Data Science and Database Technology

Exam 2024-01-29

	EDUARD ANTONOVIC OCCHIPINTI 332100
Iniziato	lunedì, 29 gennaio 2024, 14:11
Terminato	lunedì, 29 gennaio 2024, 15:56
Tempo impiegato	1 ora 45 min.
Valutazione	**23,88** su un massimo di 32,00 (**75**%)

Domanda 1

Parzialmente corretta

Punteggio ottenuto 2,36 su 5,00

5 total points (penalty -15% for each wrong answer)

The following tables are provided:

EMPLOYEE(EID, FirstName, LastName, Department, Salary)
PROJECT(PID, ProjectName, ProjectType, Budget, TeamSize)
EMPLOYEE-WORKS-ON(EID, PID, DateBeginAssignment)
CLIENT(CID, ClientName, Industry)
CONTRACT(PID, CID, ContractAmount, ContractDate)

Assume the following cardinalities:

- card(EMPLOYEE) = 10^5 tuples
  - distinct values of Department = 10
- card(PROJECT) = 100 tuples
  - distinct values of TeamSize = 5
- card(EMPLOYEE-WORKS-ON) = $6 \cdot 10^5$ tuples
  - MIN(DateBeginAssignment) = 1/1/2004, MAX(DateBeginAssignment) = 31/12/2023
- card(CLIENT) = $5 \cdot 10^4$ tuples
  - distinct values of Industry = 50
- card(CONTRACT) = $2 \cdot 10^5$ tuples

Furthermore, assume the following reduction factor for the having clauses:

```
Having COUNT(DISTINCT ProjectType) > 1 = 1/3
Having SUM(ContractAmount) > 10k = 1/2
```

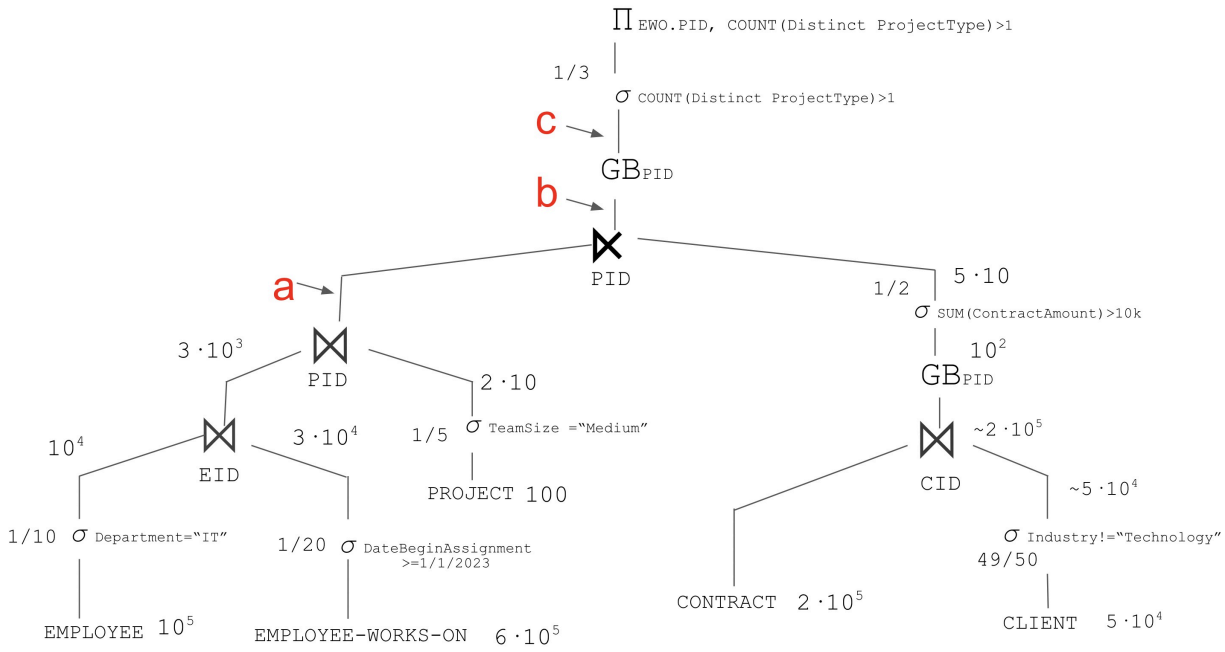
Consider the following query:

```
SELECT EWO.PID, COUNT(DISTINCT P.ProjectType)
FROM EMPLOYEE E, PROJECT P, EMPLOYEE-WORKS-ON EWO
WHERE E.EID = EWO.EID
    AND P.PID = EWO.PID
    AND P.TeamSize = 'Medium'
    AND EWO.DateBeginAssignment >= 1/1/2023
    AND E.Department = 'IT'
    AND EWO.PID IN (SELECT C1.PID
                    FROM CONTRACT C1, CLIENT CL
                    WHERE C1.CID = CL.CID
                        AND CL.Industry != 'Technology'
                    GROUP BY C1.PID
                    HAVING SUM(ContractAmount) > 10k)
GROUP BY EWO.PID
HAVING COUNT(DISTINCT P.ProjectType) > 1;
```

Cardinalities

(1.5 points, penalty -15% per each wrong answer)

The figure below represents the query tree for the query above.



Specify the correct answer for the cardinality of **(a)**:

- $6 \cdot 10^2$ ✓
- $6 \cdot 10^3$
- $6 \cdot 10^4$
- $6 \cdot 10^5$

Punteggio ottenuto 5,00 su 5,00

La risposta corretta è: $6 \cdot 10^2$

Specify the correct answer for the cardinality of **(b)**:

- $\sim 5 \cdot 10^3$
- $\sim 3 \cdot 10^2$ ✓
- $\sim 5 \cdot 10$
- $\sim 3 \cdot 10^3$

Punteggio ottenuto 5,00 su 5,00

La risposta corretta è: $\sim 3 \cdot 10^2$

Specify the correct answer for the cardinality of **(c)**:

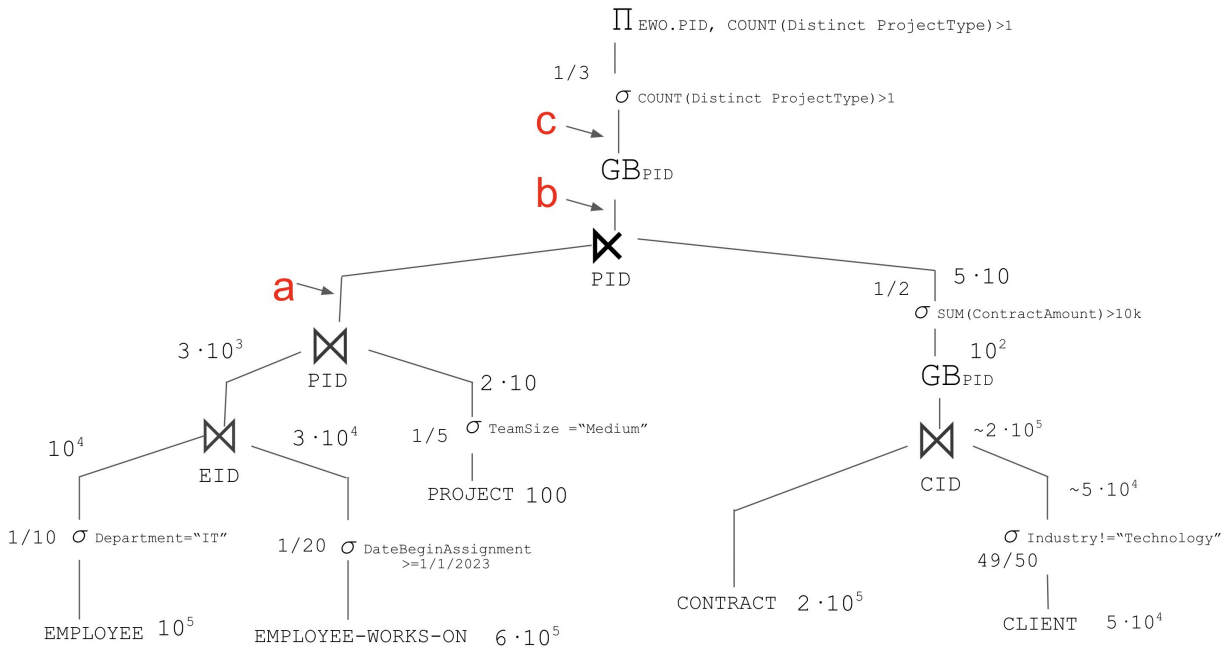
- $3 \cdot 10^2$
- $< 10^2$
- $3 \cdot 10^3$
- $< 2 \cdot 10$ ✓

Punteggio ottenuto 5,00 su 5,00

La risposta corretta è: <2 · 10

Indexes

(1.5 points, penalty -15% for each wrong answer)
The figure below represents the query tree for the query above.



For each table, select one or more secondary physical structures to increase query performance (if possible) among the options below.

Table EMPLOYEE

- None - secondary physical structures would not increase query performance.
- CREATE INDEX IndexB ON EMPLOYEE(Department) - B+-Tree
- CREATE INDEX IndexA ON EMPLOYEE(Department) - HASH ✓

Punteggio ottenuto 3,00 su 3,00

La risposta corretta è: CREATE INDEX IndexA ON EMPLOYEE(Department) - HASH

Tabella EMPLOYEE-WORKS-ON

- CREATE INDEX IndexD ON EMPLOYEE-WORKS-ON(DateBeginAssignment) - B+-Tree
- None - secondary physical structures would not increase query performance.
- CREATE INDEX IndexC ON EMPLOYEE-WORKS-ON(DateBeginAssignment) - HASH ✗

Punteggio ottenuto -0,45 su 3,00

La risposta corretta è: CREATE INDEX IndexD ON EMPLOYEE-WORKS-ON(DateBeginAssignment) - B+-Tree

Tabella PROJECT

- None - secondary physical structures would not increase query performance. ✓
- CREATE INDEX IndexE ON PROJECT(ProjectType) - HASH
- CREATE INDEX IndexF ON PROJECT(ProjectType) - B+-Tree

Punteggio ottenuto 3,00 su 3,00

La risposta corretta è: None - secondary physical structures would not increase query performance.

Tabella CLIENT

- CREATE INDEX IndexG ON CLIENT(Industry) - HASH
- None - secondary physical structures would not increase query performance. ✓
- CREATE INDEX IndexH ON CLIENT(Industry) - B+-Tree

Punteggio ottenuto 3,00 su 3,00

La risposta corretta è: None - secondary physical structures would not increase query performance.

