dimension and fact tables for this snowflake schema have been created, and the contents of these tables are shown as follows:

```
SQL> select * from TruckDim1;

TRUCKID     VOLCAPACITY WEIGHTCATE   COSTPERKM
--------------- ---------------- --------------- ---------------
Truck1               250 Medium             1.2
Truck2               300 Medium             1.5
Truck3               100 Small               .8
Truck4               550 Large              2.3
Truck5               650 Large              2.5

SQL> select * from TripSeasonDim1;

SEASONID   SEASONPERIOD
---------- --------------------

Summer     Dec-Feb
Autumn     Mar-May
Winter     Jun-Aug
Spring     Sep-Nov
```

```
SQL> select * from TripDim1;

TRIPID      TRIPDATE      TOTALKM  WEIGHTFACTOR
---------- ------------ ---------- ------------
Trip2       14/APR/13         570    .333333333
Trip1       14/APR/13         370            .2

SQL> select * from Bridge_Table_Dim1;

TRIPID     STOREID
---------- ----------
Trip1      M1
Trip1      M2
Trip1      M3
Trip1      M4
Trip1      M8
Trip2      M1
Trip2      M2
Trip2      M4

SQL> select * from StoreDim1;

STOREID    STORENAME                 STOREADDRESS
---------- ------------------------- ------------------------------
M1         Myer City                 Melbourne
M2         Myer Chaddy               Chadstone
M3         Myer HiPoint              High Point
M4         Myer West                 Doncaster
M5         Myer North                Northland
M6         Myer South                Southland
M7         Myer East                 Eastland
M8         Myer Knox                 Knox

SQL> select * from TruckFact1;

TRUCKID    SEASONID   TRIPID      TOTALSHIPMENTCOST
---------- ---------- ---------- -------------------------
Truck1     Autumn     Trip1                      444
Truck3     Autumn     Trip3                      200
Truck2     Winter     Trip5                    262.5
Truck2     Autumn     Trip2                      855
Truck1     Winter     Trip4                      540
```

The structures of these tables are as follows:

```
SQL> desc TruckDim1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 TRUCKID                                   NOT NULL VARCHAR2(10)
 VOLCAPACITY                                        NUMBER(5,2)
 WEIGHTCATEGORY                                     VARCHAR2(10)
 COSTPERKM                                          NUMBER(5,2)

SQL> desc TripSeasonDim1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 SEASONID                                           VARCHAR2(10)
 SEASONPERIOD                                       VARCHAR2(20)
```

```
SQL> desc TripDim1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 TRIPID                                    NOT NULL VARCHAR2(10)
 TRIPDATE                                           DATE
 TOTALKM                                            NUMBER(5)
 WEIGHTFACTOR                                       NUMBER

SQL> desc Bridge_Table_Dim1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 TRIPID                                    NOT NULL VARCHAR2(10)
 STOREID                                   NOT NULL VARCHAR2(10)

SQL> desc StoreDim1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 STOREID                                   NOT NULL VARCHAR2(10)
 STORENAME                                          VARCHAR2(20)
 STOREADDRESS                                       VARCHAR2(20)

SQL> desc TruckFact1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 TRUCKID                                   NOT NULL VARCHAR2(10)
 SEASONID                                           VARCHAR2(10)
 TRIPID                                    NOT NULL VARCHAR2(10)
 TOTALSHIPMENTCOST                                  NUMBER
```

Questions:

(a) Write the SQL statement to display the shipment cost for each trip. Write down the output of this query as well.

(b) Write the SQL statement to display the shipment cost for each store. Write down the output of this query as well.

