Office Use Only

Monash University

Semester Examination Period
Faculty of Information Technology

Sample Exam SOLUTIONS

EXAM CODES: Sample-

TITLE OF PAPER: Sample-

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

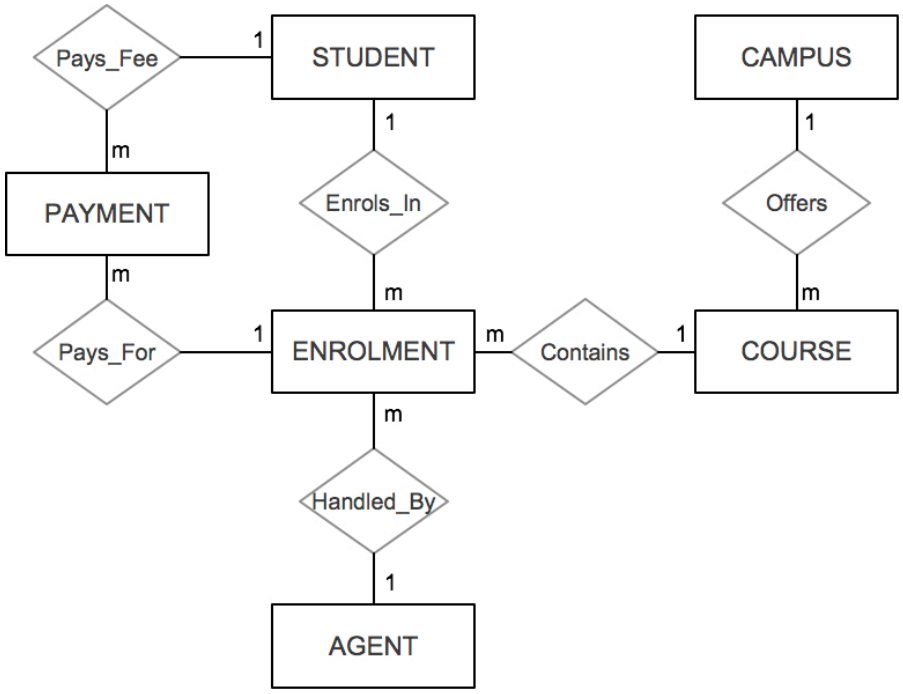
STUDENT ID __ __ __ __ __ __ __ __ __ DESK NUMBER __ __ __ __

Question 1:

Monash University is an international university. It has a dedicated office to handle international student matters, particularly enrolment, payments, and marketing campaigns. This office has an operational system that maintains all the details of international students enrolled at Monash. Payment details are also handled by this office. Basically, the operational system has the following features:

- Every student details are kept in the database. This includes the courses that the students enrol.
- As Monash University is a multi-campus university, some courses are offered in a different campus. Monash International Office handles international students of all campuses.
- Some international students coming to Monash are handled by an educational agent. This is particularly common for the first course that a student enrols in. Subsequent courses are not normally handled by an agent, because the students themselves deal directly with Monash University.
- International students pay tuition fees several times (normally once every semester) for each course they are doing.

An E/R diagram to show the current operational system is shown as follows:



The operational database that maintains the above system has following tables:

STUDENT (StuID, Lname, Fname, Address, Phone, DOB, Country, VisaExpDate, Sponsor)
CAMPUS (CampusID, Description, Address)
COURSE (CourseCode, CourseName, Duration, CourseLevel, *CampusID*)
AGENT (AgentNum, AgentName, Address, PhoneNum, ContactPerson)
ENROLMENT (EnrolID, StartSemesterYear, Status, *StuID*, *AgentNum*, *CourseCode*)
PAYMENT (PaymentNum, Date, Amount, *StuID*, *EnrolID*)

Monash International Office now requires a data warehouse for analysis purposes. The analysis is needed for identifying at least the following questions:

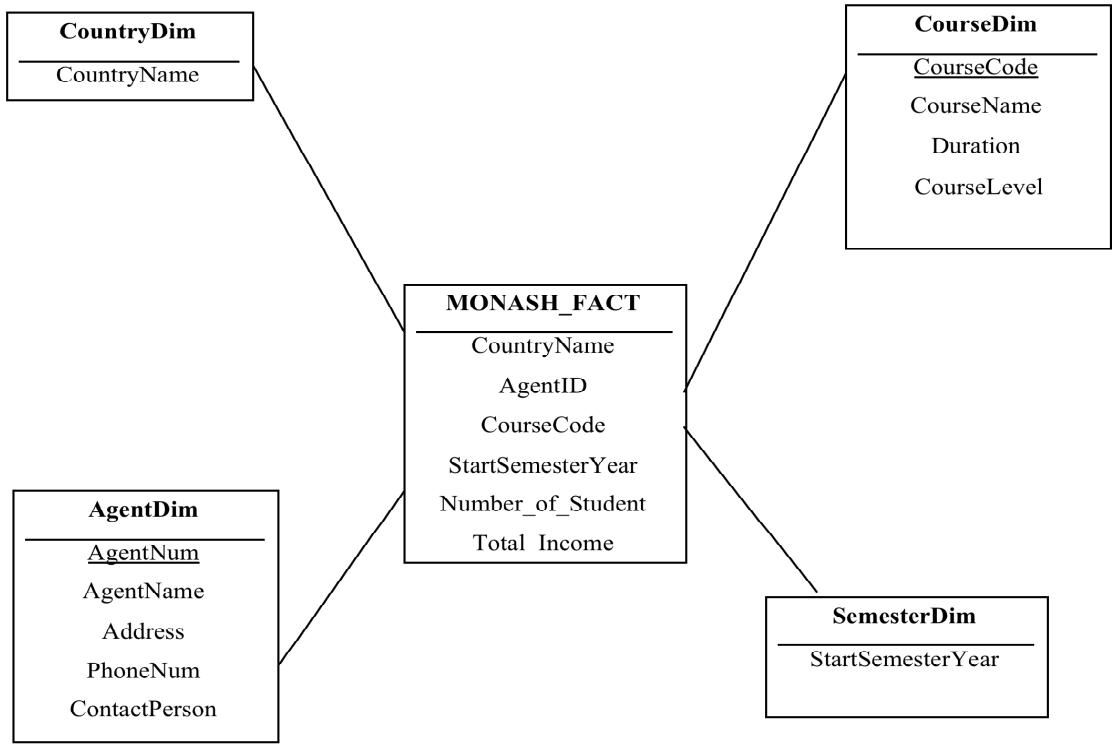
- How many students come from certain countries?
- What is the total income for certain postgraduate courses?
- How many students are handled by certain agents?
- How the number of enrolment of courses fluctuates across different semesteryears?

