Given a collection of documents, each describing a photo, the statement

```javascript
db.photos.updateMany( {user: "john", tag: "seaside"}, { $addToSet: {tag: "Riccione"} } );
```

- (a) removes the tag “Riccione” to one photo belonging to the user “john” and having the value “seaside” in the tag list
- (b) adds the tag “Riccione” to one photo belonging to the user “john” and having the tag field equal to “seaside”
- (c) adds the tag “Riccione” to all the photos belonging to the user “john” and having the value “seaside” in the tag list
- (d) sets the tag field to be equal to “Riccione” to all the photos belonging to the user “john” and having the tag field equal to “seaside”

Risposta errata.
La risposta corretta è: adds the tag “Riccione” to all the photos belonging to the user “john” and having the value “seaside” in the tag list

In the MongoDB aggregation pipeline, which stage operator is used to output a new document for each element of an array:

- (a) $unwind
- (b) $match
- (c) $group
- (d) $foreach
- (e) $project

Risposta errata.
La risposta corretta è: $unwind
In a master-slave distributed database setting, when the replication is asynchronous:

- (a) a failure of the master always causes the data to be lost
- (b) data can be lost only if the majority of the slaves fail
- (c) data can be lost even if the master has already committed
- (d) data cannot be lost if the slaves do not fail

Risposta errata.
La risposta corretta è: data can be lost even if the master has already committed

Which one of the following answers is a direct consequence of Steven's law?

- (a) Ordinal measure should be mapped to increasing saturation and intensity
- (b) It is important to avoid comparisons between areas
- (c) For every single attribute no more than four distinct levels are discernible
- (d) There is no common magnitude assessment for the curvature
- (e) The length of non-aligned objects is harder to compare

Risposta errata.
La risposta corretta è: It is important to avoid comparisons between areas
Data analysts of an Italian high-speed train operator are interested in designing a new data warehouse to analyze some key performance indicators of their train trips. A trip consists of a specific train travelling from a departure to a destination station, stopping by in different intermediate stations. In the original database, the start and stop timestamps of each trip are recorded together with the scheduled times.

The new data warehouse must be designed to efficiently analyze:

A. the average **duration** of the trips,
B. the average **length** (in km) of the trips,
C. the average number of minutes of **delay** of the trips (measured at the destination station),
D. and the average number of intermediate **stations** of the trips,

according to the following dimensions:

- Departure and destination **station**, the **city** of the station, its **province**, its **region**.
- The **model** of the train.
  - Each model is built by a specific **manufacturer**.
  - A manufacturer can be associated with many train models.
- Each train model offers several **services**. The systems stores which services are available for each train model.
  - Examples of additional services are “bar”, “restaurant”, “wi-fi”, “air conditioning”.
  - The number of additional services is large and growing, hence the full list is not known a priori.
- Each trips is characterized by an **interruption class**, defined as follows.
  - High class with 5 or more stops in intermediate stations
  - Medium class with 2, 3 or 4 stops in intermediate stations
  - Low class with less than 2 stops in intermediate stations

Select, among the following dimensions, those that meet the requirements described in the problem specification (at most one answer is correct).
Risposta errata.

La risposta corretta è:

![Diagram](image-url)
Data analysts of an Italian high-speed train operator are interested in designing a new data warehouse to analyze some key performance indicators of their train trips. A trip consists of a specific train travelling from a departure to a destination station, stopping by in different intermediate stations. In the original database, the start and stop timestamps of each trip are recorded together with the scheduled times. The new data warehouse must be designed to efficiently analyze:

A. the average **duration** of the trips,
B. the average **length** (in km) of the trips,
C. the average number of minutes of **delay** of the trips (measured at the destination station),
D. and the average number of intermediate **stations** of the trips,

according to the following dimensions:

- Departure and destination **station**, the **city** of the station, its **province**, its **region**.
- The **model** of the train.
  - Each model is built by a specific **manufacturer**.
  - A manufacturer can be associated with many train models.
- Each train model offers several **services**. The system stores which services are available for each train model.
  - Examples of additional services are “bar”, “restaurant”, “wi-fi”, “air conditioning”.
  - The number of additional services is large and growing, hence the full list is not known a priori.
- Each trip is characterized by an **interruption class**, defined as follows.
  - High class with 5 or more stops in intermediate stations
  - Medium class with 2, 3 or 4 stops in intermediate stations
  - Low class with less than 2 stops in intermediate stations

Select, among the following dimensions, those that meet the requirements described in the problem specification (at most one answer is correct).
La risposta corretta è:

![Diagram](image.png)
Data analysts of an Italian high-speed train operator are interested in designing a new data warehouse to analyze some key performance indicators of their train trips.