Tabella PROJECT

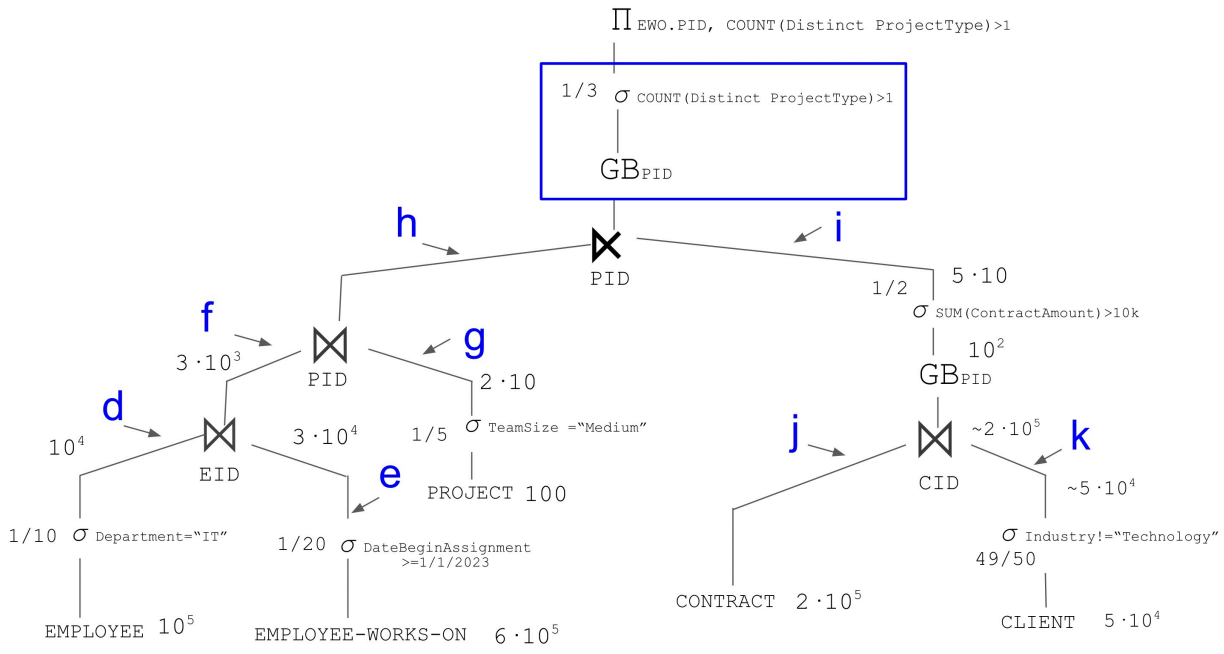
- CREATE INDEX IndexI ON PROJECT(TeamSize) - HASH
- None - secondary physical structures would not increase query performance. ✓
- CREATE INDEX IndexJ ON PROJECT(TeamSize) - B+-Tree

Punteggio ottenuto 3,00 su 3,00

La risposta corretta è: None - secondary physical structures would not increase query performance.

Anticipation Group By

(2 points, penalty -15% for each wrong answer)
The figure below represents the query tree for the query above.



Analyze the group by anticipation of **GROUP BY EWO.PID HAVING COUNT(DISTINCT P.ProjectType) > 1** represented in the box. Select the solution that **allows maximum efficiency** in executing the query (if any).

- It is possible to anticipate it in branch h
- It is possible to anticipate it in branch i
- It is possible to anticipate it in branch g ✗
- It is possible to anticipate it in branch e
- It is possible to anticipate it in branch d
- It is possible to anticipate it in branch j
- It is not possible to anticipate the Group BY GROUP BY EWO.PID HAVING COUNT(DISTINCT P.ProjectType) > 1
- It is possible to anticipate it in branch k
- It is possible to anticipate it in branch f

Punteggio ottenuto -3,00 su 20,00

La risposta corretta è: It is possible to anticipate it in branch h

1) La risposta corretta è : $6 \cdot 10^2$
2) La risposta corretta è : $\sim 3 \cdot 10^2$
3) La risposta corretta è : $< 2 \cdot 10$
4) La risposta corretta è : CREATE INDEX IndexA ON EMPLOYEE(Department) - HASH
5) La risposta corretta è : CREATE INDEX IndexD ON EMPLOYEE-WORKS-ON(DateBeginAssignment) - B+-Tree
6) La risposta corretta è : None - secondary physical structures would not increase query performance.
7) La risposta corretta è : None - secondary physical structures would not increase query performance.
8) La risposta corretta è : None - secondary physical structures would not increase query performance.
9) La risposta corretta è : It is possible to anticipate it in branch h

Domanda 2

Completo

Punteggio ottenuto 9,00 su 11,00

11 points (no penalty for a wrong answer)

The following data warehouse stores information on patient accesses (characterized by age range and gender) to the emergency room of hospitals located in Italy. Each hospital belongs to a single type (public or private) and there may be one or more departments among pediatrics, geriatrics, orthopedics, and cardiology (these attributes are Boolean and represent a configuration). In the Triage a color code (among white, green, blue, yellow, and red) is assigned and the mode of access to the emergency room (by ambulance or independently) is tracked. When someone accesses the emergency room, a diagnosis characterized by one or more specialistic examinations is assigned.

The data warehouse is characterized by the following conceptual schema and corresponding logical schema. Each event is characterized by four measures: number of patients, number of visits, number of physicians who made the visits and hours spent in the emergency room (TotalHours).

PATIENT_CHARACTERISTICS (CodCP, AgeRange, Gender)
TRIAGE (CodTriage, AccessMode, ColorCode)
HOSPITAL (CodO, Hospital, Typology, City, Province, Region, Pediatrics, Geriatrics, Orthopedics, Cardiology)
DIAGNOSIS (CodD, Diagnosis, Illness)
EXAM-DIAGNOSIS (CodD, SpecialisticExamination)
TIME (CodT, Date, DayOfWeek, Week, DayOfMonth, Holiday, Month, MonthOfYear, 4-Months, 6-Months, Year)
EMERGENCY-ROOM-ACCESS (CodCP, CodTriage, CodO, CodD, CodT, #Patients, #Visits, #Physicians, TotalHours)

Given the previous logical schema, write in the box below the following extended SQL queries, separated by a space.

Query 1 (3 points)

Considering female patients in the 24-34 age range, separately by four-month period (4-Months) and province, show:

- the average number of visits per physician, the average number of hours spent in the emergency room per patient
- the percentage of the number of hours spent in the emergency room compared to the total number of hours spent in the emergency room by region

- the cumulative number of visits for increasing four-month period (4-Months), separately by year.

Perform the analysis separately by mode of access (AccessMode) to the emergency room.

Query 2 (4 points)

Considering the accesses to the emergency room with red code (ColorCode) made by female patients, separately by diagnosis, province and four-month period (4-Months), show:

- the average monthly number of visits
- the average number of patients per physician
- the total number of patients regardless of diagnosis and region
- the total number of hours spent in the emergency room regardless of the four-month period (4-Months).

Perform the analysis separately for public and private hospitals.

Query 3 (4 points)

Separately by month and city where a hospital is located, show:

- the average daily number of hours spent in the emergency room
- the percentage of patients compared to the total of the semester (6-Months)
- the total number of patients separately by province where the hospital is located

Assign to each record:

- the rank separately by province. Position 1 should be assigned to the record with the highest average number of visits per physician.
- the rank separately per semester. Position 1 should be assigned to the record with the lowest number of patients.

---

1)
```
SELECT 4-Months, Province,
    SUM(#Visits) / SUM(#Physicians), SUM(TotalHours) / SUM(#Patients),
    100 * SUM(TotalHours) / SUM(SUM(TotalHours) OVER (PARTITION BY 4-Months, Region)),
    SUM(SUM(#Visits) OVER (PARTITION BY Province, Year ORDER BY 4-Months ROWS
UNBOUND PRECEDING)
FROM PATIENT_CHARACTERISTICS pc, TIME t, EMERGENCY-ROOM-ACCESS era,
HOSTPITAL h
WHERE pc.CodCP == era.CodCP AND t.CodT == era.CodT AND era.CodO == h.CodO AND
pc.AgeRange == '24-34' AND pc.Gender == 'F'
GROUP BY 4-Months, Province, Region
```

2)
```
SELECT Diagnosis, Province, 4-Months,
    SUM(#Visits) / COUNT(DISTINCT Month),
    SUM(#Patients) / SUM(#Physiccians),
    SUM(SUM(#Patients) OVER (PARTITION BY 4-Months)),
```

```
    SUM(SUM(TotalHours) OVER (PARTITION BY Diagnosis, Province))
FROM TIME t, TRIAGE tr, DIAGNOSIS d,  EMERGENCY-ROOM-ACCESS era, PATIENT-
CHARCATERISTICS pc, HOSPITAL h
WHERE t.CodT == era.CodT AND tr.CodTriage == era.CodTriage AND d.CodD == era.CodD AND
era.CodCP == pc.CodCP AND era.CodO == h.CodO AND ColorCode == 'RED' AND pc.Gender
== 'F'
GROUP BY Diagnosis, Province, 4-Months, Month

3)
SELECT Month, City,
    SUM(TotalHours) / COUNT(DISTINCT Date)
    100 * SUM(#Patients) / SUM(SUM(#Patients) OVER (PARTITION BY City, 6-Months)),
    SUM(#Patients) / SUM(SUM(#Patients) OVER (PARITTION BY Month, Province)),
    RANK() OVER (PARTITION BY Month, Province ORDER BY SUM(#Visits) /
SUM(#Phyisicians) DESC),
    RANK() OVER (PARTITION BY 6-Months, City ORDER BY SUM(#Patients) ASC)
FROM TIME t, HOSPITAL o, EMERGENCY-ROOM-ACCESS era
WHERE era.CodT == t.CodT, era.CodO == o.CodO
GROUP BY Month, City, Date, 6-Months, Province
```

Query 1

SELECT 4-Months, Province, AccessMode

SUM (#Visits)/SUM(#Physicians), SUM(TotalHours)/SUM(#Patients),

100*SUM(TotalHours)/ SUM(SUM(TotalHours)) OVER (PARTITION BY 4-Months, AccessMode, Region),

SUM(SUM(#Visits)) OVER (PARTITION BY Province, AccessMode, Year ORDER BY 4-Months ROWS UNBOUNDED PRECEDINGS)

FROM PATIENT_CHARACTERISTICS PC, EMERGENCY-ROOM-ACCESS E, HOSPITAL H, TIME TI, TRIAGE T

WHERE PC.CodCP= E.CodCP AND H.CodO=E.CodO AND TI.CodT=E.CodT AND T.CodTirage=E.CodTriage AND Genre ='F' and AgeRange ='24-34'

GROUP BY 4-Months, Province, AccessMode, Region, Year

Query 2

SELECT Diagnosis, Province, 4-Months, Typology

SUM (#Visits)/COUNT(DISTICT Month),

SUM(#Patients)/SUM(#Physicians),

SUM(SUM(#Patients)) OVER (PARTITION BY 4-Months, Typology),

SUM(SUM(TotalHours)) OVER (PARTITION BY Diagnosis, Province, Typology)