The first question could be used by the management to identify countries that may be targeted for future international marketing campaigns. The second question could be used by the financial office for further planning. The third question could be used in conjunction with future international marketing campaigns.

Questions:

a. Draw a star schema containing dimensions and fact, together with their attributes
b. Write the SQL statements for the implementation of the star schema

Write your answer here:



Continue your answer here:

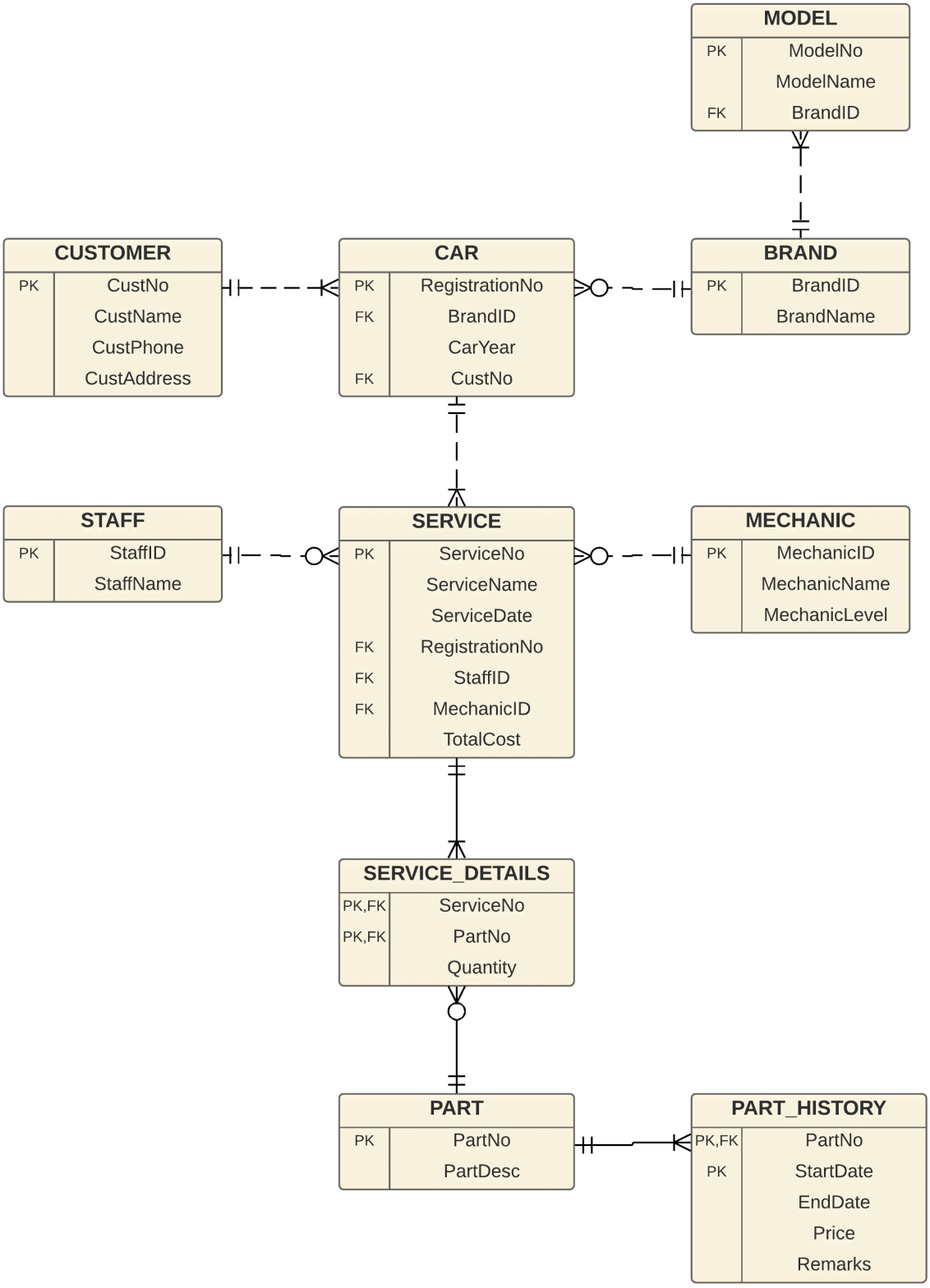
```
-- create the dimensions
create table CountryDim
as select distinct country from student;

create table CourseDim as
select CourseCode, CourseName, Duration, CourseLevel
from course;

create table AgentDim as
select * from agent;

create table SemesterDim
as select distinct StartSemesterYear from enrolment;
```

Continue your answer here:

```
-- create the fact
create table Monash_fact as
select
     s.Country, e.AgentNum, e.CourseCode, e.StartSemesterYear,
     count(s.StuID), sum(p.Amount)
from Agent a, Course c, Student s, Enrolment e, Payment p
where e.EnrolID=p.EnrolID
and e.StuID=s.StuID
and a.AgentNum=e.AgentNum
and e.CourseCode=c.CourseCode
Group by s.Country, e.AgentNum, e.CourseCode, e.StartSemesterYear
```

Question 2:

The "Auto Car Service" performs car services for their customers. Every time a service is conducted, a record is entered into the database. The information recorded includes service number, service name, date, car registration number, the staff who handled the service and liaised with the customer, the mechanic who performed the repair, and total cost of the repair.