A trip consists of a specific train travelling from a departure to a destination station, stopping by in different intermediate stations.

In the original database, the start and stop timestamps of each trip are recorded together with the scheduled times.

The new data warehouse must be designed to efficiently analyze

A. the average **duration** of the trips,
B. the average **length** (in km) of the trips,
C. the average number of minutes of **delay** of the trips (measured at the destination station),
D. and the average number of intermediate **stations** of the trips,

according to the following dimensions.

- Departure and destination **station**, the **city** of the station, its **province**, its **region**.
- The **model** of the train.
  - Each model is built by a specific **manufacturer**.
  - A manufacturer can be associated with many train models.
- Each train model offers several **services**. The systems store which services are available for each train model.
  - Examples of additional services are “bar”, “restaurant”, “wi-fi”, “air conditioning”.
  - The number of additional services is large and growing, hence the full list is not known a priori.
- Each trip is characterized by an **interruption class**, defined as follows.
  - High class with 5 or more stops in intermediate stations
  - Medium class with 2, 3 or 4 stops in intermediate stations
  - Low class with less than 2 stops in intermediate stations

Select, among the following dimensions, those that meet the requirements described in the problem specification (at most one answer is correct).

(a)  
(b)  
(c)
Risposta errata.

La risposta corretta è:
Data analysts of an Italian high-speed train operator are interested in designing a new datawarehouse to analyze some key performance indicators of their train trips.

A trip consists of a specific train travelling from a departure to a destination station, stopping by in different intermediate stations.

In the original database, the start and stop timestamps of each trip are recorded together with the scheduled times.

The new data warehouse must be designed to efficiently analyze

A. the average duration of the trips,
B. the average length (in km) of the trips,
C. the average number of minutes of delay of the trips (measured at the destination station),
D. and the average number of intermediate stations of the trips,

according to the following dimensions.

- Departure and destination station, the city of the station, its province, its region.
- The model of the train.
  - Each model is built by a specific manufacturer.
  - A manufacturer can be associated with many train models.
- Each train model offers several services. The systems stores which services are available for each train model.
  - Examples of additional services are “bar”, “restaurant”, “wi-fi”, “air conditioning”.
  - The number of additional services is large and growing, hence the full list is not known a priori.
- Each trip is characterized by an interruption class, defined as follows.
  - High class with 5 or more stops in intermediate stations
  - Medium class with 2, 3 or 4 stops in intermediate stations
  - Low class with less than 2 stops in intermediate stations

Select all and only the required measures of the fact table in the conceptual schema design among the following (multiple choice question). Hint: do consider the dimensions defined by the previous answers.

Scegli una o più alternative:

- (a) Average delay per destination station (minutes)
- (b) Total duration of trips (minutes)
- (c) Total delay of the trips (minutes)
- (d) Total number of trips (count)
- (e) Total number of intermediate stations of the trips (count)
- (f) Average length per trip (km)
- (g) Total number of train models (count)
- (h) Average number of intermediate stations per trip (count)
- (i) Average duration per trip (minutes)
- (j) Total number of departure stations per trip (count)
- (k) Total number of destination stations per trip (count)
- (l) Total length of the trips (km)
- (m) Average delay per trip (minutes)
- (n) Number of services (count)
- (o) Average number of trips (count)
- (p) Total number of trains (count)

Risposta errata.
La risposta corretta è: Total number of trips (count), Total duration of trips (minutes)
Given the following conceptual schema:

- A video game has a unique name and it is distributed by a video game company.
- Each video game has a specific genre.
- A videogame can be appropriate for children or not.
  - The value of this field can be "0" for not appropriate and "1" for appropriate.
- A store is identified by a unique name. Stores are analyzed according to their city and country.
- Each store may sell some additional articles. There are 4 possible types of additional articles: "collectable", "toys", "manga" and "accessories".
- The system records the sales with their date, the day of the week and if the day was an holiday or not. It also records the month, year, bimester, trimester and semester of the sales.