Write your answers here:

```
Select TF.TripID, Sum(TF.TotalShipmentCost) as TotalShipmentCost
From TruckFact1 TF, TripDim1 T
Where TF.TripID = T.TripID
Group By TF.TripID;
```

```
TRIPID     TOTALSHIPMENTCOST
---------- -----------------
Trip1                    444
Trip2                    855
```

Notes:
- data is incomplete

```
Select S.StoreID, S.StoreName, Sum(T.WeightFactor * TF.TotalShipmentCost)
as StoreDeliveryCost
From TruckFact1 TF, TripDim1 T, Bridge_Table_Dim1 B, StoreDim1 S
Where TF.TripID = T.TripID
And T.TripID = B.TripID
And B.StoreID = S.StoreID
Group By S.StoreID, S.StoreName;
```

```
STOREID    STORENAME                  STOREDELIVERYCOST
---------- -------------------------- -----------------
M3         Myer HiPoint                            88.8
M4         Myer West                              373.8
M8         Myer Knox                               88.8
M2         Myer Chaddy                            373.8
M1         Myer City                              373.8
```

Question 3:

Let's consider the Clothing case study. The operational database consists of the following tables:

CUSTOMER1

CUSTID	LNAME	FNAME	ADDRESS	PHONE	CITY
107	Smith	John	731 Plenty Road	9231455	Clayton
232	Wong	Franklin	638 Voss Street	9756945	Preston
133	Zelaya	Alicia	3321 Castle Ave	9867055	Balwyn
154	Wallace	Jennifer	291 Berry Street	9234536	Preston
179	Narayan	Ramesh	975 Fire Road	9456738	Carlton
181	Jane	Adam	229 Clayton Road	9543877	Clayton
183	Judy	Backhouse	122 Rose Street	9235345	Caulfield

ORDER1

ORDERID	ORDERDATE	PAYMETHOD	ORDERSOURCE	CUSTID
1057	20/02/06	CARD	WEB SITE	107
1058	03/03/06	CARD	PHONE	232
1059	12/03/06	CHEQUE	WEB SITE	133
1060	20/03/06	CHEQUE	WEB SITE	133
1061	10/04/06	CARD	FAX	179
1062	01/04/06	CARD	FAX	179
1063	07/09/06	CARD	WEB SITE	154
1064	14/07/06	CARD	WEB SITE	154
1065	30/11/06	CARD	PHONE	179
1066	20/01/06	CHEQUE	WEB SITE	179

ORDER_INV1

ORDERID	INVID	ORDERPRICE	QUANTITY
1057	11668	259.99	10
1058	11668	239.99	20
1059	11780	21.99	5
1060	11776	20.99	50
1061	11779	29.95	25
1061	11780	29.95	50
1062	11669	229.99	40
1063	11778	25.95	50
1064	11779	29.95	12
1065	11780	26.95	32
1066	11775	29.95	30

