

# **Data Warehousing: Multi-dimensional Data Models and OLAP**

Jose Hernandez-Orallo

Dep. of Information Systems and Computation  
Technical University of Valencia, Spain  
jorallo@dsic.upv.es

## **I N T R O D U C T I O N**

Information systems provide organizations with the necessary information to achieve their goals. Relevant information is gathered and stored in order to allow decision makers to obtain quick and elaborated reports from the data.

A data warehouse is an especially designed database that allows large amounts of historical and contextual information to be stored and accessed through complex, analytical, highly aggregated, but efficient queries. These queries capture strategic information, possibly presented in the form of reports, and support management decision making. As we will see, data warehouses differ from general, transactional databases in many ways.

## **B A C K G R O U N D**

As we have said above, the pristine motivation for building an information system was to integrate the information that would be necessary to support decision making. With the advent of databases and transactional applications, the main goal of information systems drifted towards the organization of data in such a way that software applications work. This drift triggered a specialization of database technology to serve transactional information systems, evolving database design and implementation, as well as DBMS, towards this end. The kind of work performed on these databases was called On-Line Transactional Processing (OLTP).

However, the need for decision making support did not fade away. On the contrary, more and more organizations were requiring further analytical tools to support decision making: reporting tools, EIS and other DSS tools, which many DBMS vendors began to call “business intelligence”. These tools were able to exploit the database in a more powerful way than traditional query tools. These new tools made it easier to aggregate information from many different tables, to summarize data in a much more powerful way, and to construct textual and graphical reports over them, full of condensed, graphical and statistical information. This other kind of work that was performed on the database was called On-Line Analytical Processing (OLAP).

Although the previous scenarios, working on the same physical database, can still be found in many organizations all over the world, there are three important problems associated with this approach.

- The first, and most important one, is precisely the use of the same database for both OLTP and OLAP. OLAP queries are usually complex, considering a lot of historical data and interweaving many tables with several levels of aggregations. Many OLAP queries are “killer” queries, requiring a lot of resources, and may highly disturb or even collapse the transactional work. Because of this, many reports and complex queries were run at night or during the weekends.
- Secondly, the data stored in the transactional database is just the data required by the applications. Data that is not to be used by any application is not stored. Additionally, many transactional databases do remove or simply not include historical data. On the other hand, adding all the historical data and other sources of information for OLAP work in the same transactional database can also be an important overhead storage for the transactional work.
- Thirdly, the design and organization of the database is specialized for OLTP: normalization is common, indexes are created for improving transactions, etc. However, these choices might not be good for OLAP operations. This means that even with a separate, devoted database (a replica), OLAP will not be efficient for large databases.

The previous problems stimulate the construction of separate data repositories, specialized for analytical purposes. In the early nineties, these repositories were called data warehouses and its associated technology data warehousing (Inmon, 1992). The attention widened from enterprises and vendors, but it was not until the late nineties when the academic world paid attention. All this and the appearance of more mature tools turned data warehousing into a new database discipline on its own.

## **D A T A   W A R E H O U S E S**

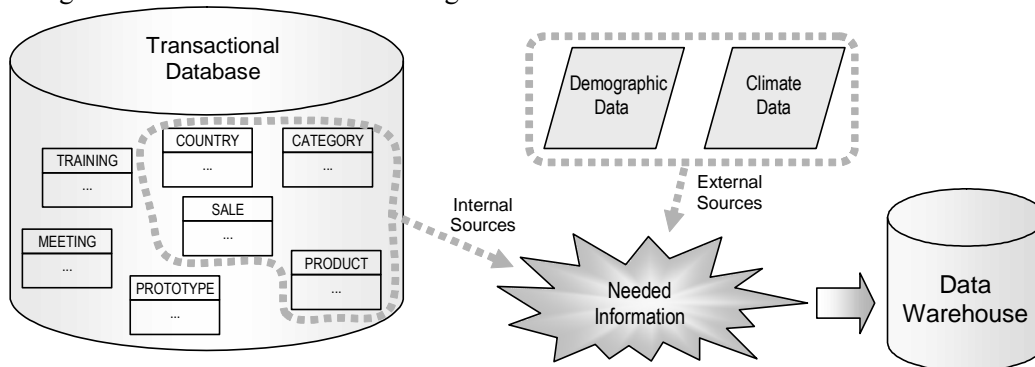
The establishment of data warehousing as a new discipline has much to do to the fact that, once we have clearly separated one database (the transactional) from the other (the historical/analytical), we see that they are quite different kind of *beasts*. Table 1 shows the strong differences between them. The special characteristics of the data and operations in the data warehouse case has led to a specialization in the data warehousing technology, establishing new designing paradigms, adopting other data models, operators and implementation tools.

