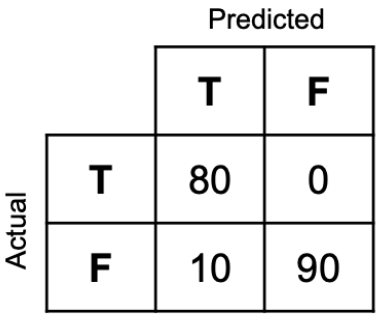
Data Science and Database Technology

Exam 2020-05-09

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

Given the confusion matrix depicted in figure, which statement is correct?



(a) Recall of class **F** is **90/100**, accuracy is **90/180**

(b) None of the other statements is correct

(c) Precision of class **F** is **90/100**, recall of class **T** is **80/90**

(d) Precision of class **T** is **80/90**, recall of class **F** is **90/100** ✔

(e) Precision of class **F** is **90/100**, accuracy is **170/180**

(f) Precision of class **T** is **80/90**, recall of class **F** is **10/100**

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

In agglomerative hierarchical clustering the **MAX** metric (or complete linkage) implies that:

(a) None of the answers is correct ✔

(b) A cluster **C** is merged with a single point **p** if the maximum distance between **p** and the points in **C** is the maximum in the distance matrix

(c) Two clusters $\mathbf{C_1}$, $\mathbf{C_2}$ are merged if there exist a pair of points $\mathbf{p_1} \in \mathbf{C_1}$, $\mathbf{p_2} \in \mathbf{C_2}$ whose distance is the maximum in the distance matrix

(d) A cluster **C** is merged with a single point **p** if the distance between **p** and **C** is the maximum in the distance matrix

(e) The obtained clusters are very sensitive to noise

(f) The first two points that are merged in the dendrogram are the ones that are the farthest from each other

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

According to the definitions of hierarchical locking for concurrency control in a DBMS:

(a) If the state of a node is **Intention of Exclusive Lock** (IXL), you cannot request an **Intention of Shared Lock** (ISL) on the same node

(b) You can request an **Intention of Exclusive Lock** (IXL) on a child node after obtaining a **Shared Lock** (SL) on its parent node

(c) To request an Intention of **Shared Lock** (ISL) it is not necessary to request permissions on the parent node

(d) If the state of a node is **Intention of Shared Lock** (ISL), you can request an **Intention of Exclusive Lock** (IXL) on the same node ✔

(e) If the state of a node is **Intention of Exclusive Lock** (IXL), you can request a **Shared Lock** on the same node

(f) To request an **Exclusive Lock** (XL) on a child node it is necessary to request an **Exclusive Lock** (XL) on its parent node

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

Give the following sequence of operations in a log file:

$B(T_1)$ $I_1(o_1)$ $B(T_2)$ $CK(T_1,T_2)$ $B(T_3)$ $U_3(o_2)$ $D_1(o_4)$ $Commit(T_1)$ $U_2(o_1)$ $Abort(T_2)$ $I_3(o_5)$
FAILURE

Notation:

- T_n = Id of transaction n
- $B(T_n)$ = Begin of T_n
- CK = checkpoint

- $U_n(o_x)$ = update executed by T_n on object o_x; same notation for I (insert) and D (delete)

Which operations are executed for a **warm restart**?

(a) Redo of $\mathbf{T_1}$, $\mathbf{T_2}$, undo of $\mathbf{T_3}$

(b) Redo of $\mathbf{T_2}$, $\mathbf{T_3}$, undo of $\mathbf{T_1}$

(c) Redo of $\mathbf{T_3}$, undo of $\mathbf{T_1}$, $\mathbf{T_2}$

(d) Redo of $\mathbf{T_1}$, $\mathbf{T_3}$, undo of $\mathbf{T_2}$

(e) Redo of $\mathbf{T_1}$, undo of $\mathbf{T_2}$, $\mathbf{T_3}$ ✔

(f) Redo of $\mathbf{T_2}$, undo of $\mathbf{T_1}$, $\mathbf{T_3}$

Cardinalities (2 points, -15% penalty for a wrong answer)

