Undone topics:

- Temporal databases
  - *Define the concepts of valid time and transaction time in temporal databases also describing their advantages and disadvantages.*
- Personalization
  - *Describe the problem of personalization and list the various kinds of personalization with a small explanation of each.*
- Pervasive data management
  - *Briefly define pervasive data management and the main problems that must be solved.*
- Data mining
  - *Define the concept of association rule in the data mining context, and give a small example.*
  - *Consider these data mining problems: classification and clustering. Define them, discuss the differences between the two problems, and provide an application example for each of them.*
  - *Consider the following data mining problems: frequent itemset mining and association rule discovery. Define them, discuss the differences between the two problems, and provide an application example for each of them.*

    Association Rule Discovery [Descriptive]: given a set of records each of which contains some number of items from a given collection. Produce dependency rules which will predict the occurrence of an item based on occurrences of other items. Rule mining that finds the frequency of relationships between data items within a large dataset. Association rule mining finds the frequency of relationships between data items within large data sets, expressed as if-then statements.
  - *Consider the following data mining problem: Sequential Pattern Discovery. Define and illustrate it clearly with an application example.*

    A sequence is an ordered list of elements (transactions) s

    = < e1 e2 e3 … >

    – Each element contains a collection of events (items)

    ei = {i1, i2, …, ik}

    – Each element is attributed to a specific time or location

    • A k-sequence is a sequence that contains k elements

    Given a set of sequences and support threshold, find the complete set of frequent subsequences

    An element may contain a set of items. Items within an element are unordered and we list them alphabetically. Given support threshold min_sup =2, is a sequential pattern
- Box-plot
  - *Describe the Box-Plot method for displaying the distribution of data, using an example to illustrate it clearly.*

    Box plot is a method for graphically demonstrating the distribution, locality and spread of the data. A box plot is standardized the dataser based on the five-number: the minimum, the maximum, the sample median, and the first and the third quantile.
    - Minimum or the 0th percentile: the lowest data point in the dataset excluding the outliers.

- Maximum or 100th percentile: the highest data point in the data set excluding any outliers.
- Median or 50th percentile: the middle value in the data set.
- Fist quartile (Q_1 OR 5th percentile): also known as the lower quartile $q_n(0.25)$, it is the median of the lower half of the dataset
- Third quartile (Q_3 OR 75th percentile): also known as the upper quartile $q_n(0.75)$, it is the median of the upper half of the dataset.

The box is drawn from Q_1 to Q_3 with a horizontal line drawn in the middle to denote the median. A box-plot usually includes two parts, a box and a set of whiskers. The lowest point on the box-plot (i.e. the boundary of the lower whisker) is the minimum value of the data set and the highest point (i.e. the boundary of the upper whisker) is the maximum value of the data set (excluding any outliers).

THEORY QUESTIONS

1. *What is data integration? Which type of data integration (materialized or virtual) is used in the case of Data Warehousing? Describe the main differences between the two approaches and justify the answer in your own words, possibly using a small example.*

   Data Integration process aims at combining data coming from different data sources providing the user with a unified vision of the data. The process involves detecting correspondences between similar concepts that come from different sources, and conflict-solving. Data has four interesting dimensions to be considered: Volume, Variety, Velocity and Veracity. In particular, VARIETY corresponds to heterogeneity among several data collections to be used together:
   - Technological heterogeneity
   - Model heterogeneity
   - Language heterogeneity
   - Schema (semantic) heterogeneity
   - instance (semantic) heterogeneity.

   The steps of Data integration are:
   - Schema reconciliation: mapping the data structure (if it exists!)
   - Record linkage (aka Entity resolution): data matching based on the same content
   - Data Fusion: reconciliation of non-identical content