Having seen the advantages of constructing a separate data repository, the first question is to determine the data that will be included in the repository. This will surely depend on the analysis of the data requirements for the business intelligence applications that will use the data warehouse, such as the OLAP tools, the DSS systems and, possibly, data mining. Frequently, this means that part of the information existing in the transactional databases (but not all and not at the same level of detail) has to be loaded into the data warehouse, as well as any external data that could be useful for the analytical processing. Figure 1 illustrates this integration.

Table 1: Differences between Transactional Databases and Data Warehouses

	Transactional Database	Data Warehouse
Purpose	Daily operations. Support to the software applications.	Information retrieval, reports, data analysis.
Data Characteristics	Data about the organization inner working, changing data, internal data, incomplete data.	Historical data, internal and external data, descriptive data.
Data Models	Relational, object-relational, normalized.	Multidimensional, snowflake, partially denormalized.
Users	Hundreds/thousands: applications, operators, administrator, ...	Dozens: managers, executives, analysts (farmers and explorers).
Access	SQL. Read and write.	SQL and specific OLAP operators (slice & dice, roll, pivot, ...). Read-only.

Figure 1. Data warehouses integrate data from internal and external sources.



Integrating data from several sources, with different formats, data models and metadata is not an easy task. Integration from different sources has always a negative effect on data quality; lots of missing data are created, because some features exist in some sources but not in others, as well as incompatible formats, units, etc. It is extremely important that data is transformed into a unified format and schema, cleansed from missing and anomalous data. We will get back on these issues, but next we have to address another crucial issue, the way in which information is organized in a data warehouse.

### Data Warehouse Organization: The Multidimensional Model

A first idea for constructing a data warehouse can be to apply a traditional design, using an entity-relationship or UML-alike conceptual model, and transform it into a relational or an object-relational database, integrating all the sources of information into one highly normalized off-the-shelf database. However, this has been shown to be a bad way for organizing data in a data warehouse. The reason-why is that classical data models, and especially the relational model, are well suited for performing transactional work but do not deal well with complex queries with a high degree of aggregation.

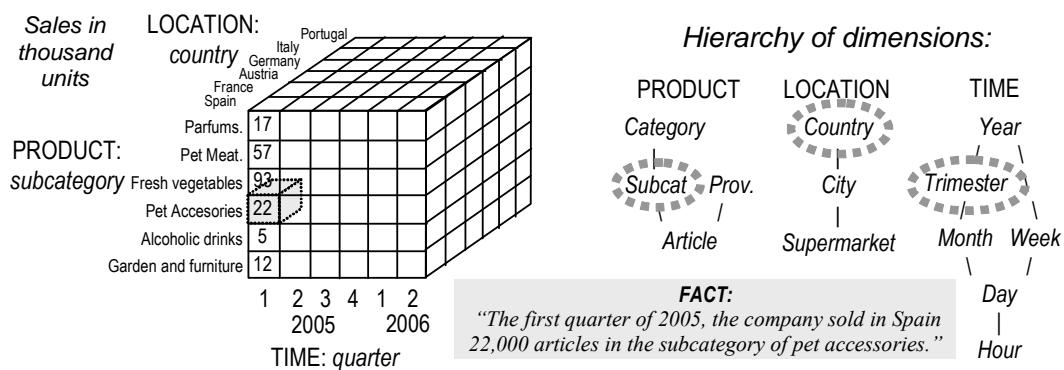
As a consequence, the most widespread model for data warehouses is a different data model: the multidimensional model (Golfarelli, Maio & Rizzi, 1998). Data under the multidimensional model is organized around *facts*, which have related *measures* and can be seen in more or less detail according to certain *dimensions*. As an example, let us consider a multinational European supermarket chain. Its basic facts are the sales, which can have several associated measures: total income amount, quantity, number of customers, etc., and can be detailed in many dimensions: the time of purchase, the product sold, the place of the sale, etc. It is enlightening to see that measures usually correspond to the question “how much / how many” whereas the dimensions correspond to the questions “when”, “what”, “where”, etc.