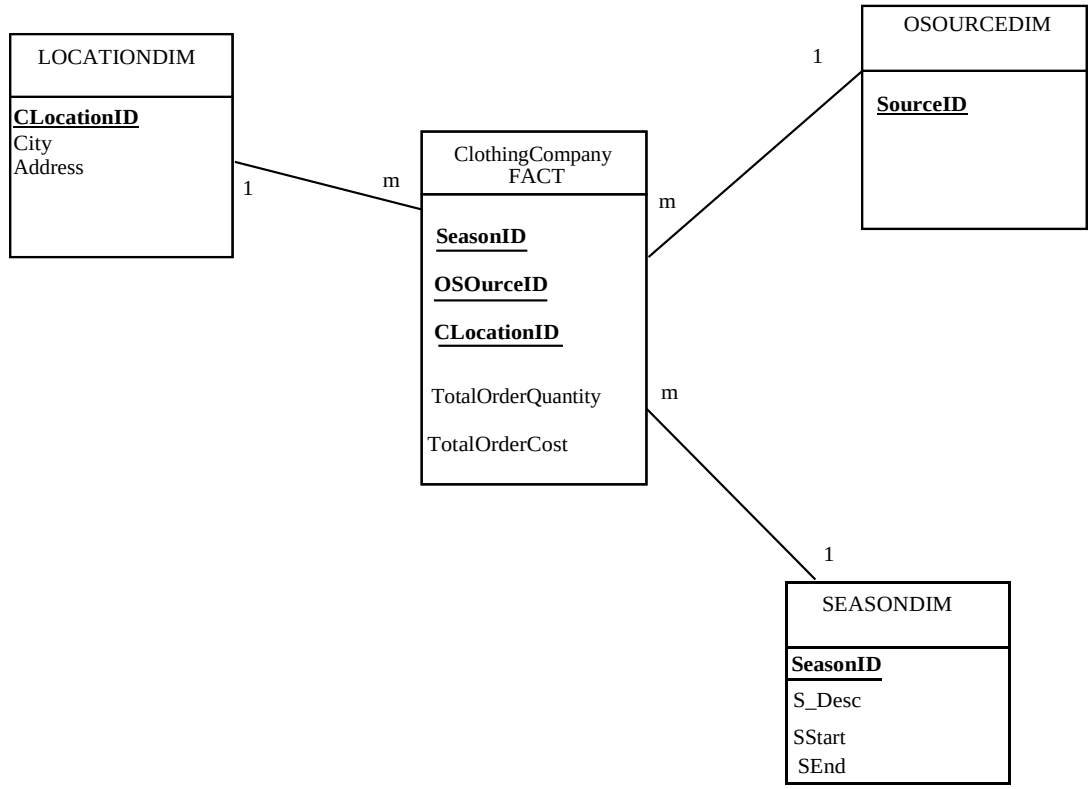
INVENTORY1

INVID	QOH	ITEMID	ITEMSIZE	COLOUR
11668	16	786	M	Sienna
11669	12	786	L	Forest
11775	150	894	S	Khaki
11776	147	894	M	Khaki
11777	0	894	L	Khaki
11778	139	894	S	Olive
11779	137	894	M	Olive
11780	115	894	L	Olive

QOH = Quantity on Hand

ITEM1			
ITEMID	**CURRENT PRICE**	**ITEMDESC**	**CATEGORY**
894	29.95	Women's Hiking Shorts	Women's Clothing
897	200.95	Women's Fleece Pullovers	Women's Clothing
995	50.00	Children's Beachcomber Sandals	Children's Clothing
559	35.00	Men's Expedition Parka	Men's Clothing
786	259.99	3-Season Jacket	Men's Clothing

A star schema has been created:



The fact measures included in the above star schema are TotalOrderQuantity, which is taken from the Quantity attribute from table Inventory, and TotalOrderCost, which is the sum of order price x quantity.

The above star schema contains highly aggregated data, and therefore this star schema is at level-2 in the data warehouse architecture.

Questions: Draw level-1 and level-0 star schemas for the above clothing data warehouse case study.