SMART-DEVICE(SerialId, Name, Brand, Type, Price)
FEATURE(FeatureId, Name, Category)
DEVICE-HAS-FEATURE(SerialId, FeatureId, Value, UnitOfMeasurement, DisplayValue)
USER(Username, FirstName, LastName, BirthDate, Address, City, Country)
PURCHASE(Timestamp, Username, SerialId, TotalCost, NumberOfItems)

Assume the following cardinalities:

- card(SMART-DEVICE) = $5 \cdot 10^6$ tuples
- distinct values of Type = 50
- card(FEATURE) = 10^2 tuples
- distinct values of Category = 10
- card(DEVICE-HAS-FEATURE) = 10^8 tuples
- card(USER) = 10^6 tuples
- distinct values of Country = 100
- card(PURCHASE) = $5 \cdot 10^7$ tuples
- MIN(DATE(Timestamp)) = 1/1/2018, MAX(DATE(Timestamp)) = 31/12/2019

Furthermore, assume the following reduction factor for the group by condition:

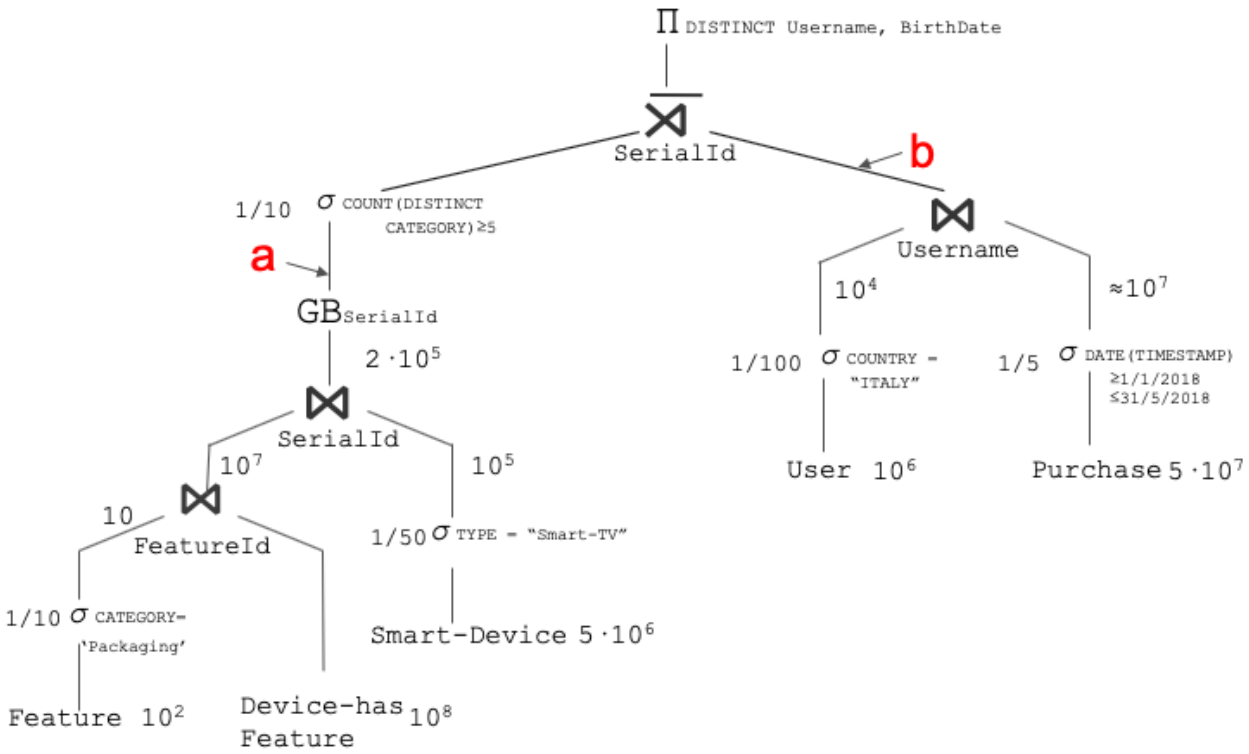
HAVING COUNT(DISTINCT F.Category) >= 5 ≈ 1/10

Consider the following query

```
select DISTINCT Username, BirthDate
from USER U, PURCHASE P
where U.Username=P.Username and Country='Italy'
and DATE(Timestamp) >= 1/1/2018 and DATE(Timestamp) <= 31/5/2018
and SerialId NOT IN (select SerialId
                     from SMART-DEVICE D, FEATURE F, DEVICE-HAS-FEATURE DHF
                     where D.SerialId=DHF.SerialId and F.FeatureId=DHF.FeatureId
                     and F.Category='Packaging'
                     D.Type = 'SmartTV'
                     group by SerialId
                     having COUNT(DISTINCT F.Category)>=5)
```

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

Specify the cardinality of each branch indicated by the red letters (**a**, **b**) in the figure below.