The interesting thing about the dimensional model is that it eases the construction of queries about facts at several levels of aggregation. For instance, the fact “the sales summed up 22,000 products in the subcategory ‘pet accessories’ in all the supermarkets in Spain in the first quarter of 2005” represents a measure (quantity: 22,000) of a sale with granularity month for the time dimension (first quarter of 2005), with granularity country for the location dimension (Spain) and with granularity subcategory for the product dimension (‘pet accessories’).

The structure of possible aggregations on each dimension constructs a hierarchy, from which we can select any granularity. For instance, as shown in Figure 2, the dimension time is hierarchical, existing several paths from the most fine-grained resolution (hour) to the most coarse-grained resolution (year). The kind of hierarchies that we may have gives several names to the whole: simple star, hierarchical star or snowflake. The example in Figure 2 is a snowflake, since there are many levels in each dimension and there are alternative paths.

The previous dimension *aggregations* do not fix the condition in each dimension, they just settle the “datacube” resolution. As we can see in Figure 2, a datacube just specifies at which level of resolution the selection can be made. In the example of Figure 2, we have a datacube with resolution country for the location dimension, subcategory for the product dimension and quarter for the time dimension. It is in this datacube where we can make a *selection* for a specific location (Spain), product (pet accessories) and time (first quarter of 2005).

Figure 2. A fact shown in a *datacube* selected from a hierarchy of dimensions.



Finally, for many data warehouses it is not possible to organize all the information around one fact and several dimensions. For instance, a company might have one kind of facts and dimensions for sales, another for invoicing, another for

personnel, etc. The solution is to create similar substructures for each of these areas. Each substructure is called a datamart. A data warehouse under the multidimensional model is then a collection of datamarts.

### Exploitation: OLAP Operators and Tools

A data model comprises a set of data structures and a set of operators over these structures. In the previous section we have seen that the data structures in the dimensional model are the facts, with their measures, and the dimensions, with their hierarchy and attributes for each level. We have seen that a single “operator” can be defined by just choosing a measure from the fact and a level for each dimension, forming a datacube, and then, selecting the values of one or more dimensions.

Query tools on data warehouses under the multidimensional model usually have a graphical interface, which allow the user to select the datamart, to pick one or more measures for the facts (the aggregated projection), to choose the resolution in the dimension hierarchies and to express additional conditions.

As we have discussed above, we can have more than three dimensions. This would prevent the graphical representation of datacubes in many situations. On the contrary, the relational model (as well as other classical models) has the advantage that the result of a query is a table, which can always be represented on a screen or a paper. The typical solution taken by the query tools for the dimensional model is a hybrid between tables and cubes. For instance, the cube of Figure 2 can be represented in two dimensions as shown in Table 2.

Table 2: A Hybrid DataCube/Table Output of a Multidimensional Query							
		2005-1	2005-2	2005-3	2005-4	2006-1	2006-2
<b>Spain</b>	<b>Parfums</b>	17000	12000	3000	11000	18000	12000
	<b>Pet Meat</b>	57000	52000	53000	56000	54000	54000
	<b>Fresh Vegetables</b>	93000	125000	131000	103000	97000	133000
	<b>Pet Accessories</b>	22000	32000	23000	19000	24000	35000
	<b>Alcoholic Drinks</b>	5000	4200	3800	7300	5300	4100
	<b>Garden &amp; Furniture</b>	12000	18000	19000	15000	14000	18000
<b>France</b>	<b>Parfums</b>	23000	17000	5000	17000	17000	13000
	<b>Pet Meat</b>	77000	63000	72000	82000	64000	53000
	<b>Fresh Vegetables</b>	102000	132000	126000	110000	99000	143000
	...	...	...	...	...	...	...
	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

The idea is that two or more dimensions can be “dragged” to the left or to the top, and we can have an arbitrary number of dimensions in a bidimensional table. This is feasible precisely because we are obtaining highly summarized (aggregated) information and the number of rows and columns is manageable. These hybrid tables are usually called “reports” in OLAP tools.