   There are two relevant ways of integrating Database Systems:
   - Materialized Integration: data are merged in a new database. This is the approach adopted in data warehouses, and the software to access, scrape, transform, and load data into warehouses, became known as extract, transform, and load (ETL) systems. ETL is performed periodically, building a history of the enterprise, and allowing systematic or ad-hoc data analysis and mining. This approach is not indicated when data needs to be kept up-to-date. Different sources are reconciled in a unique database and query goes on the database and not on the original sources which could be updated. When data warehouses are queried, we don't need fresh and up-to-date information but we need trends, we don't need up-to-date to the last minute information (data warehouses are updated periodically (say once a week/month for example) --> we do not need virtual integration --> MATERIALIZED INTEGRATION. Example of context in which is useful to have materialized integration: a sale company, the history of sales can be useful to detect trends and patterns to improve the company's performance
   - Virtual Integration: this approach leaves the information requested in the local sources. The virtual approach will always return a fresh answer to the query (because of the data exchange from source-to-target). The query posted to the global schema is reformulated into the formats of the local information system. The information retrieved needs to be combined to answer the query. The integration designer creates the Global schema which is a schema that, when queried, provides corresponding queries to the components' sources. The integration engine will transform the query over the global schema to queries on the data sources, the answers from the data sources will be reconciled again by the integration system and provided to the user. Example of context in which virtual integration is fundamental: a travel agency wants to access to information regarding hotel reservations coming form different sources and visible all together on the same platform, in this case data must be up to date.

2. *Define what is a Data Warehouse and describe the dimensional fact model used in the data warehouse context and define its main elements. Provide a small example.*

(A data warehouse is a data collection built to support decision-making processes. It must allow analytical queries useful for the management of the company. It is usually built starting from several data sources and we update it only periodically.). A Data Warehouse is a single, complete and consistent store of data obtained from a variety of different sources made available to end users, so that they can understand and use it in a business context for organizational decision making. This collection of data is used primarily in organizational decision making (typical of direction and management department) and is:

- integrated (not only the information across all parts of data warehouse but also other information that can come from other companies, websites, etc...)
- subject-oriented (data are organized according to a certain subject)
- non-volatile (it never changes the part that we have already stored, we do it only if there is a mistake, it is only growing and it is never updated)
- time-varying (we keep all the variations in time, we not only store what is interesting for us today but also the past). Usually are very large databases.

The main purpose of a data warehouse is to allow systematic or ad-hoc data analysis and mining (→ materialized data integration). Data warehouses are very large databases. A DW is useful for:

- Commerce: sales and complaints analysis, client fidelization, shipping and stock control
- Manufacturing plants: production cost control, provision and order support
- Financial services: risk and credit card analysis, fraud detection
- Telecommunications: call flow analysis, subscribers' profiles
- Healthcare structures: patients' ingoing and outgoing flows, cost analysis

In order to build a Data Warehouse, we start from designing its conceptual model. This phase is supported by the dimensional fact model, an intuitive, graphic formalism developed ad-hoc for this phase, which can used also by analysts and non-technical users. A model is a collection of fact schemata. A fact schema is composed by the following elements (for example for a policies company):

- Fact: a relevant concept for the decision-making process, typically it models a set of events of the organization. It describes an N:M relationship among its dimensions (=policies).
- Dimensions: fact properties defined on a finite domain. They are the coordinates of the fact(=year,type,age).
- Measures: numerical properties of the fact(=n° of policies).
- Dimension hierarchy: a dimension can be structured in a hierarchy, for instance: Day -> Month -> Year.

(Another example, store chain: fact sales, measures: sold quantity, gross income, dimensions: product, time, zone).

3. *Describe the conflict analysis step in data integration; describe the most important classes of conflicts and provide a set of examples*

Once the identification of the sources and the reverse engineering of their schemata have been completed, the next step in the data integration process is conflict resolution and reconstructing. The phases are:

- identification of related concepts across sources: simple process if manual but very difficult to be automatized.
- Conflict analysis and resolution: the most important classes of conflicts are the following:

        - Name: such as homonyms and synonyms, for instance price that means standard and discounted, or employee/agent.
        - Type: when the same attribute has different types, for instance M/F or 0/1. It can be also two different entities that represent the same real concept with different attributes (example: message and report have name conflict and also they can have different attributes, even if in the real world they depict the same concept).
        - Data Semantics: different currencies (euros, US dollars, etc.) or measures (kilos vs pounds, centigrade vs. Fahrenheit) or different granularities of the same measure (grams, kilos, etc.).
        - Structure: for example when the same concept is represented with an attribute In one source and with an entity in another (example entity Person is a generalization of MAN and WOMAN and in other schema entity Person has attribute GENDER)
        - Cardinality: same relation between entities but with different cardinalities.
        - Key: the same entity in different sources has different keys.
    - Conceptual Schema integration, (here we produce the Entity-Relationship global schema, solving the conflicts as we have decided in the previous phase)

4. *Describe the GAV (global as view) and LAV (local as view) approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context. Discuss the main differences between the two approaches and describe under which conditions GAV is more appropriate than LAV, and vice versa.*