FROM PATIENT_CHARACTERISTICS PC, EMERGENCY-ROOM-ACCESS E, HOSPITAL H, TIME TI, DIAGNOSIS D, TRIAGE T

WHERE PC.CodCP= E.CodCP AND H.CodO=E.CodO AND Ti.CodT=E.CodT AND E.CodTriage=T.CodTriage D.CodD=E.CodD AND

Genre ='F' AND ColorCode =’Red'

GROUP BY Diagnosis, Province, 4-Months, Typology

Query 3

SELECT Month, City,

SUM(TotalHours)/COUNT(DISTINCT Date),

100*SUM(#Patients)/SUM(SUM(#Patients)) OVER (PARTITION BY 6-Months, City),

SUM(SUM(#Patients)) OVER (PARTITION BY Month, Province)

RANK() OVER (PARTITION BY Province ORDER BY SUM(#Visits)/SUM(#Physicians) DESC)

RANK() OVER (PARTITION BY 6-Months ORDER BY SUM(#Patients))

FROM EMERGENCY-ROOM-ACCESS E, HOSPITAL H, TIME T

WHERE H.CodO=E.CodO AND T.CodT=E.CodT

GROUP BY Month, City, 6-Months, Province

Commento:
1)

SELECT 4-Months, Province,

SUM(#Visits) / SUM(#Physicians), SUM(TotalHours) / SUM(#Patients),

100 * SUM(TotalHours) / SUM(SUM(TotalHours) OVER (PARTITION BY 4-Months, Region**, AccessMode**)),

SUM(SUM(#Visits) OVER (PARTITION BY Province, Year **AccessMode** ORDER BY 4-Months ROWS UNBOUND PRECEDING)

FROM PATIENT_CHARACTERISTICS pc, TIME t, EMERGENCY-ROOM-ACCESS era, HOSTPITAL h

WHERE pc.CodCP == era.CodCP AND t.CodT == era.CodT AND era.CodO == h.CodO AND pc.AgeRange == '24-34' AND pc.Gender == 'F'

GROUP BY 4-Months, Province, Region **AccessMode, Year**

2)

SELECT Diagnosis, Province, 4-Months,

SUM(#Visits) / COUNT(DISTINCT Month),

SUM(#Patients) / SUM(#Physiccians),

SUM(SUM(#Patients) OVER (PARTITION BY 4-Months**, Typology**)),

SUM(SUM(TotalHours) OVER (PARTITION BY Diagnosis, Province**, Typology**))

FROM TIME t, TRIAGE tr, DIAGNOSIS d, EMERGENCY-ROOM-ACCESS era, PATIENT-CHARCATERISTICS pc, HOSPITAL h

WHERE t.CodT == era.CodT AND tr.CodTriage == era.CodTriage AND d.CodD == era.CodD AND era.CodCP == pc.CodCP AND era.CodO == h.CodO AND ColorCode == 'RED' AND pc.Gender == 'F'

GROUP BY Diagnosis, Province, 4-Months, ~~**Month**~~**, Typology**

3)

SELECT Month, City,

SUM(TotalHours) / COUNT(DISTINCT Date)

100 * SUM(#Patients) / SUM(SUM(#Patients) OVER (PARTITION BY City, 6-Months)),

SUM(#Patients) / SUM(SUM(#Patients) OVER (PARITTION BY Month, Province)),

RANK() OVER (PARTITION BY ~~**Month**~~, Province ORDER BY SUM(#Visits) / SUM(#Phyisicians) DESC),

RANK() OVER (PARTITION BY 6-Months, ~~**City**~~ ORDER BY SUM(#Patients) ASC)

FROM TIME t, HOSPITAL o, EMERGENCY-ROOM-ACCESS era

WHERE era.CodT == t.CodT, era.CodO == o.CodO

GROUP BY Month, City, ~~**Date**~~, 6-Months, Province

Domanda 3

Risposta errata

Punteggio ottenuto -0,23 su 1,50

Theory (1.5 points, -15% penalty for a wrong answer)

Notation:

rN(V): read of object V by transaction N
wN(V): write of object V by transaction N

The following schedule of 3 transactions is given:

S = W2(z) W0(x) W2(z) W1(z) R2(y) R1(y) R2(x) R1(y) R0(z) W0(y)

S is conflict serializable because it is conflict equivalent to the serial schedule:

- (a) W2(z), W2(z), R2(y), R2(x), W1(z), R1(y), R1(y), W0(x), R0(z), W0(y) ✗
- (b) W2(z), W2(z), R2(y), R2(x), W0(x), R0(z), W0(y), W1(z), R1(y), R1(y)
- (c) W0(x), R0(z), W0(y), W2(z), W2(z), R2(y), R2(x), W1(z), R1(y), R1(y)
- (d) S is not conflict serializable
- (e) W1(z), R1(y), R1(y), W2(z), W2(z), R2(y), R2(x), W0(x), R0(z), W0(y)
- (f) W1(z), R1(y), R1(y), W0(x), R0(z), W0(y), W2(z), W2(z), R2(y), R2(x)
- (g) W0(x), R0(z), W0(y), W1(z), R1(y), R1(y), W2(z), W2(z), R2(y), R2(x)

Risposta errata.

La risposta corretta è: S is not conflict serializable

Domanda 4

Risposta non data

Punteggio max.: 1,50

Theory (1.5 points, -15% penalty for a wrong answer)

Definitions

Precision(C): fraction of elements correctly classified for class C in all folds, out of all elements assigned to C in all folds
Recall(C): fraction of elements correctly classified for class C in all folds, out of all elements belonging to C in all folds

We have a dataset consisting of 6000 samples of which 2400 belong to the positive class (1) and 3600 belong to the negative class (0). We are evaluating a classifier by stratified 3-fold cross-validation (maintaining class distribution). The classifier obtains the following results for each fold:

1. 600 correct predictions for class 1, 1000 correct predictions for class 0
2. 500 correct predictions for class 1, 900 correct predictions for class 0
3. 600 correct predictions for class 1, 900 correct predictions for class 0

Which of the following statements regarding **class 1** is correct?

- (a) In at least one fold the recall is less than 0.6
- (b) Precision is lower than average recall
- (c) Precision is higher than average recall
- (d) No answer is correct
- (e) In at least one fold the precision is less than 0.5
- (f) The recall is lower than 0.6
- (g) The precision is greater than 0.9
- (h) In at least one fold the precision exceeds 0.8

Risposta errata.

La risposta corretta è: Precision is lower than average recall

Domanda 5

Risposta corretta

Punteggio ottenuto 1,50 su 1,50

Theory (1.5 points, -15% penalty for a wrong answer)

MAX Linkage policy states that the distance between two clusters is calculated as:

$$dist(X, Y) = max(dist(x, y))$$

where $x \in X, y \in Y$ and $dist(x, y)$ is function that calculates the distance between two points.

Given the following matrix of distances between 5 points:

	a	b	c	d	e
a	0	9	7	10	13
b	9	0	6	16	4
c	7	6	0	8	6
d	10	16	8	0	17
e	13	4	6	17	0

How are the points aggregated to get 2 clusters if we use the previous policy in agglomerative hierarchical clustering?

- (a) { b, e, c } {d, a} ✓
- (b) { d, b, c } {a, e}
- (c) { a, b, c, d } {e}
- (d) { a, d, b } {c, e}
- (e) {b, e, c, d} {a}
- (f) { b, e, d } {c, a}
- (g) { a, e, c } {b, d}
- (h) No answer is correct

Risposta corretta.

La risposta corretta è: { b, e, c } {d, a}

Domanda 6

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Theory (1 point, -15% penalty for a wrong answer)

The following sequence of operations within a log file is given:

CK(), B(T1), C(T1), B(T3), U3(0), B(T0), C(T0), B(T2), D2(2), CK(T2,T3), U2(2), A(T2), I3(0), C(T3), FAILURE

Notation:

- Tn: Id of transaction n
- B(Tn): Begin of the transaction Tn
- CK(Ta,Tb,…): checkpoint with unfinished transactions Ta, Tb, …
- C(Tn): commit of transaction Tn
- A(Tn): abort (rollback) of transaction Tn
- Un(x): update executed by transaction Tn on object x
- In(x): insert executed by transaction Tn on object x
- Dn(x): delete executed by transaction Tn on object x

What are the final UNDO and REDO sets for warm restart?

- (a) None of the preceding answers
- (b) redo = {3}, undo = {2} ✓
- (c) redo = { }, undo = {2, 3}
- (d) redo = {0}, undo = { }
- (e) redo = {2}, undo = {3}
- (f) redo = {2, 3}, undo = { }
- (g) redo = {3, 0}, undo = {2}
- (h) redo = {3}, undo = {0, 2}

Risposta corretta.

La risposta corretta è: redo = {3}, undo = {2}

Domanda 7

Completo

Non valutata

This question is not a part of the exam

You can use the text area below to write any note or draft (e.g. intermediate steps of an exercise).

Any text written below will not be considered toward the correction of the exam.

1 -> 0

2-> 0

2 -> 1

a, e

b, e

b, e -> a = 13

b, e -> c = 6

b,e, -> d = 17

d 8

a 7

c 6

b 5

e 4

f 2

dacb

dac

dab

dbe

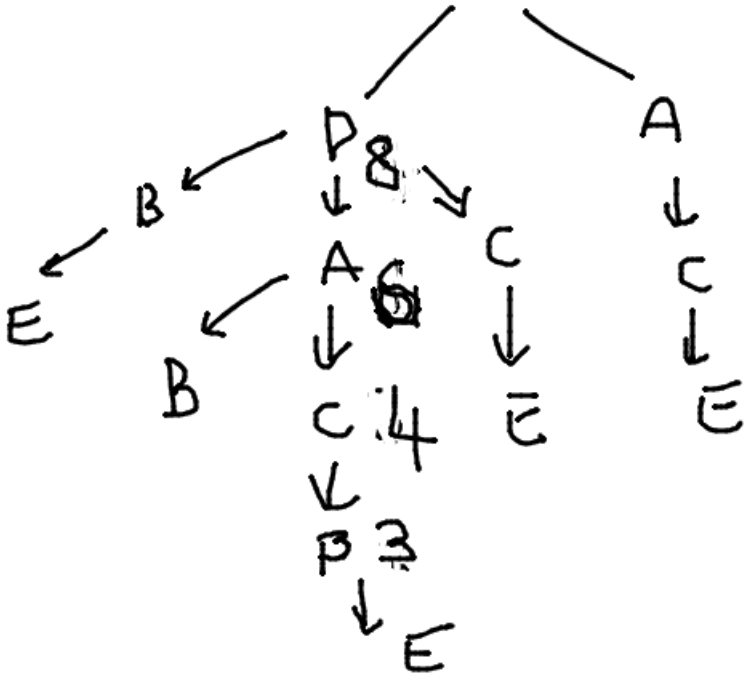
ace

dce

dacbe

da

dacb



Domanda 8

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Theory (1 point, -15% penalty for a wrong answer)

The following document is taken from a MongoDB collection called "catalog," which contains information about products for sale on an E-Commerce site.

```
{
  "productId": "45956",
  "name": "T-Shirt",
  "description": "A comfortable and stylish T-shirt.",
  "price": 29.99,
  "image": "https://example.com/tshirt.jpg",
  "category": "Clothing",
  "stock": 100
}
```

Identifies product categories with an average price higher than 30 and a minimum stock greater than 100 units.