(a) a: 10^5, b: $5 \cdot 10^5$

(b) a: 10^5, b: 10^5 ✔

(c) a: $5 \cdot 10^6$, b: 10^5

(d) a: $5 \cdot 10^5$, b: $5 \cdot 10^5$

(e) a: $5 \cdot 10^6$, b: $2 \cdot 10^3$

(f) a: $5 \cdot 10^5$, b: 10^5

(g) a: 10^5, b: $2 \cdot 10^3$

Group by anticipation (2 points, -15% penalty for a wrong answer)

SMART-DEVICE(SerialId, Name, Brand, Type, Price)
FEATURE(FeatureId, Name, Category)
DEVICE-HAS-FEATURE(SerialId, FeatureId, Value, UnitOfMeasurement, DisplayValue)
USER(Username, FirstName, LastName, BirthDate, Address, City, Country)
PURCHASE(Timestamp, Username, SerialId, TotalCost, NumberOfItems)

Assume the following cardinalities:

- card(SMART-DEVICE) = $5 \cdot 10^6$ tuples
- distinct values of Type = 50
- card(FEATURE) = 10^2 tuples
- distinct values of Category = 10
- card(DEVICE-HAS-FEATURE) = 10^8 tuples
- card(USER) = 10^6 tuples
- distinct values of Country = 100
- card(PURCHASE) = $5 \cdot 10^7$ tuples
- MIN(DATE(Timestamp)) = 1/1/2018, MAX(DATE(Timestamp)) = 31/12/2019

Furthermore, assume the following reduction factor for the group by condition:

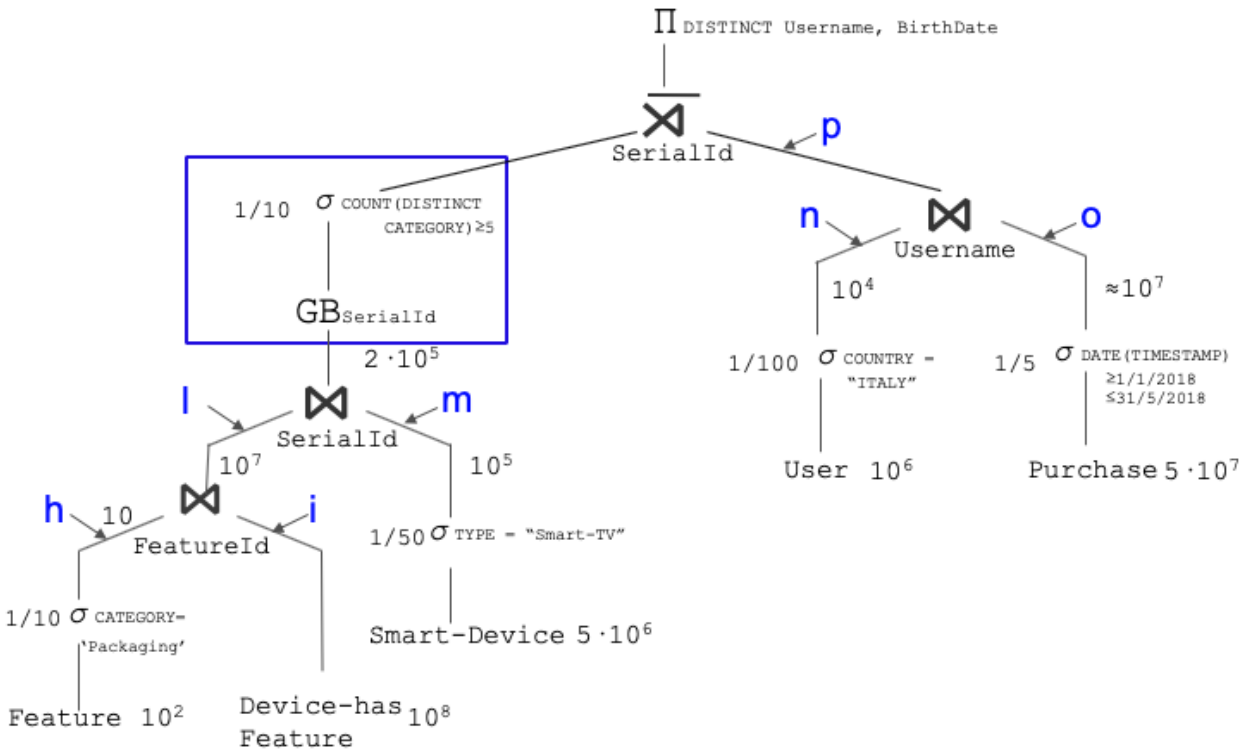
HAVING COUNT(DISTINCT F.Category) >= 5 ≈ 1/10