A data integration system can be identified by a tuple (G,S,M) where G is the global schema, S is the source schema and M is a set of mappings (assertions) between G and S. There are two basic approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context:

- GAV (Global As View): the global schema is expressed in terms of the data source schemata. For each element g of G is defined a mapping g -> q_s as a query over the data sources. This mapping tells us exactly how the element g is computed. This approach is well suited when data sources are stable and is difficult to add new sources since the schema will need to be reconsidered.
- LAV (Local As View): in this approach the content of each source is characterized in terms of a view over the global schema G: s -› q_G This approach is applicable when the global schema is stable, for instance when it is based on a domain ontology or an enterprise model. Furthermore, this model favors extendibility but it also leads to a much more complex query processing.

GAV	*LAV*
Global database schema is expressed as the view of the local databases	Each local schema being a physical database is a materialized view of the global schema
Global schema is defined in terms of data sources	Sources are defined in terms of global schema
Global schema be derived from the integration process of the data source schemata	The global schema has been designed independently of the data source schemata
Set of assertions ONE FOR EACH ELEMENT • OK for stable data sources • Difficult to extend with a new data source	Set of assertions ONE FOR EACH ELEMENT s OF EACH SOURCE S

	• OK if the global schema is stable (based on a domain description or an enterprise model) • It favors extensibility
Mapping quality depends on how well we have compiled the sources into the global schema through the mapping	Mapping quality depends on how well we have characterized the sources
Whenever a source changes or a new one is added, the global schema needs to be reconsidered	High modularity and extensibility (if the global schema is well designed, when a source changes or is added, only its definition is to be updated)
	Not possible to obtain answers by a simple, obvious or direct computation of the query definition
	Information is in the sources, while the mapping, i.e., the view definition, specifies the opposite transformation w.r.t. the one we need!
	No simple rule or view unfolding for the relations in the body as in GAV
	A plan is a rewriting of the query as a set of queries to the sources and a prescription of how to combine their answers
	Both sources are implicitly needed: the system must extract the necessary information from SOURCE1.VERSION and from SOURCE2.PRODUCTS
	Query processing needs reasoning

5. *Describe dynamic data integration: techniques based on mediators and wrappers; techniques based on meta-models. Provide a small example.*

   For techniques based on mediators and wrappers see the previous answer. A metamodel is an abstract model for specifying concrete models. It solves the problem of mapping between different models. They are useful when dealing with semi structured data sources; in this way we can use just one model. There are two types of metamodel:
   - Metamodels in which general entities specializes into object of the specific target model. GSMM (General Semi structured Meta-Model) is a metamodel which specifies the concrete model in graph. Constraints define the semantic aspects of the concrete model
   - Entities that describe the objects of the target model. GDF (Geographical Data Files) is the standard model for the description of road networks.

6. *Describe concisely the concept of mashup and its utility in data integration, highlighting its distinctive features w.r.t. using other integration techniques.*

   Data mashups are a Web-based, lighweight form of data integration, meant to solve different problems. A mashup is an application that integrates components, which can be DB, websites, software, in a new way which can provide additional information, functions or visualization. It is one of the techniques of lightweight data integration. Integration is the added value provided by the mashup. Components may be put into communication with each other. Key elements of this application:

- A mashup component is any piece of data, logic or user interface which can be reused locally or even remotely.
- The mashup logic is the logic which specifies the invocation of components, the control flow, the data flow, the data transformations, and the UI of the mashup.

Mashups introduce integration at the presentation layer and typically focus on non-mission critical applications. If I have to integrate data for a bank I won't use a mashup because it is a superficial integration where there is no real control on what is going on and we cannot grant that everything will be ok. There are different types of mashup:

- Data mashups
- Logic mashups (merge pieces of software)
- UI mashups (merge of interface level)
- Hybrid mashups (mashup happens or more than one leyel)

7. *Describe what we mean by "lightweight approaches to data integration", using as an example the Mashup technique, and explain in which circumstances these approaches are appropriate, with advantages and disadvantages.*

We may need to integrate data from multiple sources to answer a question asked once or twice. The integration needs to be done quickly and by people without technical expertise (e.g., a disaster response situation in which reports are coming from multiple data sources in the field, and the goal is to corroborate them and quickly share them with the affected public)
Problems typical of lightweight data integration:

- locating relevant data sources
- assessing source quality
- helping the user understand the semantics
- supporting the process of integration.