(a)

```
db.collection.aggregate([
 {
  $group: {
    _id: "$category",
    avg_price: {{ $avg: "$price" }: { $gt: 30 } },
    min_stock: {{ $min: "$stock" }: { $gt: 100 }}
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

(b)

```
db.collection.aggregate([
 {
  $match: {
   price: { $gt: 30 },
   stock: { $gt: 100 },
  }
 },
 {
  $group: {
    _id: "$category",
    avg_price: { $avg: "$price" },
    min_stock: { $min: { $sum: "$stock" } }
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

(c)

```
db.collection.aggregate([
 {
  $group: {
    _id: "$category",
    avg_price: { $avg: "$price" },
    min_stock: { $min: "$stock" }
  }
 },
 {
  $match: {
   avg_price: { $gt: 30 },
   min_stock: { $gt: 100 },
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

✓

(d)

```
db.collection.aggregate([
 {
  $match: {
   price: { $gt: 30 },
   stock: { $gt: 100 },
  }
 },
 {
  $group: {
    _id: "$category",
    avg_price: { $avg: "$price" },
    min_stock: { $min: "$stock" }
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

(e)

```
db.collection.aggregate([
 {
  $group: {
    _id: "$category",
    avg_price: { $avg: "$price" },
    min_stock: { $min: { $sum: "$stock" } }
  }
 },
 {
  $match: {
   avg_price: { $gt: 30 },
   min_stock: { $gt: 100 },
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

(f) None of the solutions are correct

Risposta corretta.

La risposta corretta è:

```
db.collection.aggregate([
 {
  $group: {
    _id: "$category",
    avg_price: { $avg: "$price" },
    min_stock: { $min: "$stock" }
  }
 },
 {
  $match: {
   avg_price: { $gt: 30 },
   min_stock: { $gt: 100 },
  }
 },
 {
  $project: {
   _id: 1,
  }
 },
])
```

Domanda 9

Completo

Punteggio ottenuto 4,75 su 5,00

5 points (no penalty for a wrong answer)

The following data warehouse stores information on patient accesses (characterized by age range and gender) to the emergency room of hospitals located in Italy. Each hospital belongs to a single type (public or private) and there may be one or more departments among pediatrics, geriatrics, orthopedics, and cardiology (these attributes are Boolean and represent a configuration). In the Triage a color code (among white, green, blue, yellow, and red) is assigned and the mode of access to the emergency room (by ambulance or independently) is tracked. When someone accesses the emergency room, a diagnosis characterized by one or more specialistic examinations is assigned.

The data warehouse is characterized by the following conceptual schema and corresponding logical schema. Each event is characterized by four measures: number of patients, number of visits, number of physicians who made the visits and hours spent in the emergency room (TotalHours).

PATIENT_CHARACTERISTICS (CodCP, AgeRange, Gender)
TRIAGE (CodTriage, AccessMode, ColorCode)
HOSPITAL (CodO, Hospital, Typology, City, Province, Region, Pediatrics, Geriatrics, Orthopedics, Cardiology)
DIAGNOSIS (CodD, Diagnosis, Illness)
EXAM-DIAGNOSIS (CodD, SpecialisticExamination)
TIME (CodT, Date, DayOfWeek, Week, DayOfMonth, Holiday, Month, MonthOfYear, 4-Months, 6-Months, Year)
EMERGENCY-ROOM-ACCESS (CodCP, CodTriage, CodO, CodD, CodT, #Patients, #Visits, #Physicians, TotalHours)

Given the above logical schema, consider the following queries of interest:

1. Considering patients in the '>70' age range, separately by province and year, show the total hours spent in the emergency room, the average hours spent in the emergency room per patient, and the average monthly number of patients.
2. Considering female patients and hospitals in the Lombardy region, show the cumulative annual value of the number of visits for increasing four-months period (attribute 4-Months)
3. Considering hospitals with pediatrics and surgery departments, separately by region and semester (attribute 6-Months), show the total number of visits and the average number of visits per patient.

Given the above logical schema, answer the following requests:

1. Define a materialized view with CREATE MATERIALIZED VIEW, in order to reduce the response time of the queries of interest from (a) to (c) above. Specifically, specify the SQL query associated with Block A in the following statement:

```
CREATE MATERIALIZED VIEW ViewAccess
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
 Block A
```

2. Define the **minimal set** of attributes that allow to identify the tuples that belong to the ViewAccess materialized view.

3. Assume that the management of the materialized view (derived table) is carried out by means of triggers. Write the trigger to propagate changes to the ViewAccess materialized view when a new record is inserted into the EMERGENCY-ROOM-ACCESS fact table.

Block A
SELECT AgeRange, Province, Year,
SUM(TotalHours) AS TotHours,
SUM(#Patients) AS TotPatients,
Month, Gender, Region, 4-Months,
SUM(#Visits) AS TotVisits,
Pediatrics, Surgery, 6-Months
FROM OSPITAL o, TIME t, PATIENT_CHARACTERISTICS pc, EMERGENCY-ROOM-ACCESS era
WHERE o.CodO == era.CodO AND t.CodT == era.CodT AND era.CodCP == pc.CodCP
GROUP BY Year, 6-Months, 4-Months, Month, AgeRange, Province, Region, Gender, Pediatrics, Surgery

2) Month, AgeRange, Province, Gender, Pediatrics, Surgery

3)
CREATE OR REPLACE TRIGGER ViewAccessTrigger
AFTER INSERT ON EMERGENCY-ROOM-ACCESS
FOR EACH ROW

DECLARE
N NUMBER;
VarAgeRange, VarProvince, VarGender, VarRegion VARCHAR(10);
VarYear, VarMonth, Var4-Months, Var6-Months DATE;
VarPediatrics, VarSurgery BOOLEAN;

```
BEGIN
SELECT AgeRange, Gender INTO VarAgeRange, VarGender
FROM PATIENT_CHARACTERISTICS
WHERE CodCP == :NEW.CodCP;

SELECT Region, Province, Pediatrics, Surgery INTO VarRegion, VarProvince, VarPediatrics,
VarSurgery
FROM OSPITAL
WHERE CodO == :NEW.CodO;

SELECT Year, 6-Months, 4-Months, Month INTO VarYear, Var6-Months, Var4-Months, VarMonth
FROM TIME
WHERE CodT == :NEW.CodT;

SELECT COUNT(*) INTO N
WHERE VarMonth == :NEW.Month AND VarAgeRange == :NEW.AgeRange AND VarProvince ==
:NEW.Province AND VarGender == :NEW.Gender AND VarPediatrics == :NEW.Pediatrics AND
VarSurgery == :NEW.Surgery;

IF (N == 0) THEN
  INSERT INTO ViewAccess
  (AgeRange, Province, Gender, Region, YEar, Month, 4-Months, 6-Months, Pediatrics, Surgery,
TotHours, TotPatients, TotVisits)
  VALUES
  (VarAgeRange, VarProvince, VarGender, VarRegion, VarYear, VarMonth, Var4-Months, Var6-
Months, VarPediatrics, VarSurgery, SUM(:NEW.TotalHours), SUM(:NEW.#Patients),
SUM(:NEW.#Visits))
ELSE
  UPDATE ViewAccess
  SET TotHours = TotHours + :NEW.TotalHours, TotPatients = TotPatients + :NEW.#Patients,
TotVisits = TotVisits + :NEW.#Visits
ENDIF;
END;
```

Queries of interest:

(a)

```
SELECT Province, Year, SUM(TotalHours), SUM(TotalHours)/SUM(#Patients),
SUM(#Patients)/COUNT(DISTINCT Month)

FROM PATIENT_CHARACTERISTICS P, HOSPITAL H, TIME T, EMERGENCY-ROOM-
ACCESS ERA

WHERE ERA.CodCP=P.CodCP AND ERA.CodO = H.CodO AND ERA.CodT = T.CodT

AND AgeRange = '>70'

GROUP BY Province, Year
```

(b)

SELECT 4-Months, Year, SUM(SUM(#Visits)) OVER (PARTITION BY Year

ORDER BY 4-Months

ROWS UNBOUNDED PRECEDING)

FROM PATIENT_CHARACTERISTICS P, HOSPITAL H, TIME T, EMERGENCY-ROOM-ACCESS ERA

WHERE ERA.CodCP=P.CodCP AND ERA.CodO = H.CodO AND ERA.CodT = T.CodT

AND Gender = 'F' AND Region = 'Lombardia'

GROUP BY 4-Mesi, Year

(c)

Select Region, 6-Months, SUM(#Visits), SUM(#Visits)/SUM(#Patients)

FROM PATIENT_CHARACTERISTICS P, HOSPITAL H, TIME T, EMERGENCY-ROOM-ACCESS ERA

WHERE ERA.CodCP=P.CodCP AND ERA.CodO = H.CodO AND ERA.CodT = T.CodT

AND Pediatrics = 1 AND Cardiology = 1

GROUP BY Region, 6-Months

1. Query for materialized view

SELECT Month, 4-Months, 6-Months, Year, Province, Region, Pediatrics, Cardiology, AgeRange, Gender, SUM(#Visits) AS TOTVisits, SUM(TotalHours) AS TOTHours, SUM(# Patients) AS TOTPatients

FROM HOSPITAL H, TIME T, PATIENT_CHARACTERISTICS P, EMERGENCY-ROOM-ACCESS ERA

WHERE H.CodO = ERA.CodO AND T.CodT = ERA.CodT AND P.CodCP = ERA.CodCP

GROUP BY Month, 4-Months, 6-Months, Year, Province, Region, AgeRange, Gender, Pediatrics, Cardiology

2. Identifier

Month, Province, AgeRange, Gender, Pediatrics, Cardiology

3. Trigger

CREATE OR REPLACE TRIGGER ViewAccess

AFTER INSERT ON EMERGENCY-ROOM-ACCESS

FOR EACH ROW

DECLARE

```
VarMese, VarAnno, Var4Mesi, Var6Mesi DATE;
VarProvincia, VarRegione, VarFascia, VarGenere  VARCHAR(10);
VarPed, VarCard BOOLEAN;
N INTEGER;
BEGIN
 SELECT Month, 6-Months, Year INTO varMonth, var6-Months, varYear
 FROM TIME
 WHERE CodT = :NEW.CodT;

 SELECT AgeRange, Gender INTO varAgeRange, varGender
 FROM PATIENT_CHARACTERISTICS
 WHERE CodCP = :NEW.CodCP;

 SELECT Province, Region, Pediatrics, Cardiology INTO varProvince, varRegion, varPediatrics,
varCardiology
 FROM HOSPITAL
 WHERE CodO = :NEW.CodO;