With the multidimensional model and this hybrid datacube/table representation in mind it is easier to understand some additional operators that are more and more common in OLAP tools, called OLAP operators:

- Drill: de-aggregates the data (more fine-grained data) following the paths of one or more dimensions.
- Roll: aggregates the data (more coarse-grained data) following the paths of one or more dimensions.
- Slice & Dice: selects and projects the data into one of both sides of the report.
- Pivot: changes one dimension from one side of the report to the other (rows by columns).

The most important trait of these operators is that they are “query modifiers”, i.e. they are used to “refine” the queries. The design of data warehouse and OLAP tools are intended to avoid the recalculation of the aggregates each time one of the previous operators is applied. The idea is to pre-calculate the aggregations at several levels of the hierarchies, so making the traversal through the dimension paths more efficient.

## **Design and Implementation**

One of the reasons to create a separate data repository for data analysis was to improve efficiency. The multidimensional model and the OLAP refinement operators are two powerful means towards this goal, but must be complemented by an adequate data warehouse design and implementation.

There are, of course, many design methodologies for data warehouses (see e.g. Gardner, 1998, Inmon, 2002, Kimball, 1996, Sapia, Blaschka, Höfling & Dinter 1996, Trujillo, Palomar & Gómez, 2001; Tryfona, Busborg & Christiansen, 1999). Some of them differ significantly from the rest, using entity-relationship extensions, UML extensions, or specific modeling paradigms. There are also international standards such as the OMG Common Warehouse Metadata (CWM) for this purpose.

We just give some hints on the most important steps for designing a data warehouse (or more precisely each data mart), under the multidimensional model:

- Choose a “process” or “domain” in the organization for which analytical processing is to be done, considering the data that complex reports, more general analysis or data mining will require.
- Choose the central fact and the granularity. Granularity should be fine-grained enough to allow every interesting report and query to be done, but not more than that.
- Identify the dimensions which characterize the domain, its levels and hierarchies, as well as the basic attributes for each level. Dimensions can vary from one domain to the other, although they usually correspond to questions such as “what”, “who”, “where”, “when”, “how”, etc.
- Determine and refine the measures for the facts (usually corresponding to questions such as “how many” or “how much”) and the attributes for the dimensions.

Once the data warehouse is designed, it is time to start its implementation (Weir, Peng & Kerridge, 2003). The first important decision for this is to determine the kind of physical environment that will hold the information. There are two main approaches:

- ROLAP (Relational OLAP): physically, the data warehouse will be implemented with a relational database.
- MOLAP (Multidimensional OLAP): physically, the data warehouse is constructed over special structures based on multidimensional matrices.

It can be argued that the MOLAP approach should be more adequate for a data warehouse which uses a multidimensional model. However, there are pros and cons for both approaches. ROLAP has several advantages: we have many RDBMS to choose from, with their typical tools: SQL, constraints, triggers, procedures, etc.; the training and cost required for the implementation of a ROLAP data warehouse is usually lower. MOLAP has a more direct correspondence between the logical level and the physical level and it is usually more efficient.

At least in the inception of data warehouses, the ROLAP approach has been more successful. There are many methodologies to convert a multidimensional design into a relational database that implements it. The *starflake* structure (Kimball, 1996) is a well-known approach and is based on a central *fact table* with foreign keys to the base (fine-grained) dimension tables, which, in turn, refer to other more coarse-grained dimension tables. These tables are called *snowflake tables*. Some additional tables (*star tables*) are added (redundantly) to improve efficiency.

Associated to the physical level, new technologies have been developed to make the data warehouse work effectively: tools to compute and instantiate the datacube, query optimization to select and refine queries on the datacube, specialized indexing techniques for improving aggregation, and many others.

### **Data Load and Refreshment**

Finally, there is an important component to be considered for the implementation of a data warehouse. Once we know the data that we want to integrate in the data warehouse (either internal or external), we have designed the schema that will be used to organize all this data, decided the implementation over ROLAP or MOLAP and created the physical schema, we can start with the hard and true implementation work. But what's left? We have to extract and transform the data from the original sources (and from several schemas) and to load them into the data warehouse (into the new schema).