Ideally, machine learning and other techniques can be used to amplify the effects of human input, through semi-supervised learning, where small amounts of human data classification, plus large amounts of additional raw (“unlabeled”) data, are used to train the system.
See previous answer referring to Mashups.
Very specific reports or ad-hoc data analyses
Simple, ad-hoc data integrations providing “situational data” that meet short term needs
Non-mission-critical integration requests

Advantage	*Disadvantage (for mashup composers)*	*Type of the Mashup*
Easy development of situational applications for power users	Mashup development is nontrivial	Data mashup Fetch data from different sources, process them and return an integrated result
Fast prototyping for developers	Luckily, mashups typically work on the “surface”	Logic mashups Integrate functionality published by logic or data components
Increased ROI for ServiceOriented Application investments	Reuse of existing components - neglecting the complexity hidden behind the service's external interface	User interface Combine the component's native UIs into an integrated UI; the components’ UIs are possibly synchronized with each other
Increased visibility by content/component providers	Composition of the outputs of (much more complex) software systems	Hybrid mashups Span multiple layers of the application stack, bringing together different

		types of components inside one and a same application;
	The work of developers can be facilitated by suitable abstractions, component technologies, development paradigms and enabling tools	

8. *Describe flow, level, and unitary measures in the data warehouse context. Provide a set of examples.*

In the context of the conceptual design phase of a Data Warehouse, we usually rely on the support of the dimensional fact model. Each fact can have zero or more associated numerical properties called measures. It is possible to identify three different categories of measures, depending on their aggregation level:

- Flow measures: they are related to a time period; at the end of the period the measures are evaluated in a cumulative way. (SUM, AVG, MIN, MAX over both temporal and non temporal hierarchies.) (Examples; number of sales in a day, total income in a month, number of birthdays in a year).
- level measures: they are evaluated in particular time instants. (AVG, MIN, MAX. but SUM only over non temporal hierarchies.) (Examples: number of products in stocks, or number of citizens in a city).
- Unitary measures: they are evaluated in particular time instants but they are relative measures. (Only AVG, MIN, MAX). (Examples: unitary price for an item in a particular instant. It cannot be aggregated with respect to time, nor category nor shop).

9. *Describe and compare the Star Schema and the Snowflake Schema used in the data warehouse context.*

Star Schema and Snowflake schema are both used in the logical design phase of a Data Warehouse. The Star Schema consists of one or more fact tables referencing any number of dimension tables. Each dimension table is characterized by a primary key d; and by a set of attributes describing the analysis dimensions with different aggregation levels. It is a simple schema, which leads to fast and simple queries. Tables are de-normalized which implies redundancy but fewer joins to perform. The Snowflake Schema reduces the de-normalization of the dimensional tables of a star schema, removing some transitive dependencies. All the attributes of a dimension table directly depend on the key, and there are zero or more external keys that allow to obtain the entire information. This schema reduces the memory space, removing data redundancy. It simplifies data update but queries are made more complex. It is possible to define views to mitigate this problem.

10. *Define what is a Data Mart in a Data Warehouse and clearly summarize the methodological steps that lead from a collection of datasets to the specification of the logical schemas of one or more Data Marts.*

A Data Mart is a structure/access pattern specific to data warehouse environments, used to retrieve client-facing data. The data mart is a subset of the data warehouse and is usually oriented to a special business area. In this context, we can distinguish between three different logical models:

- MOLAP (Multidimensional On-line Analytical Processing): data is natively stored in a multidimensional cube, in proprietary formats. Excellent performance, fast data retrieval, and are optimal for slicing and dicing operations, since all complex

calculation have been pre-generated. Unfortunately, for this reason, it is not possible to include a large amount of data in the cube itself.

- ROLAP (Relational): it is the de-facto standard, it uses the relational data model to represent multidimensional data, relies on data stored in a relational database to give the appearance of traditional LAP's slicing and dicing functionality. Can handle large amounts of data, but performance can be slow. Each ROLAP report is essentially a SQL query.
- HOLAP (Hybrid): attempts to combine the advantages of MOLAP and ROLAP. For summary type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.

The logical modelling process includes a sequence of steps that, starting from the conceptual schema (the fact schema), workload (what will be the queries? Wrt them we know what materialized views will be needed), data volume and system constraints, allows to obtain the logical schema for a specific data mart.