 SELECT COUNT(*) INTO N
 FROM ViewAccess
 WHERE Month = varMonth AND Province = varProvince AND Pediatrics = varPediatrics AND
Cardiology = varCardiology AND     AgeRange = varAgeRange AND Gender = varGender;

IF N>0 THEN
 UPDATE ViewAccess
 SET TOTVisits = TOTVisits + :NEW.#Visits, TOTHours = TOTHours + :NEW.TotalHours,
                          TOTPatients = TOTPatients + :NEW.#Pazients
 WHERE Month = varMonth AND Province = varProvince AND Pediatrics = varPediatrics AND
Cardiology = varCardiology AND     AgeRange = varAgeRange AND Gender = varGender;
ELSE
 INSERT INTO ViewAccess(…) VALUES (varMonth, var6-Months, varYear, varProvince,
varRegione, varPediatrics, varCardiology,   varAgeRange , varGender, :NEW.#Visits,
:NEW.TotalHours, :NEW.#Pazients);
END IF;
END;
```

Commento:
Block A

SELECT AgeRange, Province, Year,

SUM(TotalHours) AS TotHours,

SUM(#Patients) AS TotPatients,

Month, Gender, Region, 4-Months,

SUM(#Visits) AS TotVisits,

Pediatrics, Surgery, 6-Months

FROM OSPITAL o, TIME t, PATIENT_CHARACTERISTICS pc, EMERGENCY-ROOM-ACCESS era

WHERE o.CodO == era.CodO AND t.CodT == era.CodT AND era.CodCP == pc.CodCP

GROUP BY Year, 6-Months, 4-Months, Month, AgeRange, Province, Region, Gender, Pediatrics, Surgery

2) Month, AgeRange, Province, Gender, Pediatrics, Surgery

3)

CREATE OR REPLACE TRIGGER ViewAccessTrigger

AFTER INSERT ON EMERGENCY-ROOM-ACCESS

FOR EACH ROW

DECLARE

N NUMBER;

VarAgeRange, VarProvince, VarGender, VarRegion VARCHAR(10);

VarYear, VarMonth, Var4-Months, Var6-Months DATE;

VarPediatrics, VarSurgery BOOLEAN;

BEGIN

SELECT AgeRange, Gender INTO VarAgeRange, VarGender

FROM PATIENT_CHARACTERISTICS

WHERE CodCP == :NEW.CodCP;

SELECT Region, Province, Pediatrics, Surgery INTO VarRegion, VarProvince, VarPediatrics, VarSurgery

FROM OSPITAL

WHERE CodO == :NEW.CodO;

SELECT Year, 6-Months, 4-Months, Month INTO VarYear, Var6-Months, Var4-Months, VarMonth

FROM TIME

WHERE CodT == :NEW.CodT;

SELECT COUNT(*) INTO N

WHERE VarMonth == :NEW.Month AND VarAgeRange == :NEW.AgeRange AND VarProvince == :NEW.Province AND VarGender == :NEW.Gender AND VarPediatrics == :NEW.Pediatrics AND VarSurgery == :NEW.Surgery;

IF (N == 0) THEN

INSERT INTO ViewAccess

(AgeRange, Province, Gender, Region, YEar, Month, 4-Months, 6-Months, Pediatrics, Surgery, TotHours, TotPatients, TotVisits)

VALUES