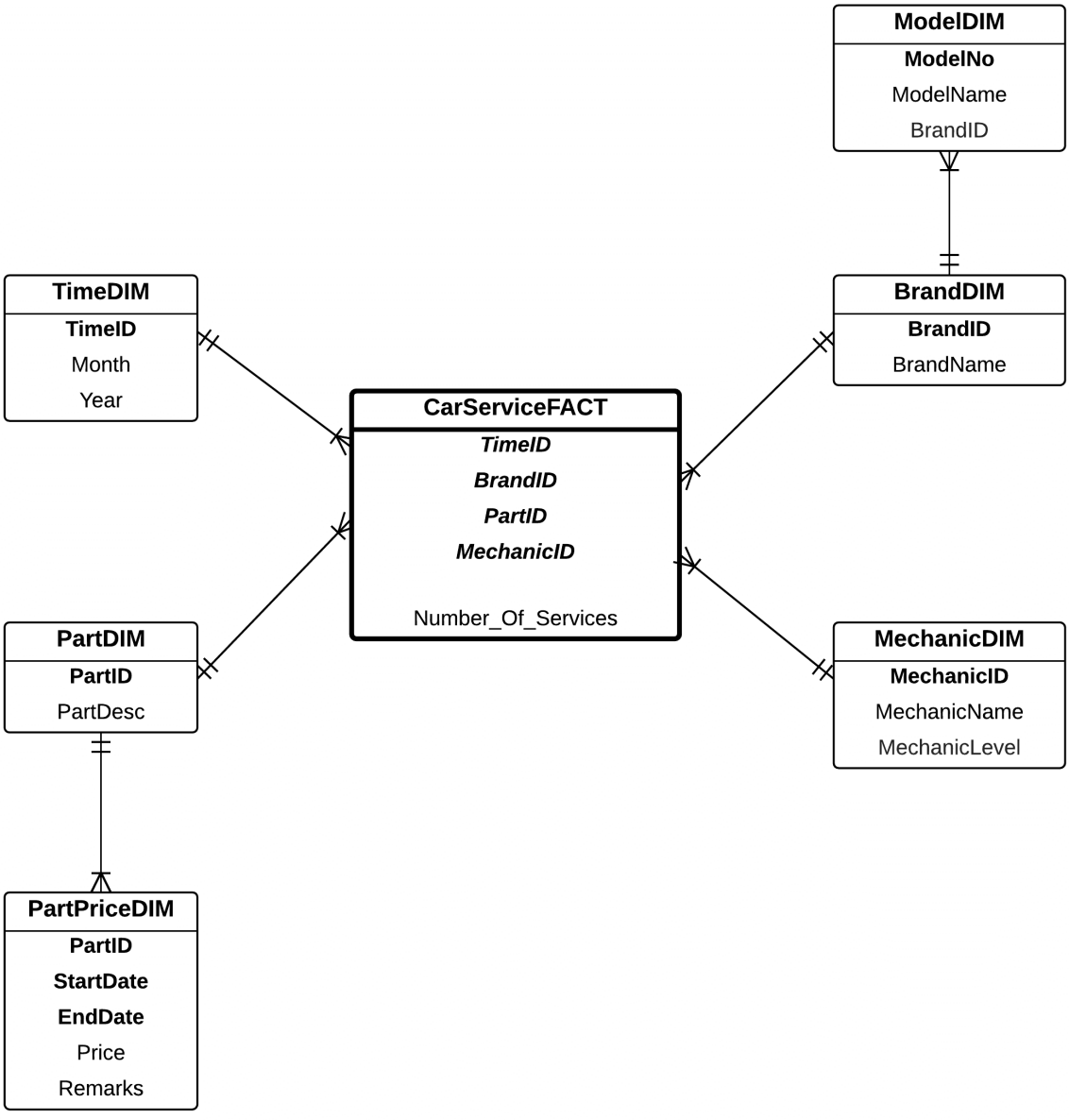
You are required to build a data warehouse to analyze the number of services for each car model, mechanic, month/year, and part. Note that every service may use several different parts. The price of each part may change from time to time, and hence a **history of prices** table is maintained in the database. The "Auto Car Service" has an operational database, as shown in the E/R diagram above:

Questions:

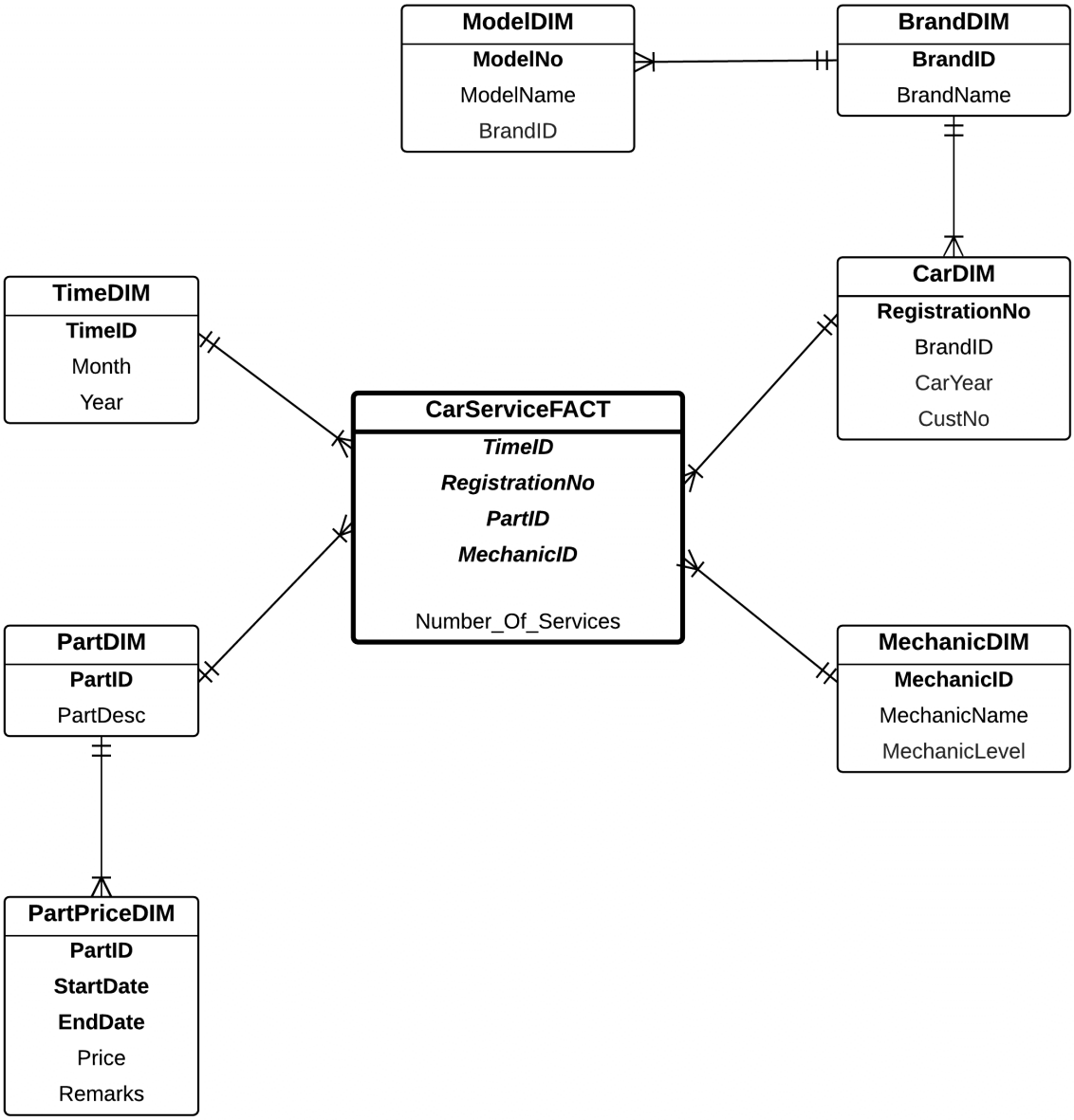
a. Draw a star schema for the "Auto Car Service" data warehouse, following the above requirements

b. Write the SQL statements to create (and populate with records) the dimension and fact tables

Write your answers here:



Continue your answer here:

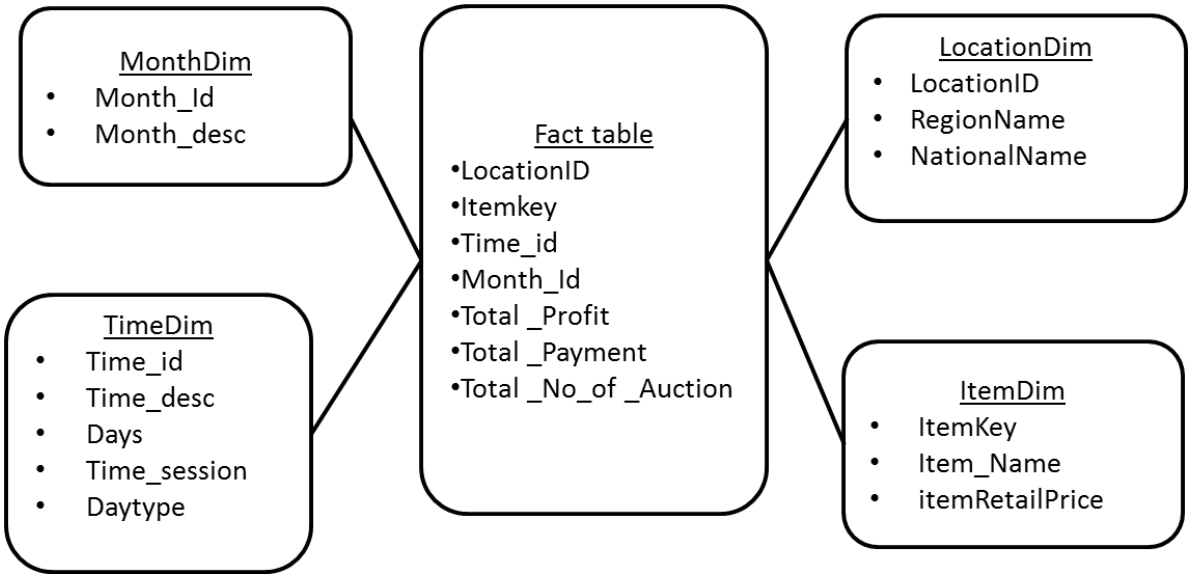


Continue your answer here:

```
create table TimeDIM
as select distinct
   to_char(ServiceDate, 'YYYYMM') as TimeID,
   to_char(ServiceDate, 'MM') as Month,
   to_char(ServiceDate, 'YYYY') as Year
from Service;

create table BrandDIM
as select * from Brand;

create table ModelDIM
as select * from Model;

create table MechanicDIM
as select * from Mechanic;

create table PartDIM
as select * from Part;

create table PartPriceDIM
as select * from PartHistory;

create table CarServiceFACT
as select
   to_char(S.ServiceDate, 'YYYYMM') as TimeID,
   B.BrandID,
   P.PartID,
   M.MechanicID
   Count(*) as Number_of_Services
from Service.S, Mechanic M, Car C, Part P, Service_Details SD, Brand B
where S.ServiceNo = SD.ServiceNo
and S.MechanicID = M.MechanicID
and S.RegistrationNo = C.RegistrationNo
and C.BrandID = B.BrandID
group by
   to_char(S.ServiceDate, 'YYYYMM') as TimeID,
   B.BrandID,
   P.PartID,
   M.MechanicID;
```

Question 3:

Given the following schema:



The tables (e.g. the fact and four dimensions) have been created and have also been populated with an adequate number of records.

Write the SQL statements for the following OLAP queries:

a. Show the top 3 total number of auctions by time sessions.
b. Show number of auctions (and subtotals) by month and region.
c. Display the total profit, total payment and total number of auction with cumulative sum for each item.