Consider the following query

```
select DISTINCT Username, BirthDate
from USER U, PURCHASE P
where U.Username=P.Username and Country='Italy'
and DATE(Timestamp) >= 1/1/2018 and DATE(Timestamp) <= 31/5/2018
and SerialId NOT IN (select SerialId
                     from SMART-DEVICE D, FEATURE F, DEVICE-HAS-FEATURE DHF
                     where D.SerialId=DHF.SerialId and F.FeatureId=DHF.FeatureId
                     and F.Category='Packaging'
                     D.Type = 'SmartTV'
                     group by SerialId
                     having COUNT(DISTINCT F.Category)>=5)
```

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

Analyze the Group By anticipation.



(a) It is possible to anticipate it in branch l ✔

(b) It is not possible to anticipate the Group By

(c) It is possible to anticipate it in branch p

(d) It is possible to anticipate it in branch h

(e) It is possible to anticipate it in branch o

(f) It is possible to anticipate it in branch m

(g) It is possible to anticipate it in branch n

(h) It is possible to anticipate it in branch i

Indices (2 points, -15% penalty for each wrong answer)

SMART-DEVICE(SerialId, Name, Brand, Type, Price)
FEATURE(FeatureId, Name, Category)
DEVICE-HAS-FEATURE(SerialId, FeatureId, Value, UnitOfMeasurement, DisplayValue)
USER(Username, FirstName, LastName, BirthDate, Address, City, Country)
PURCHASE(Timestamp, Username, SerialId, TotalCost, NumberOfItems)

Assume the following cardinalities:

- card(SMART-DEVICE) = $5 \cdot 10^6$ tuples
- distinct values of Type = 50
- card(FEATURE) = 10^2 tuples
- distinct values of Category = 10
- card(DEVICE-HAS-FEATURE) = 10^8 tuples
- card(USER) = 10^6 tuples
- distinct values of Country = 100
- card(PURCHASE) = $5 \cdot 10^7$ tuples
- MIN(DATE(Timestamp)) = 1/1/2018, MAX(DATE(Timestamp)) = 31/12/2019

Furthermore, assume the following reduction factor for the group by condition:

HAVING COUNT(DISTINCT F.Category) >= 5 ≈ 1/10

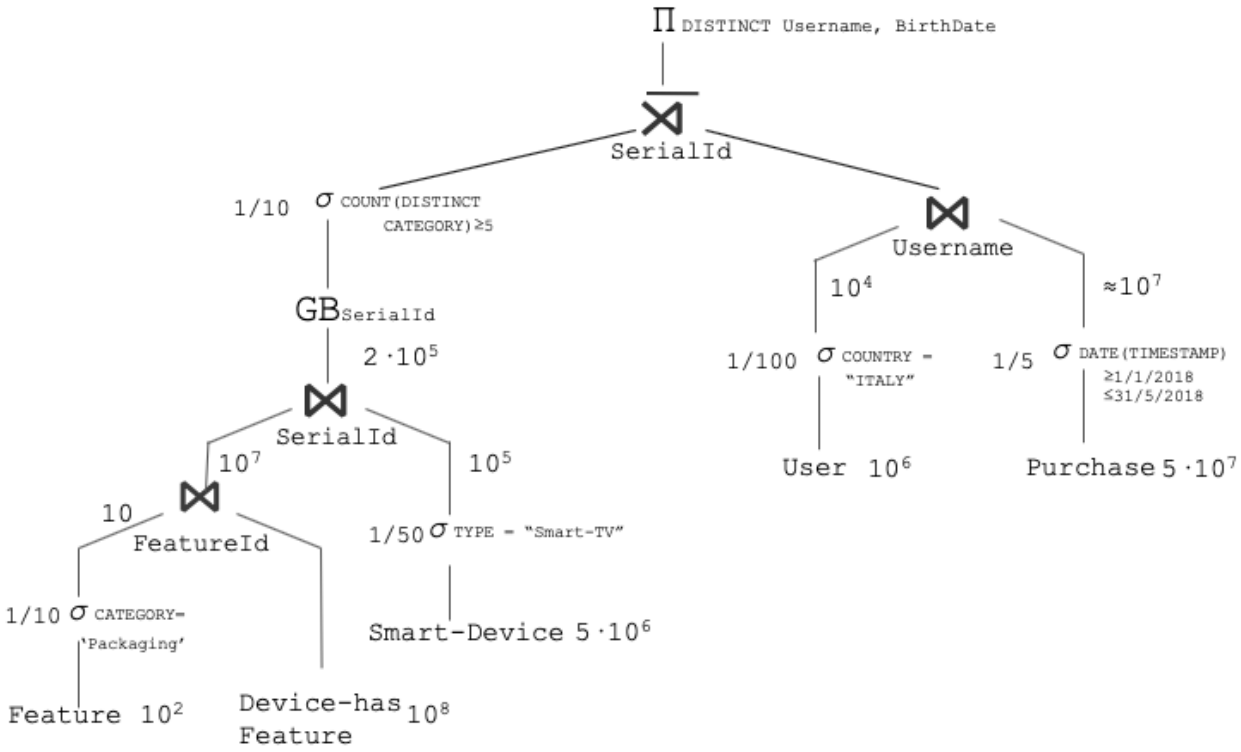
Consider the following query

```
select DISTINCT Username, BirthDate
from USER U, PURCHASE P
where U.Username=P.Username and Country='Italy'
and DATE(Timestamp) >= 1/1/2018 and DATE(Timestamp) <= 31/5/2018
```

```
and SerialId NOT IN (select SerialId
                     from SMART-DEVICE D, FEATURE F, DEVICE-HAS-FEATURE DHF
                     where D.SerialId=DHF.SerialId and F.FeatureId=DHF.FeatureId
                     and F.Category='Packaging'
                     D.Type = 'SmartTV'
                     group by SerialId
                     having COUNT(DISTINCT F.Category)>=5)
```

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

Select the secondary physical structures to increase query performance (if possible).



Select one or more alternatives:

(a) CREATE INDEX IndexB ON DEVICE-HAS-FEATURE(SerialId) - HASH

(b) CREATE INDEX IndexF ON USER(Country) - HASH ✔

(c) None - secondary physical structures would not increase query performance

(d) CREATE INDEX IndexG ON PURCHASE(Date) - B^+-Tree

(e) CREATE INDEX IndexC ON DEVICE-HAS-FEATURE(FeatureId) - B^+-Tree

(f) CREATE INDEX IndexA ON FEATURE(Category) - HASH

(g) CREATE INDEX IndexD ON SMART-DEVICE(Type) - HASH ✔

(h) CREATE INDEX IndexE ON SMART-DEVICE(SerialId) - HASH

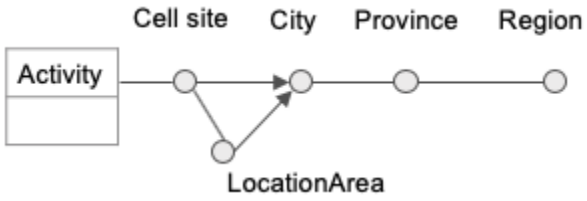
Conceptual schema 1 (1 point, -15% penalty for a wrong answer)

Data analysts of an Italian mobile service provider are interested in analyzing statistics about the usage of their cell sites, such as the number of phone calls and their average duration.