(VarAgeRange, VarProvince, VarGender, VarRegion, VarYear, VarMonth, Var4-Months, Var6-Months, VarPediatrics, VarSurgery, SUM(:NEW.TotalHours), SUM(:NEW.#Patients), SUM(:NEW.#Visits))

ELSE

UPDATE ViewAccess

SET TotHours = TotHours + :NEW.TotalHours, TotPatients = TotPatients + :NEW.#Patients, TotVisits = TotVisits + :NEW.#Visits

WHERE CLAUSE is missing

ENDIF;

END;

Domanda 10

Risposta corretta

Punteggio ottenuto 3,00 su 3,00

3 total points (penalty 15% for each wrong answer)

An oil company wants to analyze information on refueling performed at its service stations on Italian territory. In particular, the company is interested in analyzing fueling information from customers who have a loyalty card with the oil company.

Each service station, in addition to the refueling service (petrol, diesel, LPG), can offer different types of additional services, such as car cleaning, minor mechanical interventions, and oil and tire checks.

Company managers want to analyze the average number of liters of fuel purchased by customers and the average time customers spend at stations based on the following information:

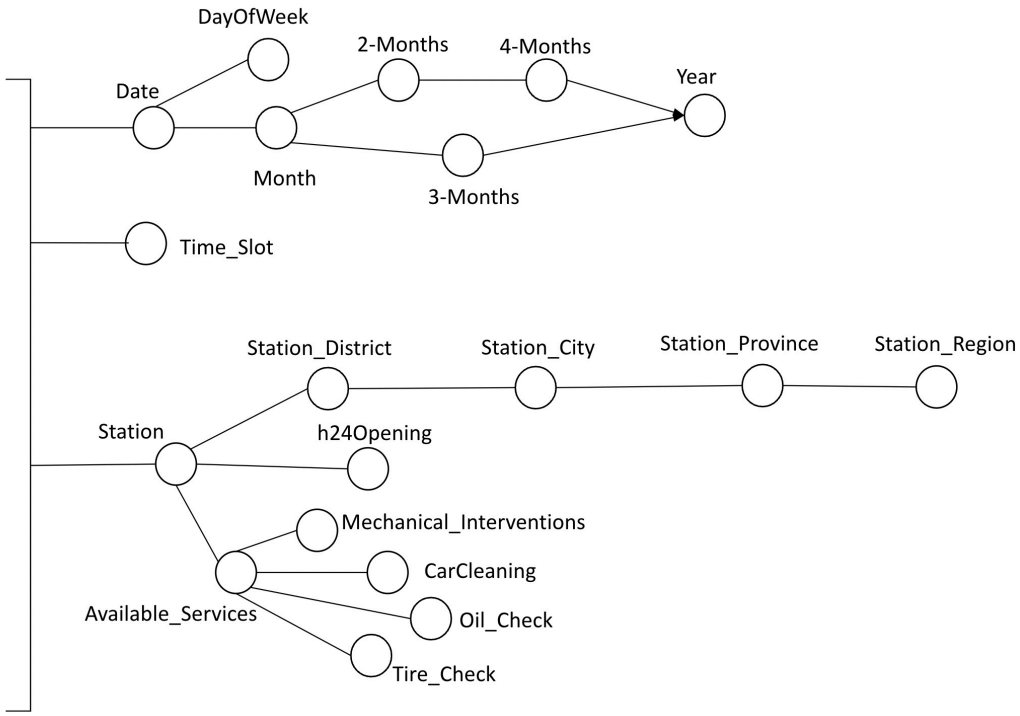
- the service station and its geographical location in terms of city district, city, province, and region. For each station, it is known whether it is a station with h24 opening hours or not. The services available for each station are also known (one or more values among "car cleaning," "mechanical interventions," "oil check," and "tire check") .
- time slot (one value among morning, afternoon, evening), date, day of the week, month, two-month period, three-month period, four-month period, and year in which the customer has refueled at the station.
- type of customer's loyalty card (a value between "basic," "gold," "privilege") and the list of all benefits associated with the type of loyalty card
- city, province, and region of the customer who has refueled
- type of fuel purchased by the customer during refueling (a value between "gasoline", "diesel", "LPG")
- refueling mode (a value between "served" and "self-service")
- type of payment (a value between "cash," "ATM," "credit card")

CONCEPTUAL SCHEMA 1

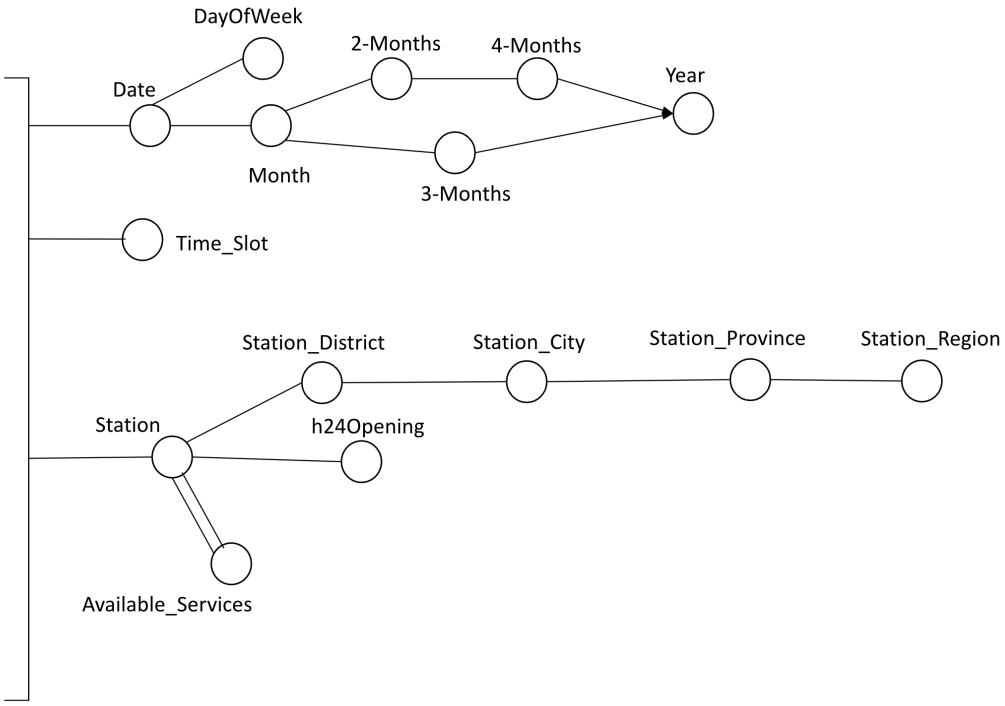
1 point (penalty 15% for wrong answer)

Select, among the proposed dimensions below, those that meet the requirements described in the problem specifications.