Write the logical design of the conceptual DW schema indicated in the picture. Write each table on a new line. Use the **bold** or the *underline* for identifying primary-key attributes.

```
VideoGame(CodV, VideoGameName, forChildren, Genre, Company)
Store(CodS, Store, City, Country, Collectable, Toys, Mangas, Accessories)
Time(CodT, date, dayOfTheWeek, holiday, month, 2M, 3M, 6M, year)
Fact(CodV, CodS, CodT, total_revenues, total_sold_videogames)
```
For each shipping, the departure and arrival cities, provinces and regions are recorded.
For the shipping, the courier agency is recorded. The courier agency has a unique name.
Each agency belongs to a Corporate group.
The system stores the arrival date, the day of the week and if the day was an holiday or not. It also stores the month, year and semester.

Separately for each courier agency and departure city, compute the following metrics:
A. the percentage of packages with respect to the total number of packages of the agency for the departure region
B. the average weight per package
C. assign a rank to each courier agency within its corporate group, based on its total number of packages (rank 1st the courier agency with the highest number of shipped packages in its corporate group for each departure city)

SELECT CourierAgencyName, L.city
     100*SUM(#packages)/SUM(SUM(#packages))
        OVER (PARTITION BY L.region, CourierAgencyId) as B,
     SUM(total_weight)/SUM(#packages) as A,
     RANK() OVER (PARTITION BY L.city, CorporateGroup
              ORDER BY SUM(#packages) DESC) as C
FROM CourierAgency CA, Location L, Shippings S
WHERE CA.CourierAgencyId=s.CourierAgencyId and S.DepartureLocationId=L.LocationId
GROUP BY CourierAgencyId, CourierAgencyName, L.city, L.region, CorporateGroup
For each shipping, the departure and arrival cities, provinces and regions are recorded.
For the shipping, the courier agency is recorded. The courier agency has a unique name. Each agency belongs to a Corporate group.
The system stores the arrival date, the day of the week and if the day was an holiday or not. It also stores the month, year and semester.

Separately for each month, departure province and arrival province, compute the following metrics:
A. the daily average number of shipped packages
B. the cumulative total weight of delivered packets since the beginning of the semester

```
SELECT month, L1.province, L2.province, (6M),
       SUM(#packages)/COUNT(distinct date) as A,
       SUM(SUM(total_weight)) OVER (PARTITION BY L1.province, L2.province, 6M
                                     ORDER BY month
                                     ROWS UNBOUNDED PRECEDING) as B,
FROM Location L1, Location L2, Shippings S, Time T
WHERE S.DepartureLocationId=L1.LocationId and S.ArrivalLocationId=L2.LocationId and T.TimeId=S.TimeId
GROUP BY month, L1.province, L2.province, 6M
```
The following document structure represents cameras sold by an e-commerce. Each document collects the aggregated metrics of one day.

```json
{ "_id": "nikon_d3500",  
  "model": "D3500",  
  "brand":{  
    "name": "Nikon",  
    "url": "https://www.nikon.it/"  
  },  
  "releaseDate": Date("2018-08-28T00:00:00.000Z"),  
  "category": "DSRL",  
  "price": 435,  
  "specs":{  
    "resolution": 24,  
    "technology": "APS-C CMOS",  
    "min_ISO": 100,  
    "max_ISO": 25600,  
    "weight": 365,  
    "viewfinder": "optical",  
    "video_resolution": "1920 x 1080"  
  },  
  "scores":{  
    "overall": 57,  
    "image_quality": 48,  
    "versatility": 62,  
    "comfort": 85,  
    "speed": 41  
  }
}
```

Write a MongoDB query to display only the model, the price, and the brand name of cameras released in 2021, belonging to the "laser" category, and whose overall score is in the 70-90 range.

**N.B.** Use the syntax `new Date (string)` to manage date attributes, e.g., "start": new Date("2021-06-01")

```javascript
db.cameras.find(  
  {  
    category: 'laser',  
    releaseDate: {  
      $gte: new Date('2021-01-01'),  
      $lt: new Date('2022-01-01')  
    },  
    'scores.overall': {  
      $gte: 70,  
      $lte: 90  
    }  
  },  
  {model:1, "brand.name":1, price:1, _id:0}
)
```
The following document structure represents cameras sold by an e-commerce.