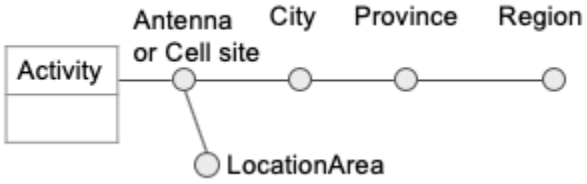
- Each **antenna** of the mobile network defines a geographical area called **cell site**.
- The geographical **position** of cell sites can be described in two different ways.
- The first one consists in dividing the territory in geographical **zones** called "**location areas**".
- Each **cell site** belongs to one and only one "**location area**"
- The second way consists in considering the city where the **antenna** is located. Subsequently, we can also consider the **province** and the **region** of the city.
- A **city** can be covered by different "**location areas**".
- A "**location area**" can independently cover different **cities**.

Model the conceptual schema that defines the hierarchy related to the position of a cell site.

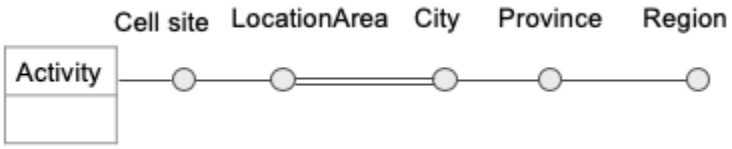
(a)



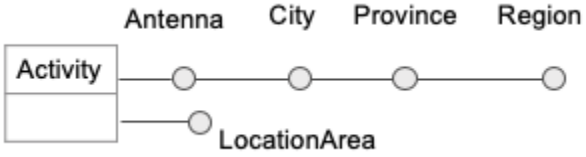
(b) ✔



(c)



(d)



(e)

(f)

Conceptual schema 2 (1 point, -15% penalty for a wrong answer)

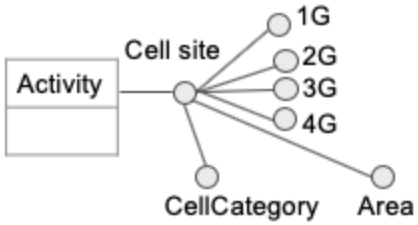
Data analysts of an Italian mobile service provider are interested in analyzing statistics about the usage of their cell sites, such as the number of phone calls and their average duration.

- Based on its **characteristics**, a **cell site** belongs to one of the following **categories**:
    - “macro-cell",
    - “micro-cell",
    - “selective-cell"
    - “umbrella-cell".
- For each **cell site**, the available **technologies** are known (one or more among the following: 1G, 2G, 3G, 4G).
- Finally, it is known the **area** of the territory covered by the cell site.
    - Big: more than 10 km^2
    - Medium: between 5 and 10 km^2

- Small: less than 5 km^2

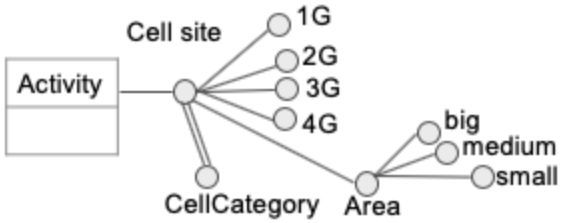
Model the conceptual schema to define the characteristics of a cell site.

(a)

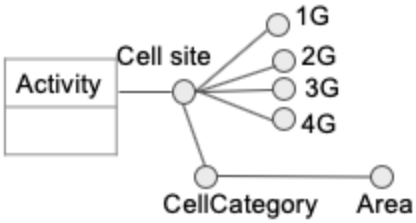


✔

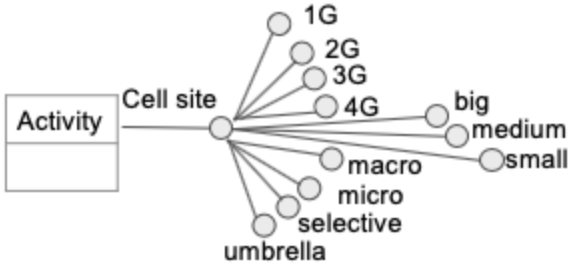
(b)