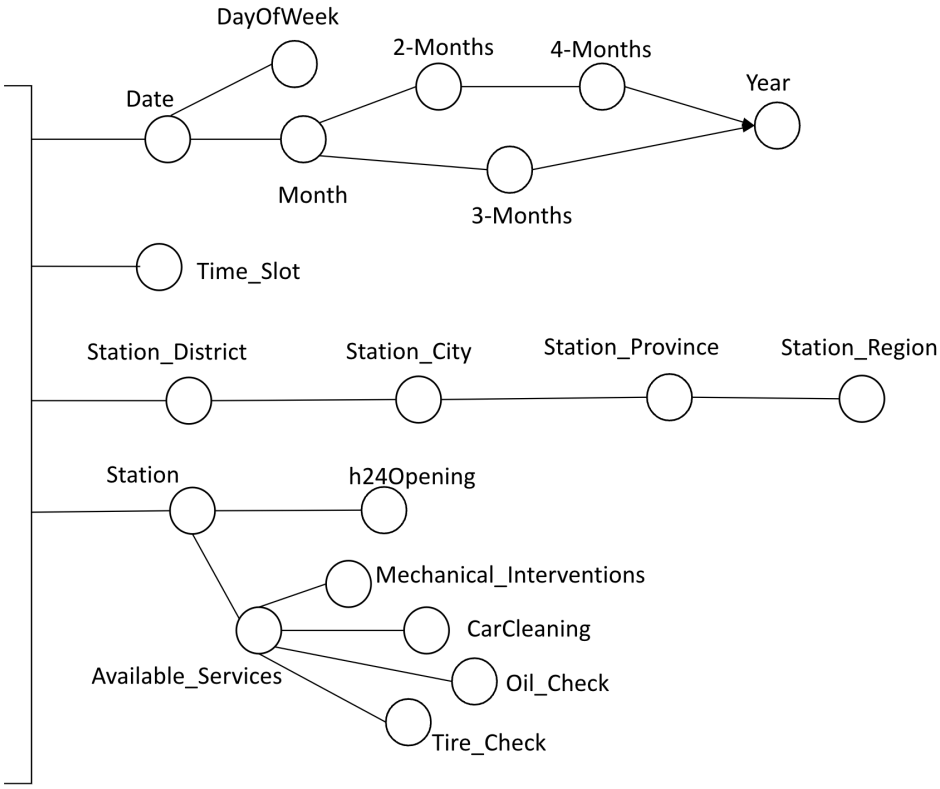
a)


DayOfWeek
2-Months
4-Months
Year
Date
Month
3-Months
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Mechanical_Interventions
CarCleaning
Available_Services
Oil_Check
Tire_Check

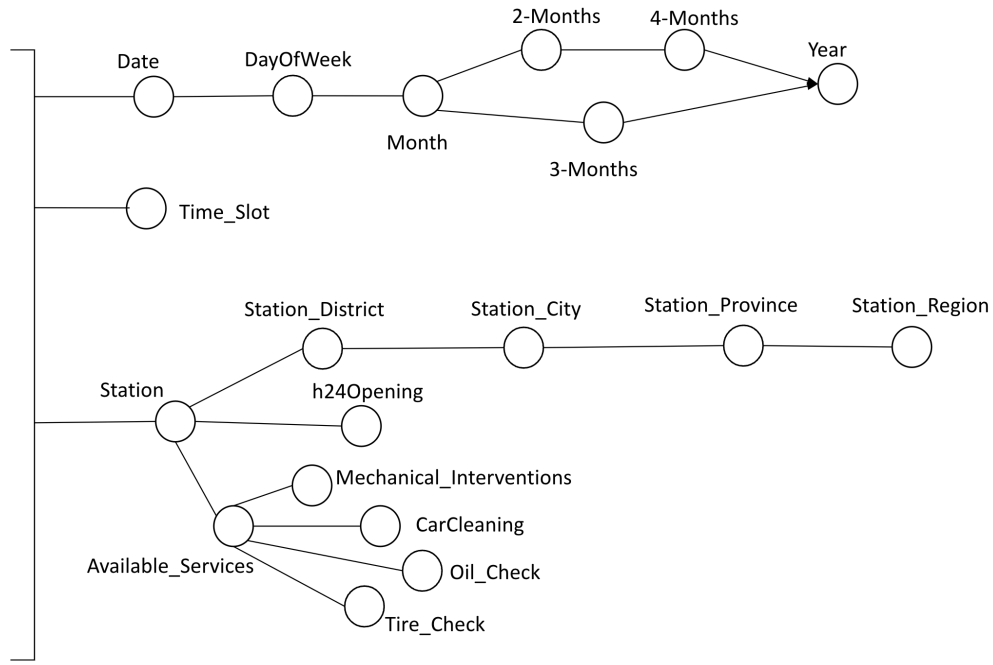
b)


DayOfWeek
2-Months
4-Months
Year
Date
Month
3-Months
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Available_Services

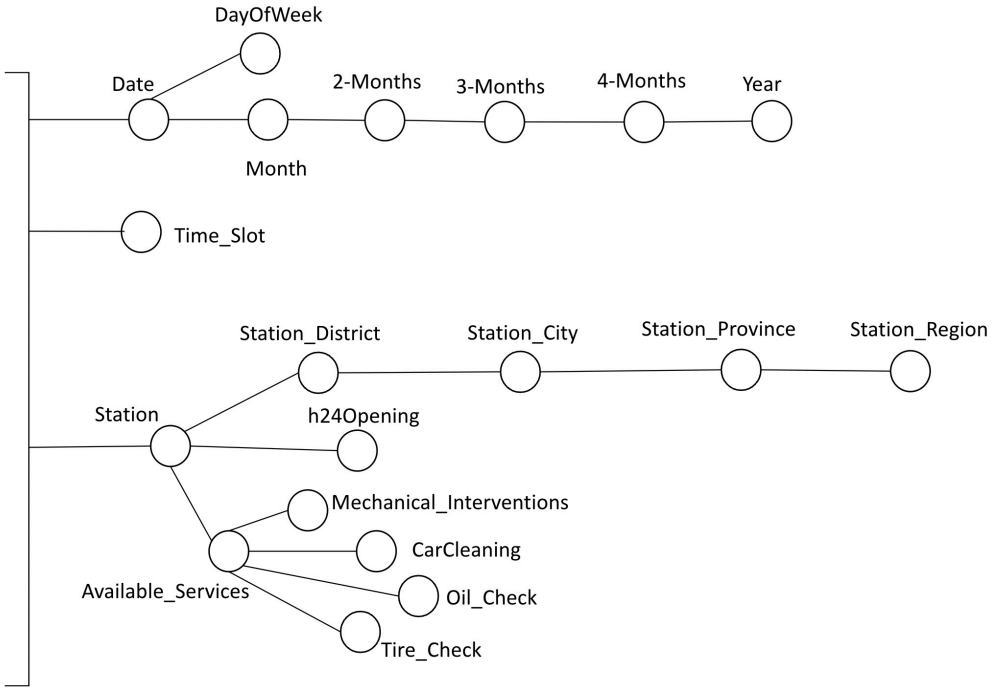
c)


DayOfWeek
2-Months
4-Months
Year
Date
Month
3-Months
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Mechanical_Interventions
CarCleaning
Available_Services
Oil_Check
Tire_Check

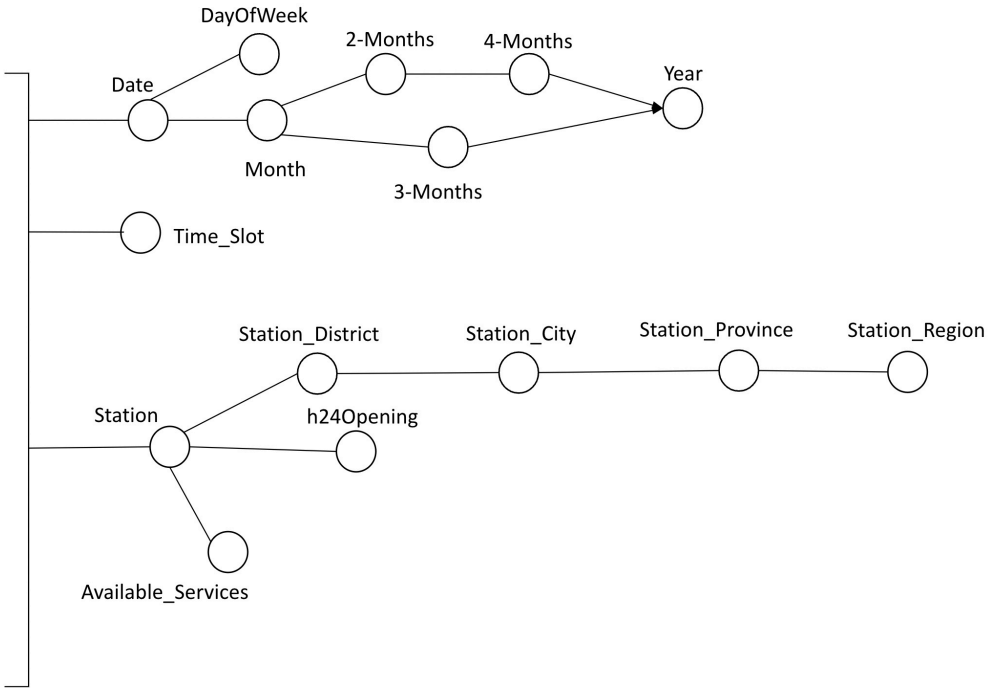
d)


2-Months
4-Months
Year
Date
DayOfWeek
Month
3-Months
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Mechanical_Interventions
CarCleaning
Available_Services
Oil_Check
Tire_Check

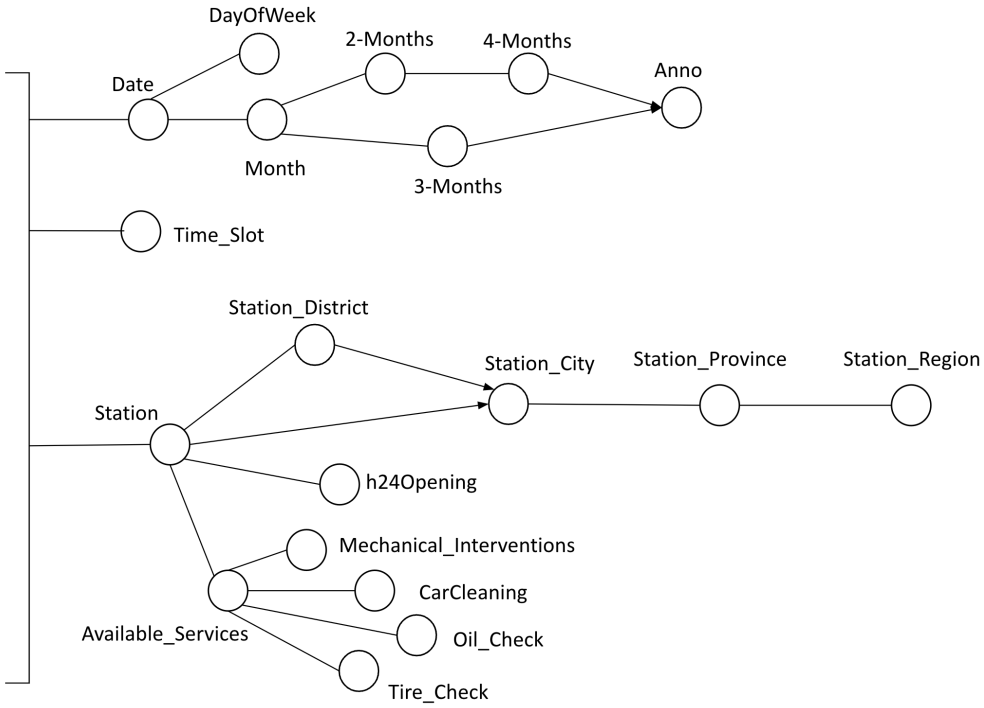
e)


DayOfWeek
Date
2-Months
3-Months
4-Months
Year
Month
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Mechanical_Interventions
CarCleaning
Available_Services
Oil_Check
Tire_Check

f)


DayOfWeek
2-Months
4-Months
Date
Year
Month
3-Months
Time_Slot
Station_District
Station_City
Station_Province
Station_Region
Station
h24Opening
Available_Services

g)



d a b e f g c

Punteggio ottenuto 1,00 su 1,00

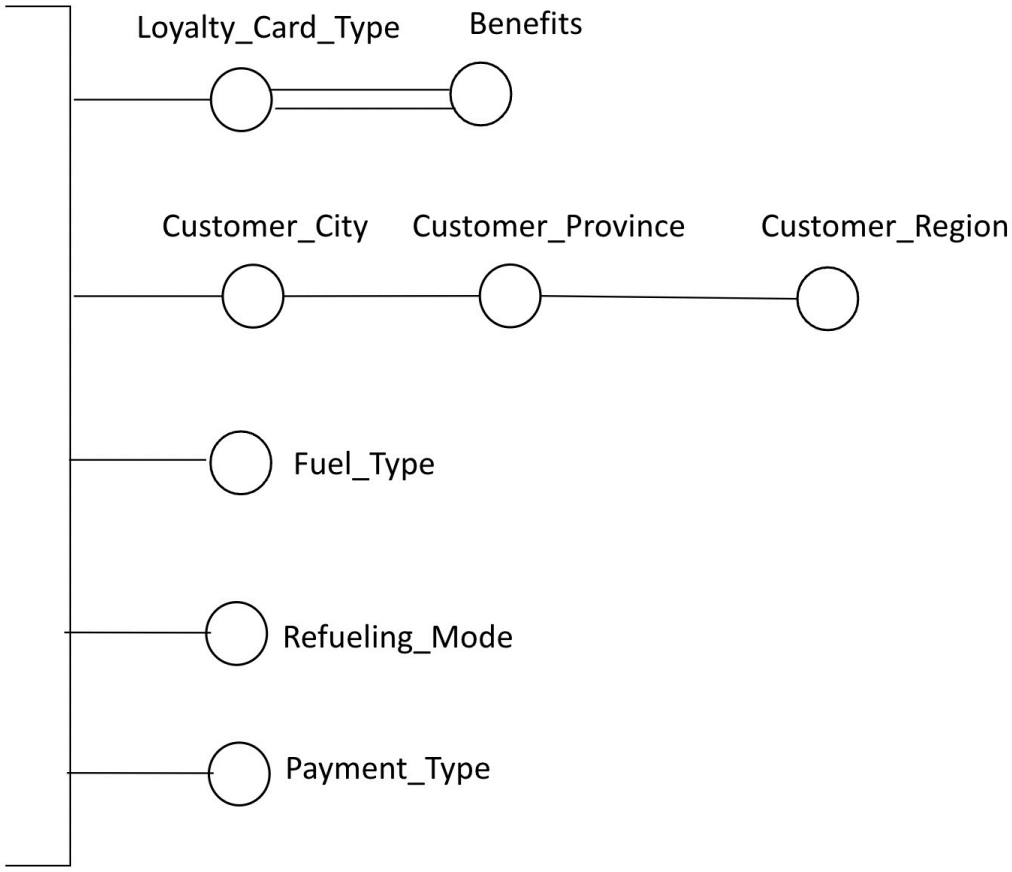
La risposta corretta è: a

CONCEPTUAL SCHEMA 2

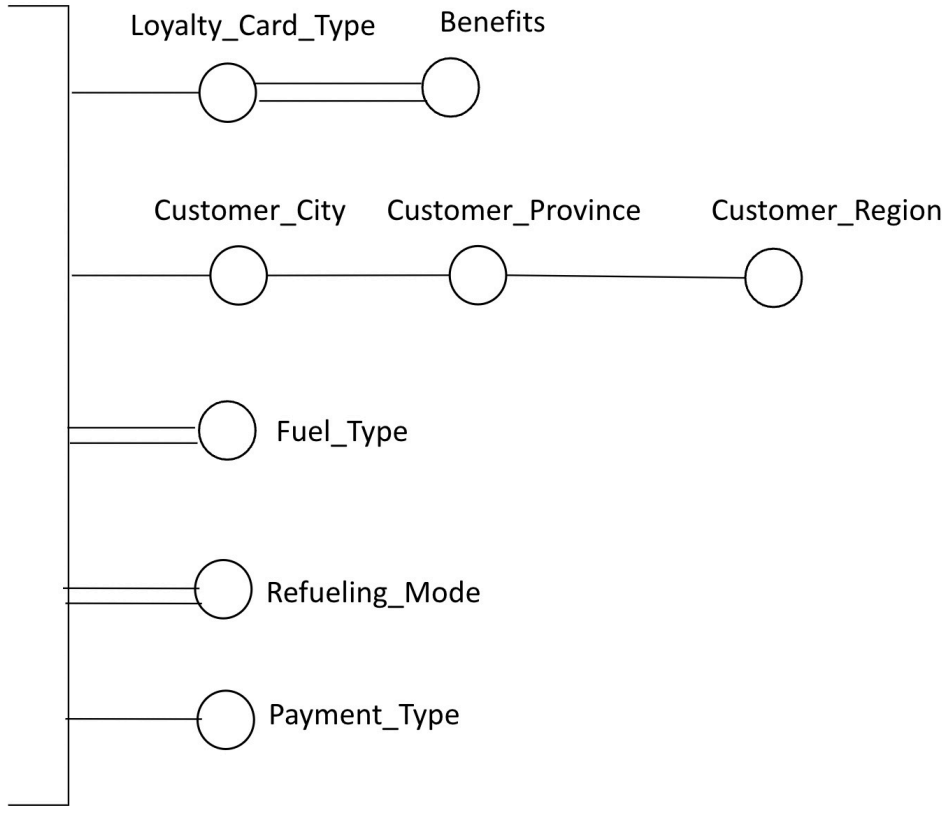
1 point (penalty 15% for wrong answer)

Select, among the proposed dimensions below, those that meet the requirements described in the problem specifications.

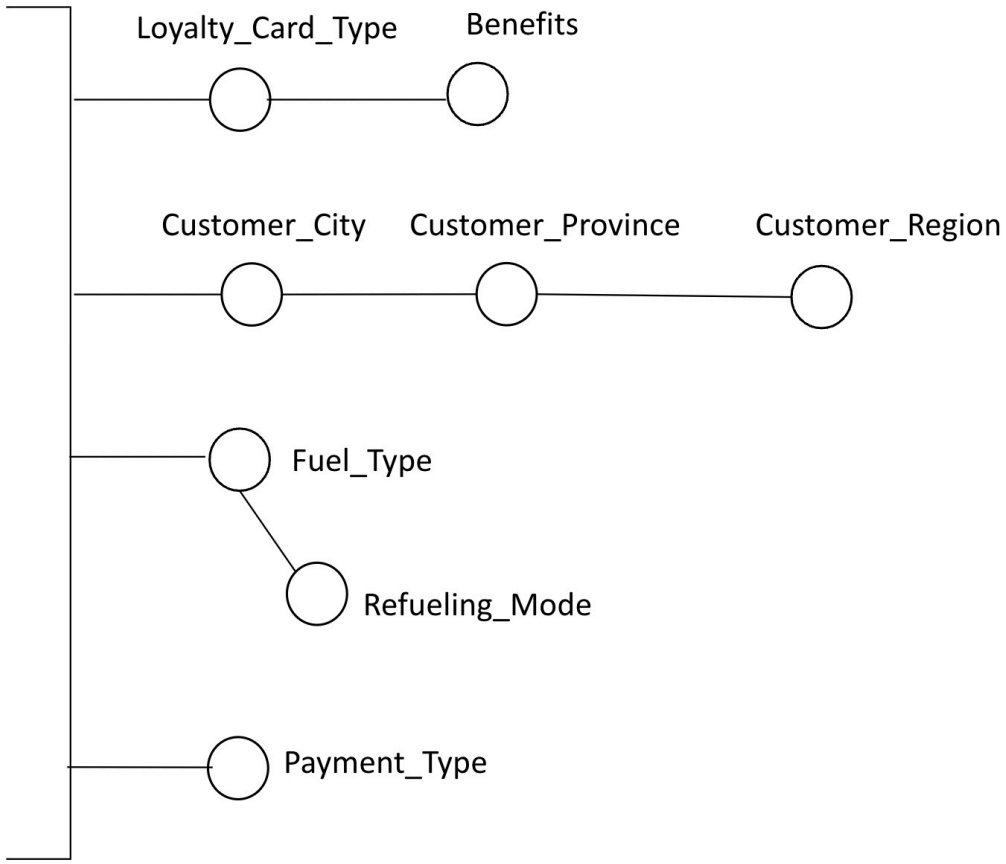
a)


Loyalty_Card_Type
Benefits
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

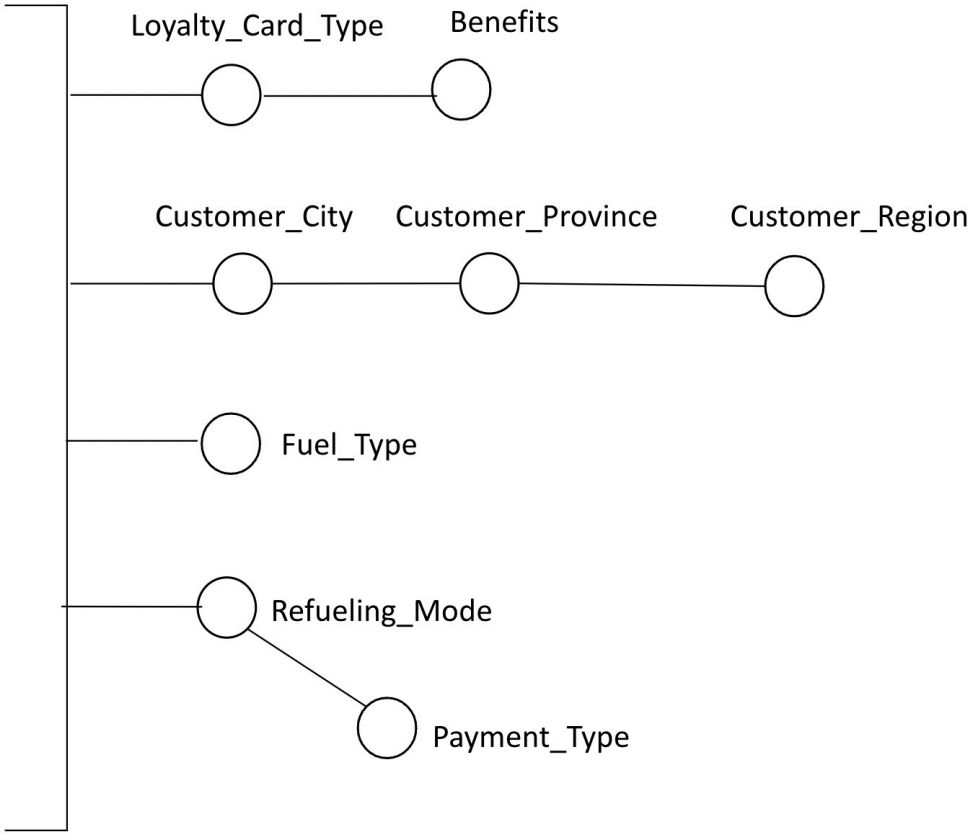
b)


Loyalty_Card_Type
Benefits
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

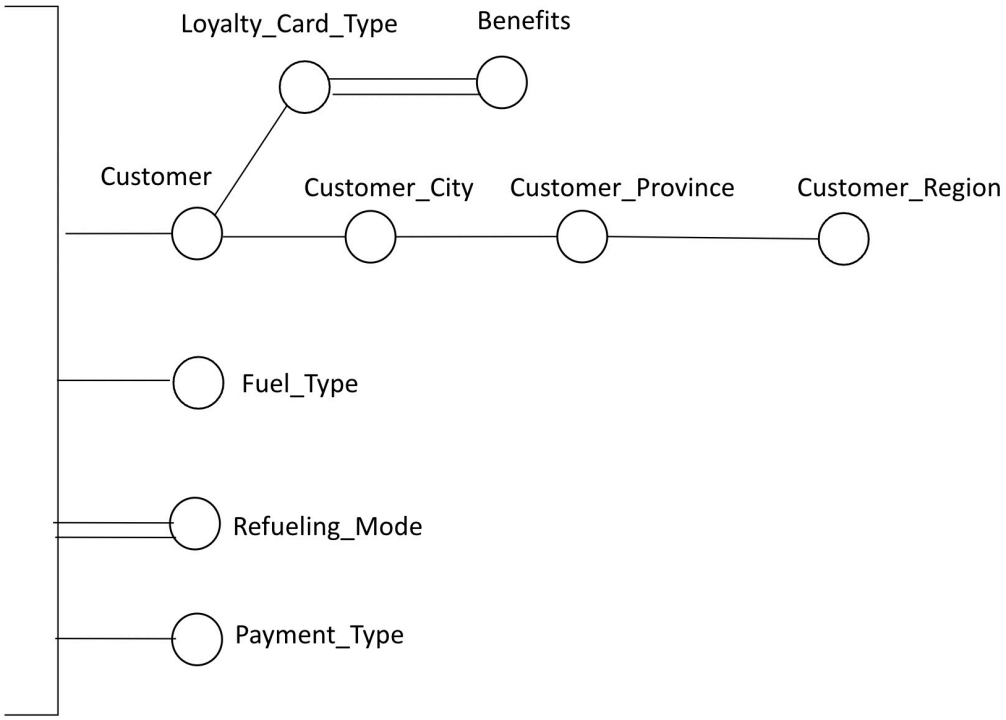
c)


Loyalty_Card_Type
Benefits
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

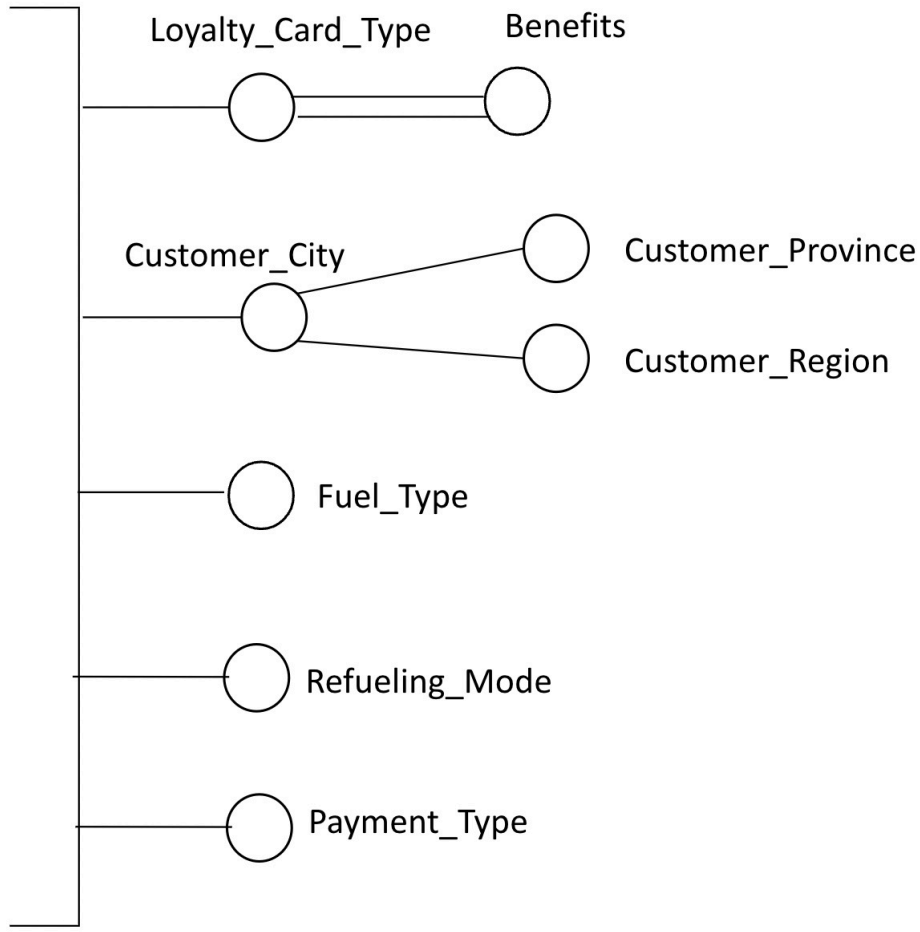
d)


Loyalty_Card_Type
Benefits
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

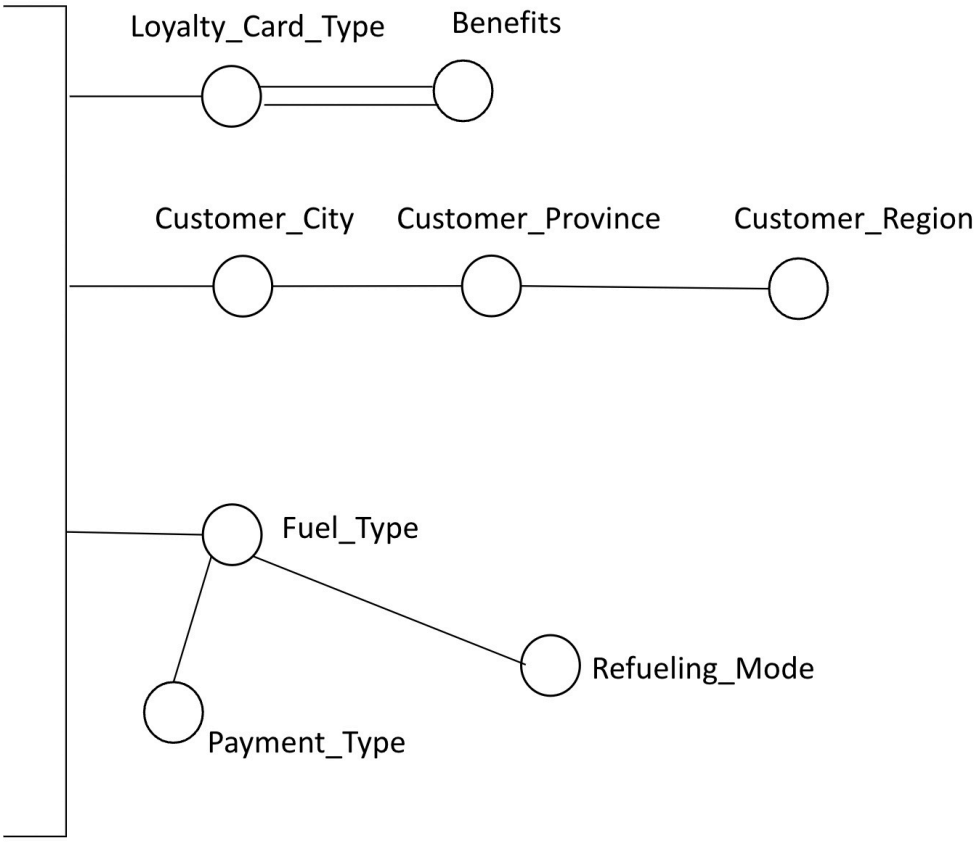
e)


Loyalty_Card_Type
Benefits
Customer
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

f)


Loyalty_Card_Type
Benefits
Customer_City
Customer_Province
Customer_Region
Fuel_Type
Refueling_Mode
Payment_Type