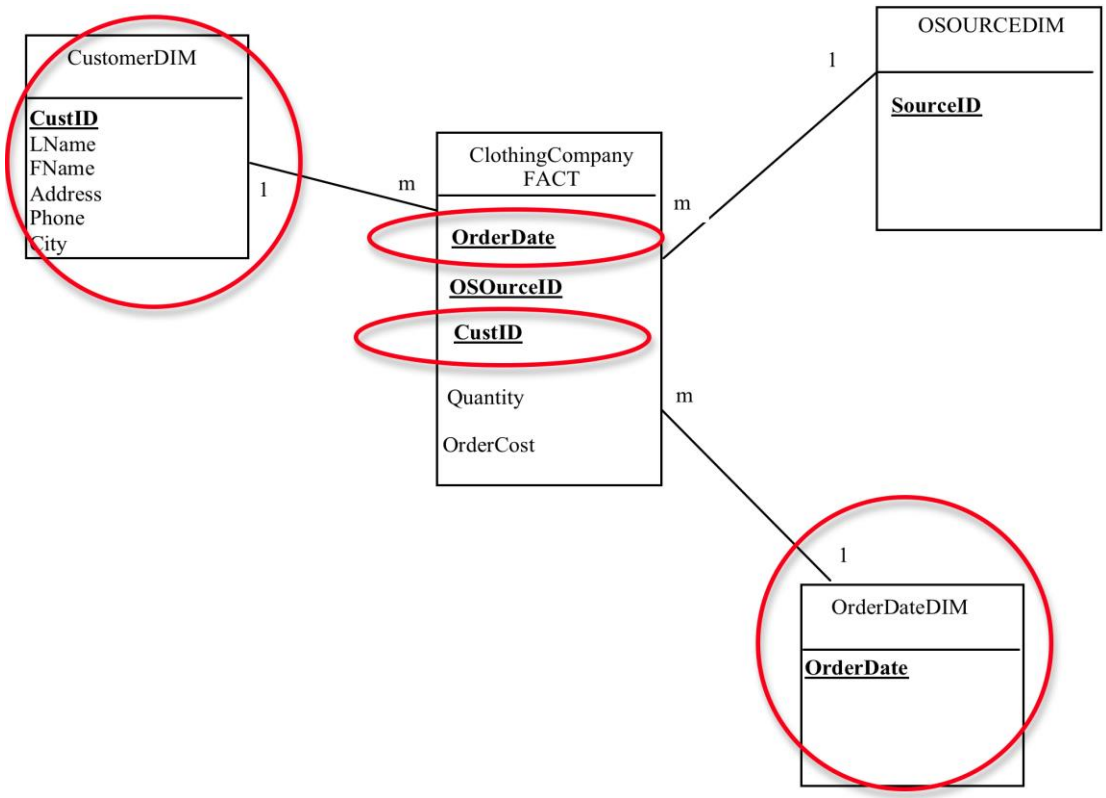
Write your answer here:

There are two aggregated dimensions: SeasonDIM and LocationDIM.

In Level-1, one of these dimensions is replaced by a lower level of aggregation. For example, SeasonDIM is changed to OrderDateDIM, or LocationDIM is changed to CustomerDIM.
In Level-1, only one of these dimensions needs to be lowered down.

In Level-0, both dimensions need to be lowered down.

Level-0



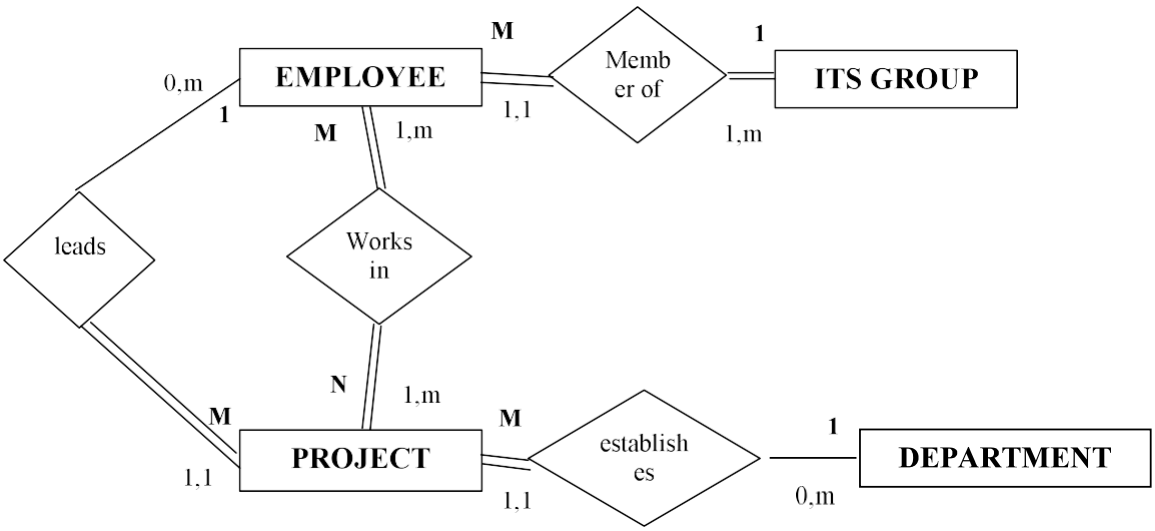
Question 4:

The Information Technology Services (ITS) department at Monash University needs a database to keep track of ITS staff members and the projects they are working on. The system must satisfy the following requirements:

- For each ITS employee, list the employee number, employee name, job title (position), and the number and name of the ITS group in which he or she works. In addition, for each project to which the employee is assigned, list the project number and name, the percent of the employee's time assignment to the project, and the total number of hours the employee has worked on the project so far.

- For each ITS project, list the project number and name (description), the name of the department that requested the project, the name of the contact person in that requesting department, the project type (maintenance, database, etc.), project status, start data, end date, total budgeted person-hours, total budgeted dollars, and the name of the ITS employee serving as project leader.

The E/R diagram for the above process is given as follows:



The system is now operating with the following tables:

EMPLOYEE (**emp#,** empname, position, ITSgroup#)
ITS-GROUP (**itsgroup#,** itsgroupname)
DEPARTMENT (**dept#,** deptname)
PROJECT (**proj#,** projname, contactperson, projtype, projstatus, start_date, end_date, budgperhours, budgtotals$, emp#, dept#)
EMP-PROJ (**emp#, proj#,** %timeassignment, projemptotalhrs)

Monash University management would like to analyse the performance of the ITS department, and in order to assist this process, you are asked to develop a data warehouse for analysis purposes. The analysis is needed for identifying at least the following questions:

- What is the total number of projects that are long duration?
- What is the total budget for a certain project type?
- What is an average budget cost per hour of a certain department?

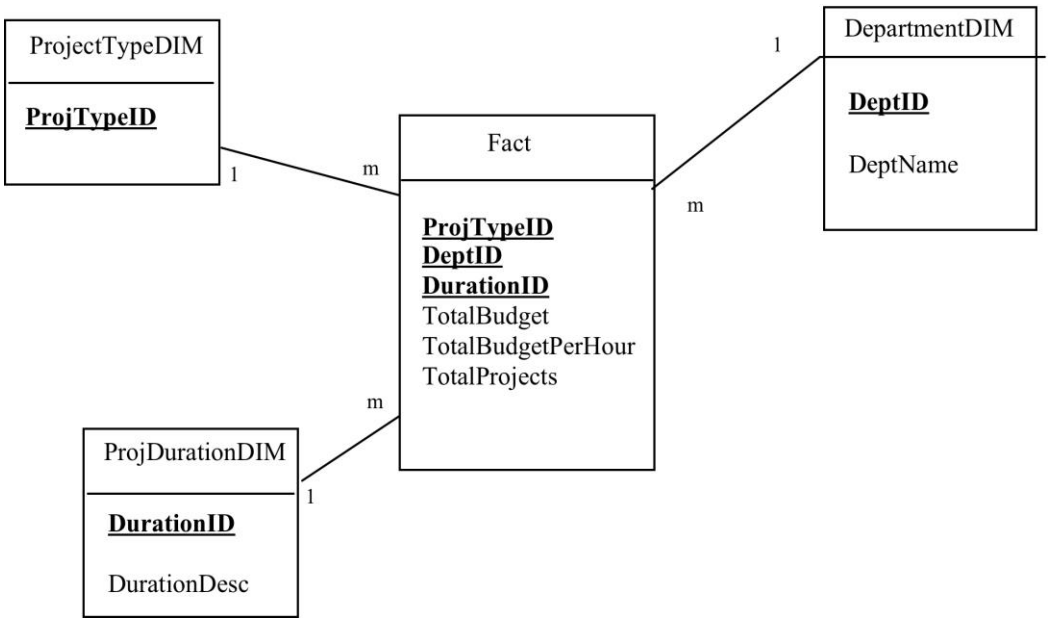
Based on the above requirements, the fact measures that the management is interested in are total budgets, average budget per hours, and total projects; and the dimensions are project type, department, and project duration. Assuming that the management classifies projects into short term (less than 10 days), medium term (between 10 and 30 days), and long term (more than 30 days).