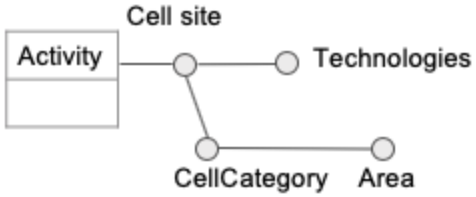
(c)

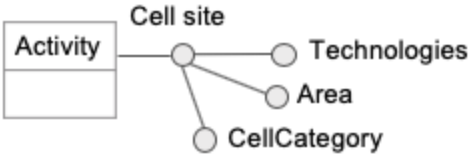


(d)



(e)





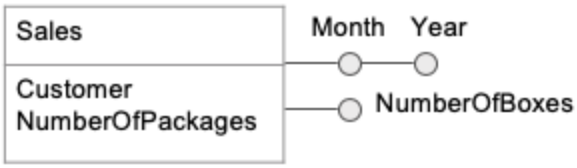
(f)

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

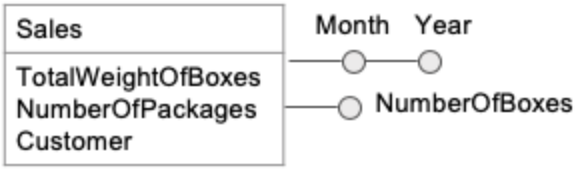
Data analysts of an industry specialized in the wholesale of nuts are interested in analyzing statistics about their sales.

- Nuts are sold in packages. Multiple packages are grouped into boxes.
- Each box can contain either 6 or 12 packages.
- They want to analyze the statistics based on the month of the sales (e.g. April 2020, May 2020, etc...), the year of the sales and the customer.
- The statistics they are interested in consist of the number of sold boxes, the average weight of a box and the average number of packages per box

Design the conceptual schema of the data warehouse according to the above specifications.



(a)

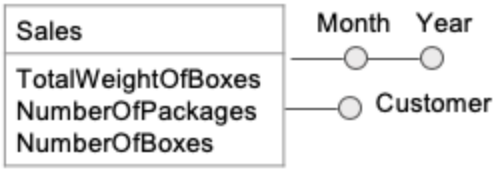


(b)

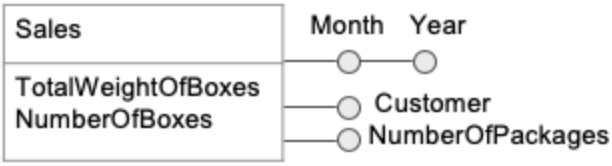
(c)

(d)

(e) ✔



(f)



Extended SQL query 1 (4 points)

Given the following relational schema, write the requested SQL queries.

CellSite (CellSiteId, city, region)
Date (DateId, date, month, monthOfYear, semester, year)
Activity (DateId, VirtualNetOperator, CellSiteId, numberOfTransitions, minutesOfCall)

Separately for city, region, and year, compute:

- the daily average number of transitions
- the percentage of transitions in each city, with respect to the total of the region
- inside each region assign a rank to to the cities based on the decreasing number of transitions

```
SELECT city, region, year
    SUM(#transitions)/COUNT(DISTINCT Date)
    SUM(#transitions)/SUM(SUM(#transitions)) OVER (PARTITION BY region, year)
    RANK() OVER (PARTITION BY region, year ORDER BY SUM(#transitions) DESC),

FROM CellSite, Date, Activity
WHERE CellSite.CellSiteId=Activity.CellSiteId AND
Activity.DateId=Date.DateId
GROUP BY city, region, year
```

Extended SQL query 2 (4 points)

Given the following relational schema, write the requested SQL queries.

CellSite (CellSiteId, city, region)
Date (DateId, date, month, monthOfYear, semester, year)
Activity (DateId, VirtualNetOperator, CellSiteId, numberOfTransitions, minutesOfCall)

Separately for each city, month and virtual network operator, compute:

- the total number of call minutes
- the percentage of call minutes of each city with respect to the monthly total over all the cities, for the considered operator
- the cumulative total of call minutes from the beginning of the year

```
SELECT city, month, year, VirtNetOperator,
    SUM(minutes),
    SUM(minutes)/SUM(SUM(minutes)) OVER (PARTITION BY month,
VirtNetOperator),
    SUM(SUM(minutes)) OVER (PARTITION BY city, year, VirtNetOperator
                    ORDER BY month ROWS UNBOUNDED PRECEDING)
FROM CellSite, Date, Activity
WHERE CellSite.CellSiteId=Activity.CellSiteId AND
Activity.DateId=Date.DateId
GROUP BY city, month, year, VirtNetOperator
```

Trigger 1 (7 points)

The following relations are given (primary keys are underlined).