g)



d a g f b e c

Punteggio ottenuto 1,00 su 1,00

La risposta corretta è: a

MEASURES

1 point (penalty 15% for wrong answer)

Select the set of attributes necessary to correctly model the measures in the fact table as required by the specifications.

- Total time spent at the station, Total number of refueling operations performed, Maximum number of liters of fuel purchased
- Total time spent at the station, Average number of refueling operations performed, Maximum number of liters of fuel purchased
- Total time spent at the station, Average number of refueling operations performed, Total number of liters of fuel purchased
- Average time spent at the station, Average number of refueling operations performed, Average number of liters of fuel purchased

Average time spent at the station, Average number of refueling operations performed, Total number of liters of fuel purchased

Total time spent at the station, Total number of refueling operations performed, Average number of liters of fuel purchased

Average time spent at the station, Total number of refueling operations performed, Total number of liters of fuel purchased

Total time spent at the station, Total number of refueling operations performed, Total number of liters of fuel purchased✓

Punteggio ottenuto 1,00 su 1,00

La risposta corretta è: Total time spent at the station, Total number of refueling operations performed, Total number of liters of fuel purchased

1) La risposta corretta è : a
2) La risposta corretta è : a
3) La risposta corretta è : Total time spent at the station, Total number of refueling operations performed, Total number of liters of fuel purchased

Domanda 11

Risposta corretta

Punteggio ottenuto 1,50 su 1,50

Theory (1.5 points, -15% penalty for a wrong answer)

Create an FP-Tree for extracting frequent itemsets with MinSup > 3 (an itemset is frequent if it appears in more than 3 transactions) for the following list of transactions. Which of the following is a node path present in the FP-tree?

Transactions:

- {a,b,c,d}
- {a,c,d}
- {a,b,d,f}
- {b,d,e}
- {a,c,e}
- {c,d,e}
- {a,b,c,d,e}
- {a,d,f}
- {a,b,c,d}

- (a) No answer is correct
- (b) d:8 → a:6 → f:1
- (c) d:8 → a:6 →b:1 → f:1
- (d) a:2 → c:1 →e1
- (e) d:8 → b:2 → e:1
- (f) d:8 → a:6 → c:3
- (g) d:8 → c:2 → e:2
- (h) d:8 → c:1 → e:1 ✓

Risposta corretta.

La risposta corretta è: d:8 → c:1 → e:1