Questions:

a. Draw a star schema containing dimensions and fact, together with their attributes.
b. Define the SQL statements for the implementation of the star schema.
c. Write the SQL statements to answer the above three query requirements.

Write your answer here:

Solution a



Solution b

```
-- create dimension tables
Create Table ProjectTypeDIM As
Select Distinct ProjType as ProjTypeID
From Project;

Create Table DepartmentDIM As
Select Dept# as DeptID, DeptName
From Department;

Create Table ProjDurationDIM
(DurationID   Number;
 DurationDesc Varchar2(20));

Insert Into ProjDurationDIM Values (1, 'Short Duration');
Insert Into ProjDurationDIM Values (2, 'Medium Duration');
Insert Into ProjDurationDIM Values (3, 'Long Duration');
```

Continue your answer here:

```
-- create TempFact
Create Table TempFact As
Select P.ProjType, P.Dept#, P.Start_Date, P.End_Date,
P.budgtotals$, P.Proj#, P.BudgPerHours
From Project P;

Alter Table  TempFact
Add (DurationID Number);

Update TempFact
Set DurationID = 1
Where End_Date - Start_Date < 10;

Update TempFact
Set DurationID = 2
Where End_Date - Start_Date >= 10
And End_Date - Start_Date <=30;

Update TempFact
Set DurationID = 3
Where End_Date - Start_Date > 30;

-- create Fact
Create Table Fact As
Select ProjType, Dept#, DurationID,
  SUM(budgtotals$) as TotalBudget,
  SUM(BudgPerHours) as TotalBudgetPerHour,
  COUNT(Proj#)as TotalProjects
From TempFact
Group By ProjType, Dept#, DurationID;
```

Continue your answer here:

Solution c:

- What is the total number of projects that are long duration?

```
Select D.DurationDesc, Sum(TotalProjects)
From Fact F, ProjDurationDIM D
Where F.DurationID = D.DurationID
Group By D.DurationDesc;
```

- What is the total budget for a certain project type?