1. Choice of the logical schema (star/snowflake schema)
2. Conceptual schema translation into the chosen logical schema
3. Choice of the materialized views (pre-computed queries that we keep updated and ready to be presented to the user, they are defined wrt the most frequent queries. It is useful to materialize a view when: it directly solves a frequent query, it reduces the costs of some queries)
4. Optimization

11. *Define the typical operations necessary in the multidimensional data model that is at the basis of data warehouses.*

OLAP operations, useful when we store DW in the relational DB:

- Roll-up: typical operation of DW, aggregates data at a higher level - e.g. last year's sales volume per product category and per region.
- Drill-down: typical operation of DW, de-aggregates data at the lower level - e.g. for a given product category and a given region, show daily sale.
- Slice-and-Dice: applies selections and projections, which reduce data dimensionality.
- Pivoting: selects two dimensions to re-aggregate data (cube re-orientation).
- Ranking: sorts data according to predefined criteria.
- Traditional operations: select, project, join, etc.

There is only one major difference between the functionality of the ROLLUP operator and the CUBE operator. ROLLUP operator generates aggregated results for the selected columns in a hierarchical way. On the other hand, CUBE generates an aggregated result that contains all the possible combinations for the selected columns.

12. *List and describe the main dimensions of Data Quality*

- Accuracy: the extent to which data are correct, reliable and certified
- Completeness: the degree to which a given data collection includes the data describing the corresponding set of real-world objects
- Consistency: the satisfaction of semantic rules defined over a set of data items
- Timeliness: the extent to which data are sufficiently up-to-date for a task
- Validity: the degree to which data values are consistent within a defined domain
- Ethics and fairness (a new entry!).

13. *Define ontologies and their possible uses in data integration.*

An ontology is data structure that contains mainly a vocabulary and relationship between the terms of this vocabulary (such as synonimy, omonimy, hyponymy etc…)

It is defined as a formal specification of a conceptualization of a shared knowledge domain.
An ontology consists of:

- Concept
  - Generic
  - Specific (domain ontologies)
- Concept Definition
  - Via formal language
  - In natural language
- Relationships between concept
  - Taxonomies (IS_A)
  - Meronymies(PART_OF)
  - Etc…

A formal specification allows for use of a common vocabulary for automatic knowledge sharing.
An ontology captures knowledge which is common, thus over which there is a consensus (objectivity is not an issue here).
Ontologies are domain oriented and it has a grammar for using the terms to express something meaningful within a specified domain of interest.
The vocabulary is used to express queries and assertions.
(An ER diagram, a class diagram, any conceptual schema is a kind of ontology!)
Possible uses in data integration:

- An ontology as a schema integration support tool
  - Ontologies used to represent the semantics of schema elements (if the schema exists)
  - Similarities between the source ontologies guide conflict resolution
- Discovery of “equivalent” concepts (mapping), we look for some kind of similarity (most of all lexical similarity), formal representation and reasoning on these mappings
- Using an ontology as global schema instead of a relational one.
- An ontology as a support tool for content interpretation and wrapping (e.g. HTML pages)
- An ontology as a mediation support tool for content inconsistency detection and resolution (record linkage and data fusion)

Two types of ontologies:

- Taxonomic: Definition of concepts through terms, their hierarchical organization, and additional (pre-defined) relationships (synonymy,composition,…), to provide a reference vocabulary
- Descriptive: not only taxonomy but also relationship between object. Provide information for “aligning” existing data structures or to design new, specialized ontologies (domain ontologies)

Example of taxonomic ontology: Wordnet (horizontally we see synonyms while vertically we see specializations, hierarchy starts from the top, down to the finest specification).

Questions already asked in the first call:

1. *Describe, in max 10 lines, the Column-family data model in NoSQL databases, highlighting the similarities and differences w.r.t. the basic Key-Value data model*
2. *Define Wrappers and Mediators, explain in which circumstances their use is advised in Data Integration and the way they work. Discuss the various types of Wrappers and Mediators that have been introduced during the course.*