Write your answer here:

A)

```
SELECT * FROM
  (SELECT
      t.time_session,
      sum(f.total_number_of_auctions),
      rank() over (order by sum(f.total_number_of_auctions) DESC) as Rank_top_3
   FROM online_auction_fact f, timeDim t
   WHERE f.time_id = t.time_id
   GROUP BY t.time_session)
WHERE Rank_top_3 <=3;
```

B)

```
  SELECT
        DECODE(grouping(m.month_DESC),1,'All Month', m.month_DESC) AS MONTH,
        DECODE(grouping(l.r_name),1,'All Region', l.r_name) as Region,
        Sum(f.total_number_of_auctions) AS total_No_Auction
  FROM monthDIM M ,LocationDim l, online_auction_fact f
  WHERE f.Locationid=l.locationid
  AND f.month_id = m.month_ID
  GROUP BY ROLLUP(m.month_DESC,l.r_name);
```

Note: a GROUP BY CUBE will also be accepted, because the question does not specifically ask for a CUBE or ROLLUP.

Continue your answer here:

C)

```
SELECT
   i.i_name, f.month_id,
   sum(f.total_profit) as total_Profit,
   to_char (sum(sum(f.total_profit)) over (order by f.month_id
      rows unbounded Preceding),'9,999,999.99') as Cumulative_Profit,
   sum(f.total_payment) as total_payment,
   to_char(sum(sum(f.total_payment)) over(order by f.month_id
      rows unbounded preceding),'9,999,999,999') as cumulative_payment,
   sum(f.total_no_of_auctions)as total_number_of_Auction,
   to_char(sum(sum(f.total_number_of_auctions)) over(order by f.month_id
      rows unbounded preceding),'9,999,999,999') as cummulative_auctions
FROM online_auction_fact f,itemDim i
WHERE f.i_itemkey = i.i_itemkey
And i.i_name = 'ITEM1'
GROUP BY i.i_name, f.month_id;
```

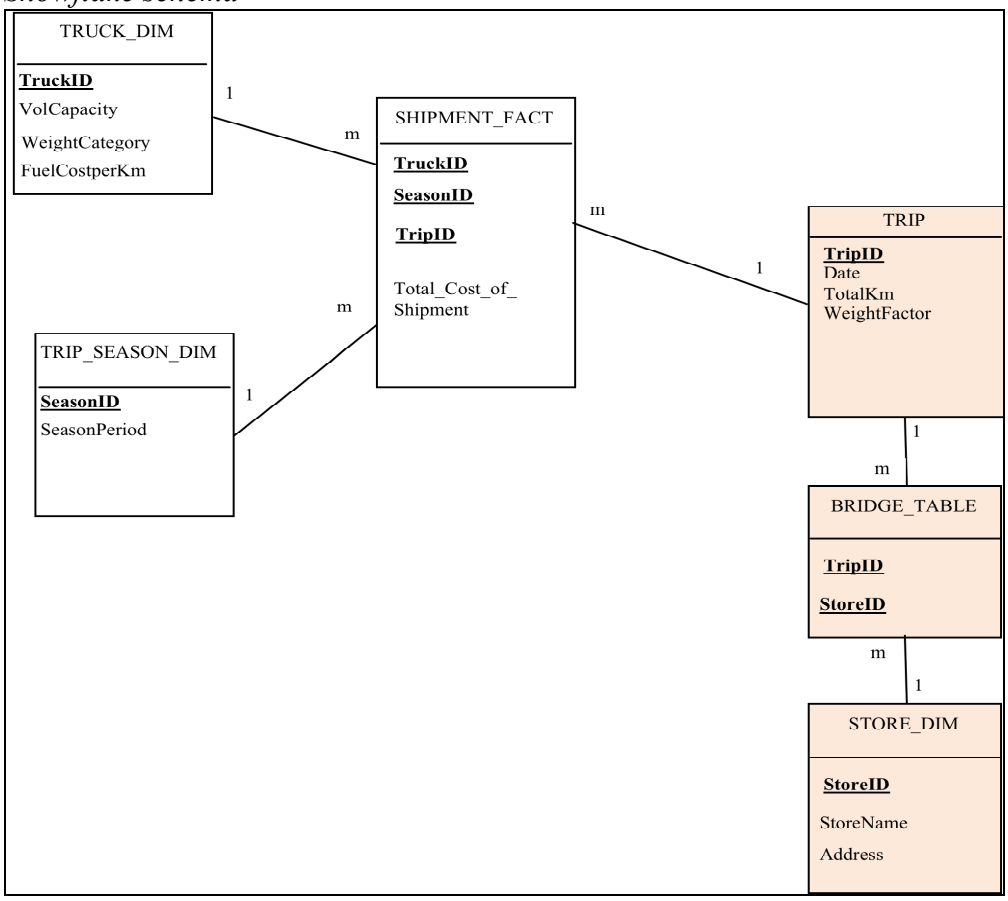
Alternatively, use PARTITION BY i.i_name:

```
SELECT
   i.i_name, f.month_id,
   sum(f.total_profit) as total_Profit,
   to_char (sum(sum(f.total_profit)) over
      (PARTITION BY i.i_name order by f.month_id
      rows unbounded Preceding),'9,999,999.99') as Cumulative_Profit,
   sum(f.total_payment) as total_payment,
   to_char(sum(sum(f.total_payment)) over
      (PARTITION BY i.i_name order by f.month_id
      rows unbounded preceding),'9,999,999,999') as cumulative_payment,
   sum(f.total_no_of_auctions)as total_number_of_Auction,
   to_char(sum(sum(f.total_number_of_auctions)) over
      (PARTITION BY i.i_name order by f.month_id
      rows unbounded preceding),'9,999,999,999') as cummulative_auctions
FROM online_auction_fact f,itemDim i
WHERE f.i_itemkey = i.i_itemkey
GROUP BY i.i_name, f.month_id;
```

Question 4:

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
Truck2     Autumn     Trip2                    855
```

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 5:

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

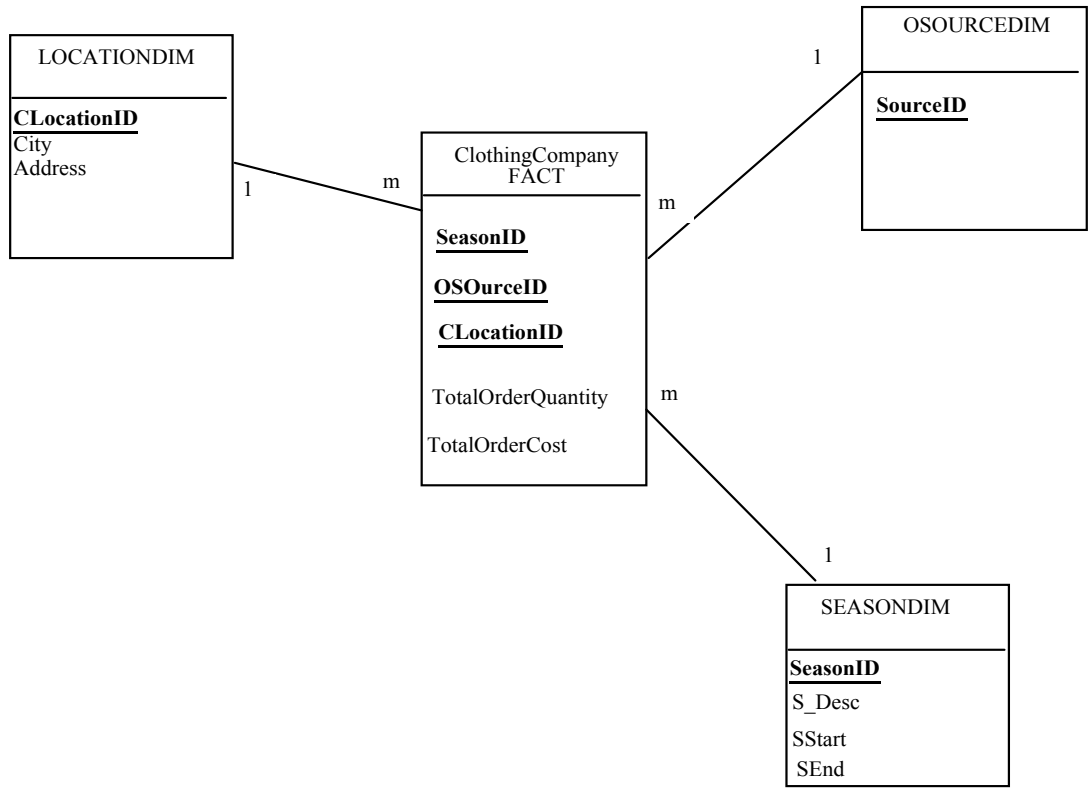
INVENTORY1

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.

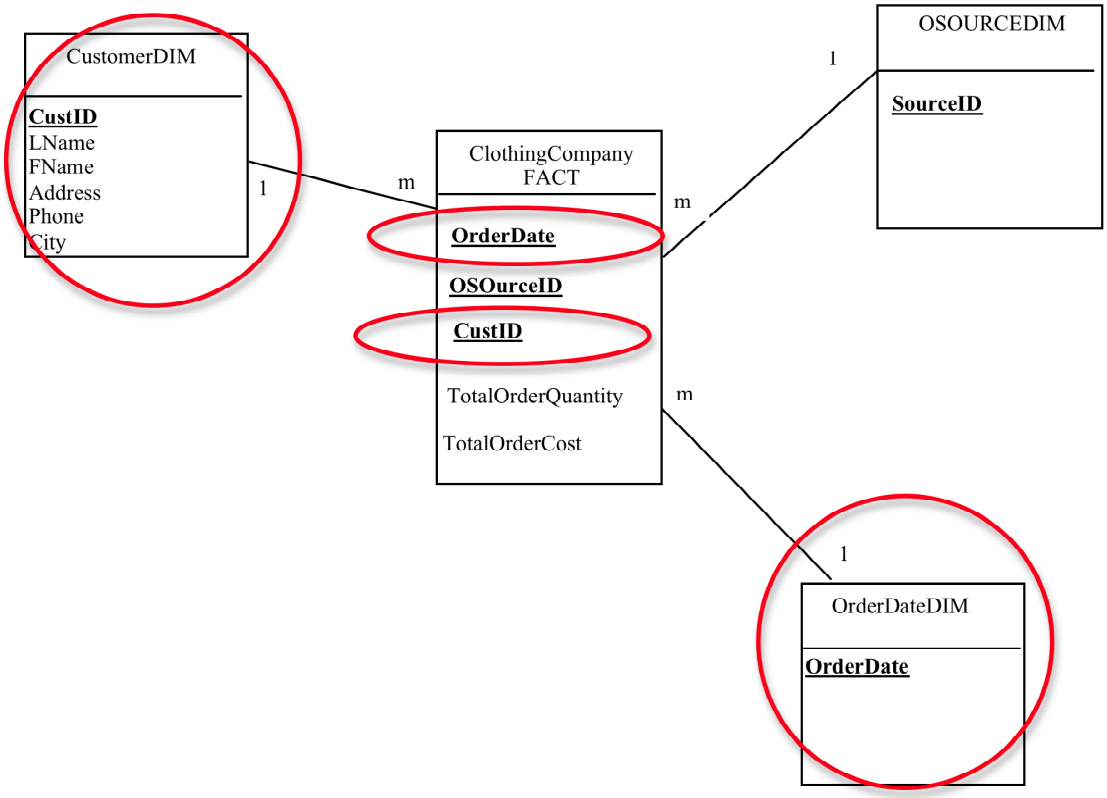
Write your answer here:

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.

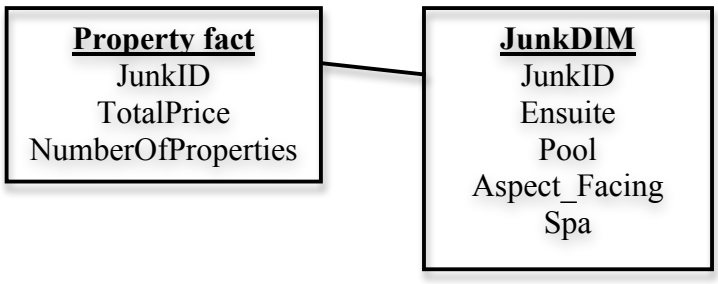
In Level-0, both dimensions need to be lowered down.

Level-0



Question 6:

Consider a star schema consisting of one fact and one junk dimension as follows:



The JunkDIM has a key attribute, called JunkID, which is a sequence number (e.g. 1, 2, …, *n*). The contents the JunkDIM table are as follows:

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 JunkDIM table has been created using the following SQL command:

```
Create Table JunkDIM
as select distinct Ensuite, Pool, Aspect_Facing, Spa
from dw.Property1;
```

However, the above JunkDIM table does not yet have a JunkID attribute.

Questions:

(a) Alter the JunkDIM table to include a JunkID attribute

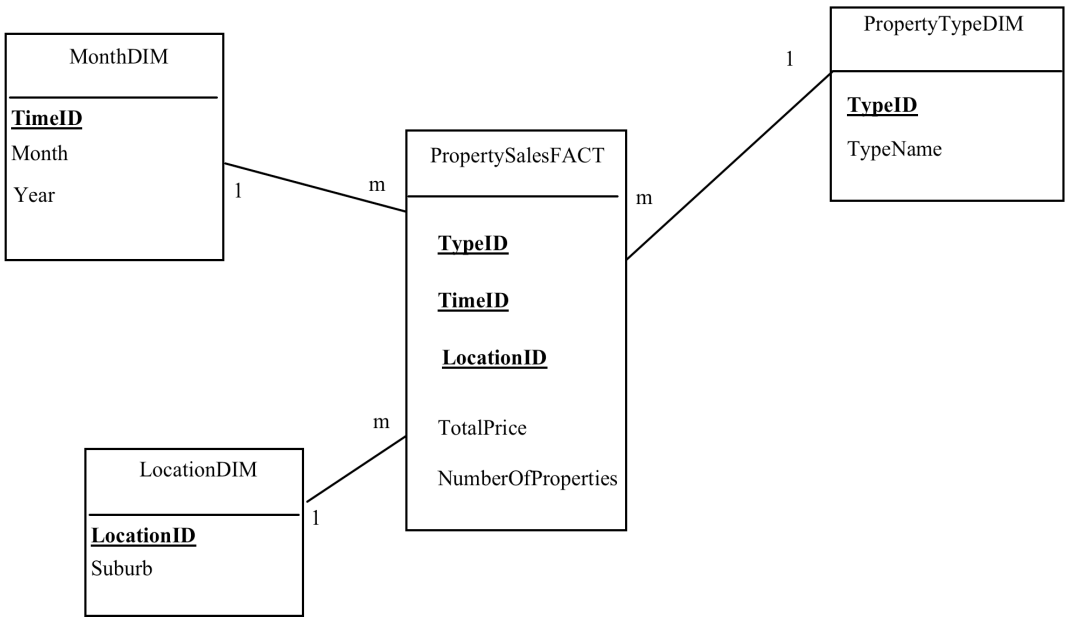
(b) Create a sequence and insert a sequence number to each record in the JunkID attribute

Write your answers here:

```
Alter Table JunkDim add (JunkID number(2));

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

Update JunkDim SET JunkID = seq_ID.nextval;
```

Question 7:

Given the following star schema:



The tables (e.g. Fact and three dimensions) have been created and have also been populated with an adequate number of records. The table names and attributes are shown in the star schema above.

Write the SQL for the following OLAP queries:

a. Display the top 10 average prices by suburb of property
b. Display the average price of properties by property type description and suburb. It is not required to show the subtotals or group totals or grand total

Write your answer here:

Solution a

```
Select *
From
      (Select P.LocationID, L.Suburb,
           Sum(F.TotalPrice)/Sum(F.NumberOfProperties) as AveragePrice,
           RANK() OVER
             (ORDER BY Sum(F.TotalPrice)/Sum(F.NumberOfProperties) DESC)
             as PROPERTY_RANK
       From     PropertyFact P, LocationDim L
       Where    P.LocationID = L.LocationID
       Group by P.LocationID, L.Suburb)
Where PROPERTY_RANK <= 10;
```

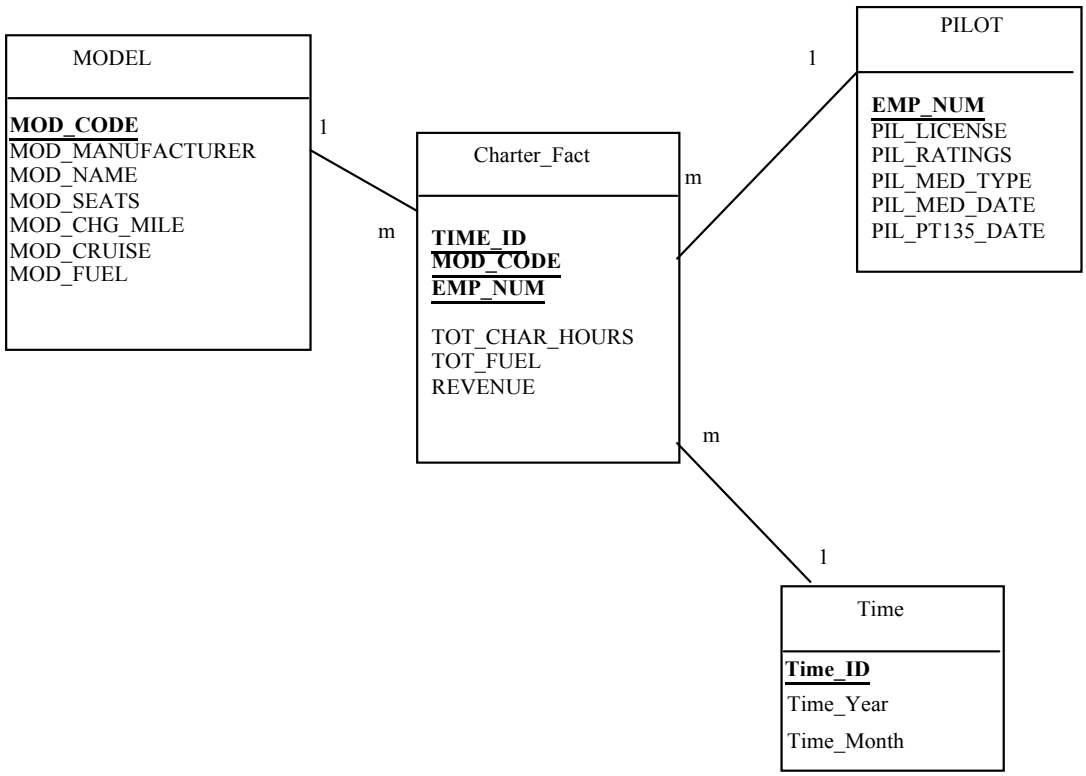
Solution b :