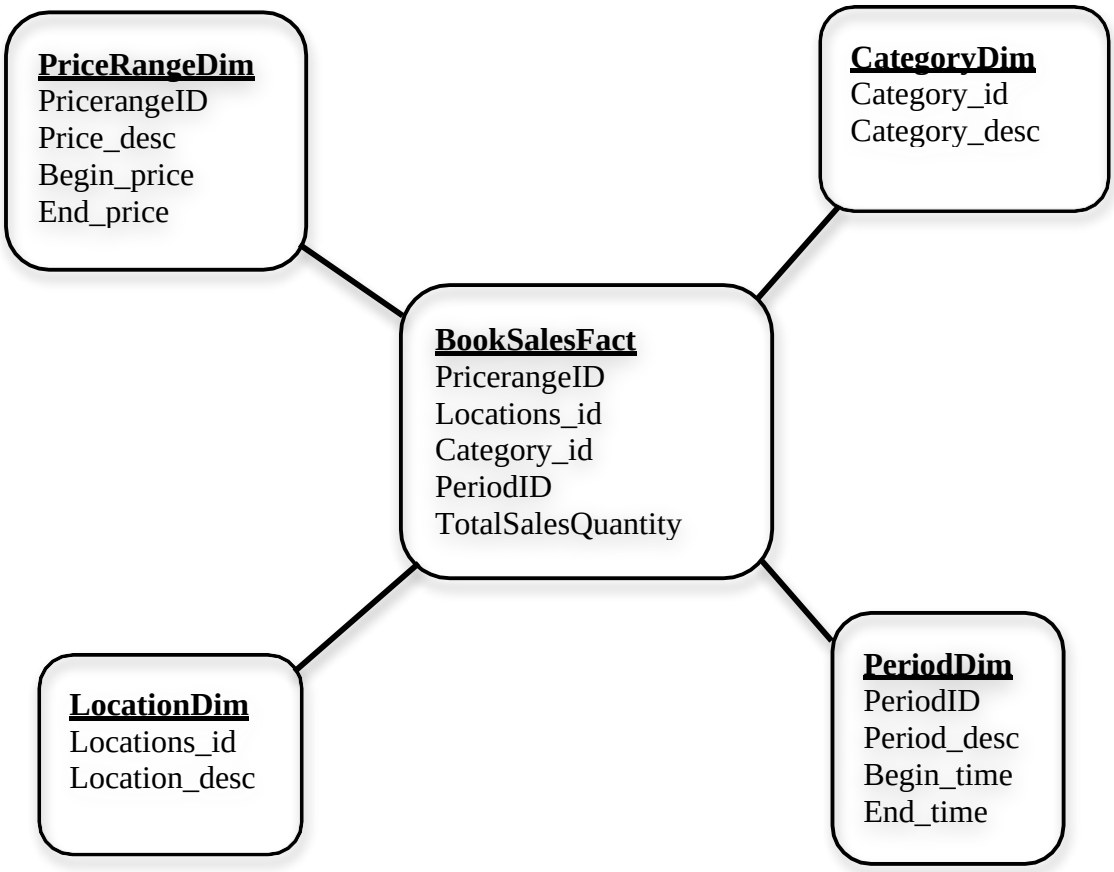
```
Select T.ProjTypeID, Sum(TotalBudget)
From Fact F, ProjTypeDIM T
Where F.ProjTypeID = T.ProjTypeID
Group By T.ProjTypeID;
```

- What is an average budget cost per hour of a certain department?

```
Select
   D.DeptName,
   Sum(TotalBudgetperHour)/Sum(TotalProjects) as Avg_Budget_Cost
From Fact F, DepartmentDIM D
Where F.DeptID = D.DeptID
Group By D.DeptName;
```

Question 5:

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 6:

Explain the differences between **`RANK()`**, **`DENSE_RANK()`**, and **`ROW_NUMBER()`** in OLAP SQL.

Write your answer here:

Suppose there is table called some_table, which consists of the following records:

```
create table some_table (
  a number,
  b varchar2(10)
);
```

a	b
5	*
2	*
5	*
3	*
5	*
3	*
2	*
4	#
8	#
4	#
8	#
4	#

```
select dense_rank() over (partition by b order by a) dr,
             rank() over (partition by b order by a) rk,
       row_number() over (partition by b order by a) rn,
       a,b
from some_table;
```

```
    DR       RK       RN        A B
---------- ---------- ---------- ----------
     1        1        1         4 #
     1        1        2         4 #
     1        1        3         4 #
     2        4        4         8 #
     2        4        5         8 #
     1        1        1         2 *
     1        1        2         2 *
     2        3        3         3 *
     2        3        4         3 *
     3        5        5         5 *
     3        5        6         5 *
     3        5        7         5 *
```

Continue your answers here:

Another example

DR	RK	RN
1	1	1
1	1	2
1	1	3
2	4	4
2	4	5
2	4	6
2	4	7
3	8	8
3	8	9
4	10	10
4	10	11
4	10	12

Notes:
•If the query is to get dense rank <= 5, then all the twelve records will be retrieved.
•If the query is to get rank <=5, then the first seven records will be retrieved.
•If the query is to get row number <=5, then only the first five records will be retrieved.
•Each of the above queries has a different semantic. Additionally, row number has to tie in with a primary key, so that the results will be deterministic. If, for example, the row number is based on $sales, and it happens that the first 10 records have the same $sales figure, then the query for row number <=5 will not always produce the same results.

THE END