Although this process may resemble some data migration processes, which are common when one or more databases have to be moved into a new and different system, there are some specific issues which make this process even more complicated. The volume of data transferred is usually very large, the data has to be transformed deeply and comes from many different sources, some original databases cannot be disturbed by this process since they are necessary for the daily transactional work, and, finally, the data has to be refreshed periodically, i.e., the data warehouse must be kept up-to-date (e.g. weekly or monthly) with the new data added to the transactional sources.

The complexity of this process and the fact that it must be operating periodically motivates the construction of a *system*, which is called the ETL (Extraction, Transformation, Load) system.

The ETL system must perform many tasks:

- Extraction of the transactional data: this must be done in such a way that the transactional database is not disturbed (e.g. at night).

- Incorporation of external data: since the data might be unstructured, it is necessary to use wrappers or scripts to convert data from text, spreadsheets, HTML or other formats into the form required for the data warehouse.
- Key creation and metadata creation: the data moved must be assigned new keys. The correspondence between the original data and the data in the warehouse must be traceable through the use of metadata.
- Data integration, cleansing, transformation and aggregation: the data must be crossed, compared and integrated, and most of the data is needed in some level of aggregation. It is not convenient to perform all this on the transactional databases, since this can slow the transactional work.
- Change identification: the system must deal about how changes are refreshed from the sources to the data warehouse (by delta files, time stamping, triggers, log files or mixed techniques).
- Load planning and maintenance: definition of the order and stages for the load process, load time windows (to minimize impact on the transactional sources), creation of indexes on the data warehouse, ... The maintenance must include periodical assessments on the quality of the data stored in the data warehouse.

Many ETL systems use an intermediate repository between the sources and the data warehouse, in order to be able to do aggregations, integration and cleansing, as well as other processes (metadata) without disturbing the original transactional databases or the destination data warehouse.

## **F U T U R E   T R E N D S**

Data warehousing technology is evolving quickly and there are many research opportunities (Widom, 1995, Roddick, 1998, Samtani, Kumar & Kambayashi, 1998, Dinter, Sapia, Hölfing & Blaschka, 1999, Rizzi, 2003, Rossopoulos, 1998) in many several topics, such as data models, logical and physical design, implementation (the construction of the ETL system in particular), metadata standardization, theoretical issues such as materialized views, schema versioning and evolution, as well as other more heterogeneous topics such as data cleansing and transformation, XML metadata, OLAP query languages, the operability and friendliness of OLAP tools and their distribution between client and server.

## **C O N C L U S I O N**

Data warehousing is an emerging technology that has spread over organizations and companies all over the world, because it is making OLAP possible, i.e. a real-time analytical processing for volumes of data for which it was simply unbelievable a decade ago. The use of separate data repositories, called data warehouses, jointly with the organization under specific data models and specific physical implementation, as well as new OLAP tools and operators have boosted the friendliness and effectiveness of reporting and other analytical tools. Data warehouses are being used as platforms not only for query and report tools with an orientation to “farming” reports, but they make a very good team with more “exploring” objectives, such as summarization and data mining. Data mining is discussed in another entry in this encyclopedia.



## REFERENCES

Dinter, B., Sapia, C. Höfling, G., & Blaschka, M. (1999) OLAP market and research: initiating the cooperation, *Journal of Computer Science and Information Management*, 2(3), 1999

Gardner, S.R. (1998) Building the data warehouse, *Communications of the ACM*, 41(9), pp. 52-60, 1998.

Golfarelli, M., Maio, D., & Rizzi, S. (1998) Dimensional fact model, *International Journal of Human-Computer Studies*, 43(5,6), pp. 865-889. 1998.

Inmon, W.H. (1992) EIS and the data warehouse: A simple approach to building an effective foundation for EIS, *Database Programming and Design*, 5(11):70-73, 1992.

Inmon, W.H. (2002) *Building the Data Warehouse*, 3rd Edition, Chichester, Wiley 2002.

Kimball, R. (1996) *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Datawarehouses*, New York, Wiley 1996.