SUPERMARKET(SupermarketCode, City, NumeroPlacesInQueue, OpeningTime, ClosingTime)
ENTRANCE_QUEUE(SupermarketCode, QueuePosition, CustomerSSN)
ENTRANCE_REQUEST(RequestCode, SupermarketCode, CustomerSSN, Date, Time)

We would like to manage automatically the entrance queues in a chain of supermarkets. Write the trigger to manage the following activity.

Request of insertion in the entrance queue

When a customer wants to enter in the queue for a specific supermarket, a new record is inserted into the ENTRANCE_REQUEST table. The following activities must be executed.

(a) *Check the availability of a place in the queue for the requested supermarket.*

Queuing is done by assigning an increasing position, starting from 1 and up to the maximum number of places available in the queue for the supermarket (equal to the value of the NumberOfPlacesInQueue attribute in the SUPERMARKET table). Free positions (if any) are at the end of the queue. The position of a customer in the supermarket queue is identified by the QueuePosition attribute in the ENTRANCE_QUEUE table.

To check the availability of free places in the queue for the requested supermarket, it is necessary to verify that the maximum number of places available in the queue has not been reached. If there is no place available in the queue for the requested supermarket, the trigger ends with an error.

(b) *Insertion in the queue.*

The insertion of a customer in the queue for the requested supermarket takes place by assigning the position immediately following the last occupied position in the queue (the new position is increased by 1 with respect to the last occupied position). You must consider also the case where the queue is empty.

```
create or replace trigger QUEUE_INSERTION
after insert on ENTRANCE_REQUEST
for each row

declare
Position number;
MaxPlaces number;

begin

---Find the last position occupied in the queue
select max (QueuePosition) INTO Position
from ENTRANCE_QUEUE
where SupermarketCode = :new.SupermarketCode;

---- Read the maximum number of places in the queue
select NumberOfPlacesInQueue INTO MaxPlaces
from SUPERMARKET
where SupermarketCode = :new.SupermarketCode;

--- Check is the queue is full or empty
if (Position = MaxPlaces) then

--- The queue is full
```

```
raise_application_error(....);
end if;

--- The queue is not full; the customer is inserted into the queue

if (Position IS NULL) then
--- The queue is empty. The first position is assigned
Position := 1;

else
- -- The queue is not empty. The position immediately after the last
one occupied is assigned
Position := Position + 1;
end if;
```

--- insertion into the queue

insert into ENTRANCE_QUEUE (SupermarketCode, QueuePosition, CustomerSSN)

values (:new.SupermarketCode, Position, :new.CustomerSSN);

end;

Trigger 2 (3 points)

The following relations are given (primary keys are underlined).

SUPERMARKET(SupermarketCode, City, NumeroPlacesInQueue, OpeningTime, ClosingTime)
ENTRANCE_QUEUE(SupermarketCode, QueuePosition, CustomerSSN)
ENTRANCE_REQUEST(RequestCode, SupermarketCode, CustomerSSN, Date, Time)

We would like to manage automatically the entrance queues in a chain of supermarkets. Write the trigger to manage the following activity.

Integrity constraint on the duration of the opening

There can be at most 10 supermarkets in Turin with a duration of the opening (difference between ClosingTime and OpeningTime attributes in the SUPERMARKET table) greater than 18 hours. Any

modification of the SUPERMARKET table that causes the constraint violation must not be executed. Carefully evaluate all the triggering events on table SUPERMARKET.

```
create trigger CheckOpeningTime
after insert or update of OpeningTime, ClosingTime, City on
SUPERMARKET

declare
X number;

begin

select count(*) into X
from SUPERMARKET
where City` = `Torino' and (ClosingTime-OpeningTime) > 18;

if (X > 10) then
      raise_application_error(...);
end if;
end;
```