Each document collects the aggregated metrics of one day.

```json
{ "_id": "nikon_d3500", "model": "D3500", "brand": { "name": "Nikon", "url": "https://www.nikon.it/" }, "releaseDate": new Date("2018-08-28T00:00:00.000Z"), "category": "DSRL", "price": 435, "specs": { "resolution": 24, "technology": "APS-C CMOS", "min_ISO": 100, "max_ISO": 25600, "weight": 365, "viewfinder": "optical", "video_resolution": "1920 x 1080" }, "scores": { "overall": 57, "image_quality": 48, "versatility": 62, "comfort": 85, "speed": 41 } }
```

Considering only cameras released since 2015, for each release year and for each category, select the median overall score.

**N.B.** Use the operator `$year` to extract the year from the date, e.g., `$year: "$releaseDate"`

Use the syntax `new Date (string)` to manage date attributes, e.g., "start": new Date("2021-06-01")
db.measures.aggregate([
  {$match: {releaseDate: {$gte: new Date('2015-01-01')}}},
  {$sort: {'$scores.overall': 1}}],
  {$group: 
    {_id: 
      {'cat': '$category',
       'y': { $year: '$releaseDate' }},
     'value': 
       {'$push': '$scores.overall'}
    }
  },
  {$project: 
    { _id: 1, 
      "median": 
        {
          $arrayElemAt: ['$value', 
                        {
                          $floor: 
                            {
                              $multiply: 
                                [0.50, {$size: '$value'}]
                            }
                        }]
        }
    }
  ])
)
Design a MongoDB database to manage online courses according to the following requirements.

Teachers are characterized by their name, surname, email, and list of subjects they can teach (e.g., maths, electronics, etc.). Each teacher can have one or more online profiles on different platforms (e.g., Facebook, LinkedIn, Wikipedia, etc.). For each online profile, if available, the database tracks the corresponding URL of the profile (e.g., https://en.wikipedia.org/wiki/Ranjitsinh_Disale). Note that for each teacher and each platform, at most one profile can exist. A teacher can teach different courses.

The courses are characterized by a name, a syllabus, a list of keywords, and the teacher. Each course has several editions. For each edition, the start date, the end date, and the number of enrolled students are known.

Given a course, the database must be designed to efficiently provide the name, the surname and the email of its teacher.

Furthermore, given a course, the number of editions and the average number of enrolled students in each edition must be efficiently returned.

Teachers are typically retrieved by subject (e.g., all those teaching maths), and by online profile platform (e.g., all those having a wikipedia page).

Write a sample document for each collection of the database.

Explicitly indicate the design patterns used.

```
Teacher
{
   _id: ObjectId(),
   name: <string>,
   surname: <string>,
   email: <string>,
   profiles: {
         facebook: <url>,
         linkedin: <url>,
         ....
   }
   subjects: [<string>]
}

Course
{
   _id: ObjectId(),
   name: <string>,
   syllabus: <string>,
   keywords: [<string>],
   teacher: {
         _id: ObjectId(),
         name: <string>,
         surname: <string>,
         email: <string>,
   }
   editions: [
            {start: <date>,
            end: <date>,
            n_students: <number>}
            n_editions: <number>,
   tot_students: <number>
   }
}

Pattern used:
Polymorphic pattern to track the online profile information in the Teacher collection.
```
Extended reference pattern to track the teacher information associated with each course.
Bucket pattern to track when a course is provided.
Computed pattern for average students on each edition.

**Question**

Is there a clearly defined question addressed by the visualization? Write it down.
Data

Is the data quality appropriate? Identify the inadequate characteristics and explain.
Visual Proportionality

Are the values encoded in a uniformly proportional way?
**Visual Utility**

All the elements in the graph convey useful information?
In football, there's a lot on the line both on and off the pitch. Today, the top 10 clubs hold a combined market value of $36 billion.

And as of late, the status quo has been challenged via the proposed European Super League (ESL).

**Visual Clarity**

Are the data in the graph clearly identifiable and understandable (properly described)?
Design data

Design the visualization based on the following data structure (to be completed).
Design schema & Sketch

Fill in the required schema elements; formulas can be used if required. Then describe in words the design proposal.