   Often the Data Integration process has to deal with multiple, distributed and heterogeneous data sources which can include semi-structured or unstructured data. For the former, data have some form of structure, but is not prescriptive, regular or complete as in traditional DBMS. The final aim is to integrate, query and compare data from multiple sources and with different structure just as if they were all structured. For this purpose, we rely on Wrappers and Mediators:
   - Wrappers: are modules binded to the sources which are in charge to translate an incoming query in one or more queries which are understandable for the specific source (in this way they can also extend its query possibilities). They also convert back query results from the source' s format to a format which is understandable for the application. Human-based maintenance of an ad-hoc wrapper is very expensive, but for some applications there are automatic wrapper generators. Example wrapper for web pages: Software module that performs an extraction step, intuition: use extraction rules which exploit the marking tags (html tags). In this example we can use automatic wrapper generation only when pages are regular to some extent, ok when many pages share the same structure and pages are dynamically generated from a DB. The wrapper is a translator that translate from one language to the other. the wrapper has to know only the language while the mediator has to know the semantic!
   - Mediators: are interfaces specialized in a certain domain which stand between application and wrappers. In one way it accepts queries written in the application's language, decomposes them, and send them to each specific wrapper. On the other hand, it is also in charge of gathering all the responses and send them back to the application, providing a unified vision of data. The term mediation has a broad meaning: it includes the processing needed to make the interfaces work, the underlying knowledge structure which guides data transformations and also any intermediate storage that is needed. The mediator has to know what are the sources, the language people and application use for querying and the semantic of the data sources. When mediator discovers new information it will accumulate it internally. The mediators are completely ad hoc. During the course we have seen the architecture of TSIMMIS project, which relies on the usage of both wrappers and mediators. In TSIMMIS, queries are posed to the mediators in a specific object-oriented language called LOREL. The data model adopted is OEM (Object Exchange Model), the way in which the mediator stores the data information (data themselves and metadata (their structure)) coming from the data sources.

Multiple choice questions

1. *A data warehouse is:* A data collection built with the aim of supporting decision-making processes
2. *The design of a data warehouse:* is driven by the queries that users will have to formulate
3. *Which of the following is the correct order for designing a data warehouse?* Operational DB schema reverse engineering → Identify facts → Attribute tree → Fact schema → Glossary → Logical design
4. *Pruning consists in:* deleting a leaf node or a leaf subtree of the attribute tree
5. *Grafting consists in:* eliminating an internal node of the attribute tree, and attaching its children to the parent of the internal node removed
6. *An aggregation operator is:* distributive if it allows to compute aggregates starting from partial aggregates without needing further information - algebraic if to compute aggregates from partial aggregates requires additional measures, which are named support measures
7. *In a data warehouse, the glossary:* defines how to compute the measures starting from the operational database
8. *When implementing a data warehouse by means of the relational model, comparing the star schema to the snowflake scheme:* the star schema is more efficient in answering queries but requires more disk space
9. *What are the differences between the GAV (global as view) and LAV (local as view) approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context?* LAV is when the sources are defined in terms of the global schema, while GAV is when the global schema is defined in terms of the sources. - In GAV the global database (schema) is expressed as a virtual view of the local databases (schemas). In LAV each local schema, being a physical database, is a materialized view over the global schema.
10. *Which of these answers describes most precisely association rule mining (e.g. distinguishing it from itemset mining)?* Association rule mining finds the frequency of relationships between data items within large data sets, expressed as if-then statements.
11. *First of all, let us establish what is the correct order for doing Data Integration:* Source schema reverse engineering → Related concept identification + Conflict analysis and resolution → Global conceptual schema design → Conceptual to logical translation of the global schema → Mapping definition
12. *Two data sources show heterogeneity of schema if:* they adopt a different logical representation of the reality of interest
13. *A data warehouse:* contains the history of the data - is subject-oriented
14. *NoSQL databases:* provide key-based access - provide flexible schemas
15. *Which of the following couples is correct?* Name conflict → synonymies and homonymies - Data semantics conflict → different currencies or units of measure
16. *Which of these answers describes most accurately the difference between classification and clustering?* Clustering is the task of grouping a set of objects in such a way that objects in the same cluster are more similar to each other (w.r.t. to a given criterion) than to those in other clusters, while classification is the problem of identifying which of a set of categories an object belongs to
17. *The KeyGen function:* can be used to solve key conflicts is likely to be used when the primary keys for corresponding tables in two different sources are integer numbers autoincremented
18. *Give some suggestions on how can ontologies be useful in data integration:* To support automatic understanding of the semantics of the instances for automatic entity resolution and

data fusion - Using an ontology as global schema instead of a relational one - To support automatic understanding of the semantics of the schemas for schema integration