```
SELECT T.TypeName, L.Suburb,
       Sum(F.TotalPrice)/Sum(F.NumberOfProperties) as AveragePrice
FROM  PropertyFACT F, PropertyTypeDIM T, LocationDIM L
WHERE F.TypeID = T.TypeID
AND   F.LocationID = L.LocationID
GROUP BY T.TypeName, L.Suburb;
```

Question 8:

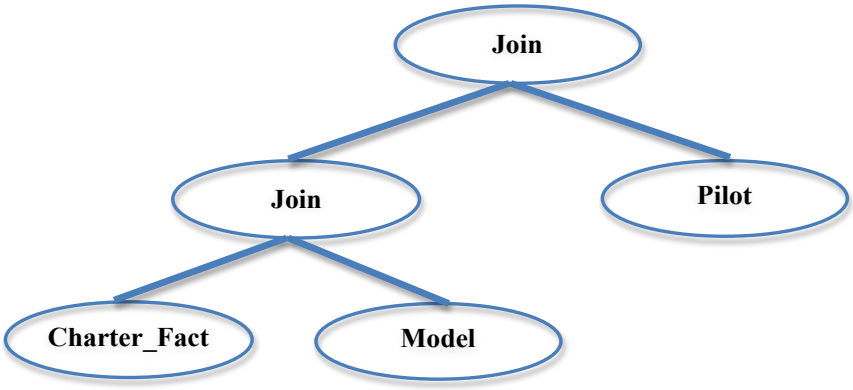
Given the following star schema, assume that we would like to produce a report that joins three tables: Charter_Fact, Model, and Pilot, using the following SQL query:

```
Select *
From Charter_Fact C, Model M, Pilot P
Where C.Mod_Code = M.Mod_Code
And C.Emp_Num = P.Emp_Num;
```

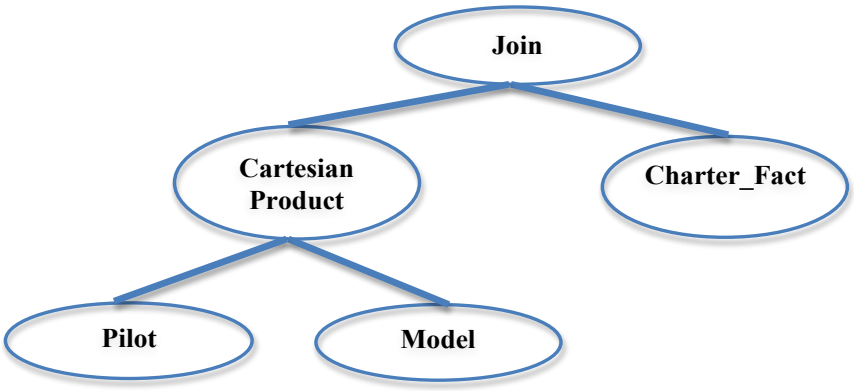


There are many options available to execute this join query. The following are two possible options:

Option 1:



Option 2:



Questions:

a. Explain the two query trees, and explain how the two query trees work
b. What is a Cartesian product in Option 2? Use some record numbers to illustrate your answer. Does the result of the Cartesian product have any particular meaning (in comparison with join operation, for example)
c. Under what circumstances that one option is better than the other. Use some record numbers to illustrate your answer

Write your answer here:

Solution a

Option 1 joins Charter_Fact with Model first.
Then the result of this join is joined with Pilot, which produces the final results.

Option 2 does a Cartesian product between the two dimensions (Pilot and Model).
The result of this Cartesian product is joined with the fact table, which produces the final results.

Solution b

- Cartesian product produces possible combinations between all records from the two tables.
- If there are 25 pilots and 10 aircraft models, the result of this product is a combination of all 25 pilots and 10 aircraft models, and therefore it will produce 250 records.
- The combination between each record of Pilot and Model does not have a particular meaning (unlike a join operation).

Continue your answer here:

Solution c

For **option 1**, the result of a join between Charter_Fact and Model will produce as many records as they are in the Charter_Fact table. For example, if Charter_Fact=10,000 records, and Model=10 records, the result of this join will produce 10,000 records.

However, for **option 2**, if Model=10 records, and Pilot=25 records, the Cartesian product will produce 250 records, which then be joined with Charter_Fact of 10,000 records, to produce 10,000 records in the query result.

So, if the dimension tables have relatively small number of records, option 2 is better, because in option 1, the temporary result between Join (Charter_Fact and Model) produces a large number of records, while we are trying to minimize number of records in the temporary.

BUT, if the dimensions have relatively large number of records, then the Cartesian product will produce gigantic temporary results, because Cartesian product needs to get all combinations between the two tables.

THE END