Rizzi, S. (2003) Open problems in data warehousing: eight years later, (Keynote slides) in H-J. Lenz, P. Vassiliadis, M. Jeusfeld, & M. Staudt (Eds.), *Design and Management of Data Warehouses 2003*, Proceedings of the 5th Intl. Workshop DMDW'2003, Berlin, Germany, September 8, 2003, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-77/keynote.pdf>

Roddick, J.F. (1998) Data Warehousing and Data Mining: Are we working on the right things? *Advances in Database Technologies*. Berlin, Springer-Verlag. Lecture Notes in Computer Science. 1552. Kambayashi, Y., Lee, D. K., Lim, E.-P., Masunaga, Y., & Mohania, M., Eds. 141-144, 1998.

Rossopoulos, N. (1998) Materialized views and Data Warehouses, *SIGMOD Record* 27(1), pp. 21-26, 1998.

Samtani, S., Kumar, V., & Kambayashi, Y. (1998) Recent advances and research problems in data warehousing, *International Conference on Conceptual Modeling (ER)*, LNCS 1507, Springer 1998.

Sapia, C., Blaschka, M., Höfling, G., & Dinter, B. (1996) Extending the E/R model for the multidimensional paradigm, *International Workshop on Data Warehouse and Data Mining (DWDM)*. 1999.

Trujillo, J.C., Palomar, M., & Gómez, J. (2001) Designing data warehouses with OO conceptual models, *IEEE Computer*, 34(12). pp. 66-75. 2001.

Tryfona, N., Busborg, F., & Christiansen, J. (1999) Star ER: a conceptual model for data warehouse design, *International Workshop on Data Warehousing and OLAP (DOLAP99)*. 1999.

Weir, R., Peng, T., & Kerridge, J. (2003) Best practice for implementing a data warehouse: a review for strategic alignment, in H-J. Lenz, P. Vassiliadis, M. Jeusfeld, & M. Staudt (Eds.), *Design and Management of Data Warehouses 2003*, Proceedings of the 5th Intl. Workshop DMDW'2003, Berlin, Germany, September 8, 2003.

Widom, J. (1995) Research problems in data warehousing, *International Conference on Information and Knowledge Management (CIKM95)*, ACM Press. 1995

## Terms and Definitions

**Farmers and Explorers:** Two typologies of data warehouse users. Farmers are users of data warehouses and other analytical tools which generate periodical reports, such as sales by week, category and department. Explorers are more ambitious users of data warehouses which try to better understand the data, to look for patterns in the data or to generate new reports.

**Data Cube:** A pre-selection of aggregated data at several levels of detail according to several dimensions in a data mart. A datacube establishes the resolution and basic projection where selections can be made. The datacube aggregation detail can be modified through OLAP operators such as drill and roll, and their dimensions can be modified by slice & dice and pivot.

**Data Mart:** Part of a data warehouse which gathers the information about a specific domain. Each data mart is usually viewed at the conceptual model as a multidimensional star or snowflake schema. Although each data mart is specialized on part of the organization information, some dimensions can be redundantly replicated in several data marts. For instance, time is usually a dimension shared by all data marts.

**OLTP (On-Line Transactional Processing):** The usual process in a transactional database, which consists of frequent queries and updates for serving the applications of an organization. This kind of processing supports the daily operations of the software applications, it is read-and-write, and generally performed through SQL queries.

**OLAP (On-Line Analytical Processing):** The process of analyzing a database or data warehouse, which consists of *heavy* queries for constructing reports or showing the information in a highly aggregated or complex way. This kind of processing supports information retrieval and data analysis in the form of complex queries and reports.

**ROLAP (Relational OLAP):** Implementation of a data warehouse and associated OLAP tools over a relational database and RDBMS. This approach must be based on an appropriate mapping between the multidimensional conceptual schema and the relational logical/physical schema. The great advantage of this approach is that inherits the existing technology and knowledge on classical relational database systems.

**MOLAP (Multidimensional OLAP):** Implementation of a data warehouse and associated OLAP tools over specific multidimensional data structures. This approach has a more direct mapping between the multidimensional conceptual schema and the multidimensional physical schema.

**Snowflake schema:** A schema for organizing multidimensional data, where we have many levels in each dimension and there can be alternative paths for aggregation. This schema is the most usual for designing data marts.

**ETL (Extraction, Transformation, Load) system:** The system which is in charge of extracting the data from internal transactional databases and other sources (such as external data), transforming it to accommodate to the data warehouse schema, loading the data initially and refreshing the data periodically. The design of the ETL system is generally the most time-consuming task in the construction of a data warehouse.