# USING SCHEMA AND DATA INTEGRATION TECHNIQUE TO INTEGRATE SPATIAL AND NON-SPATIAL DATA : DEVELOPING POPULATED PLACES DB OF TURKEY (PPDB_T)

Abdulvahit Torun

General Command of Mapping (GCM), Cartography Department, 06100 Cebeci, Ankara, Turkey – atorun@hgk.mil.tr

**KEY WORDS:** Spatial, Databases, Integration, Schema Integration

**ABSTRACT:**
Databases (DBs) are designed from scratch due to application specifications of organizations. Sometimes the same real world object occurs in different semantics in different DBs. DBs may be autonomous on different computers and computers may have different operating systems, DBs may be designed for different purposes and applications and DBs may use different data formats. In order to use legacy data instead of re-collecting the same content, data in a wide variety databases are needed to be integrated. Data integration is done either gathering data without any integration or by integrating several source data partly or as a whole into a single DB. The method, integrating several DBs into one single DB, is implemented while developing Populated Places DB of Turkey (PPDB_T). To implement the integrated DB, information about populated places except population was collected from a plain table, population information are extracted from State Statistics Institute (SSI) DB and administrative boundaries are taken from Digital Chart of Turkey (DCT) at scale 1:1000000. Ladder technique is applied for schema and data integration. Although the data is stored in and managed by a non-spatial DB system, in order to improve geometric accuracy of location information, a subset of PPDB_T is transformed into a spatial DB and location information are corrected with the help of standard topographic map index and administrative boundaries of provinces and districts. This need forces mapping a subset of the non-spatial DB containing location information into spatial world and integrate it with DCT to do corrections and transfer the corrected location and information back to PPDB_T.

## 1. INTRODUCTION

Organizations have a large variety of databases to conduct their everyday business. Large organizations have their databases on different platforms. Usually, databases are designed from scratch due to application specifications of organizations. Sometimes, the same real world object occurs in different semantics in different DBs. Some users store very needed data independently which are existing in the database of organization (Elmagarmid et.al., 1999).

As organizations have become sophisticated, data management and data sharing become a mess. With the enabling technology for networking, distributed computing and communication, data and process sharing become applicable. Connecting and sharing sources among existing DBs forces to deal with autonomy and heterogeneity. These DBs may be autonomous on different computers and computers may have different operating systems, may be designed for different purposes and applications and may use different data formats.

DBMSs and applications are designed considering incorporate data and process sharing to ease autonomy. Main efforts are spent in three directions; ‘schema integration’, ‘federated DB’ and ‘multi-database language’ (Elmagarmid et.al. 1999, et.al. Ozsu et.al. 1999). Schema integration is defined as integrating component DBs at schema level. Schema integration is translation of a schema from a data model into another data model manually, automatic or with both techniques. Although, federated DB approach allows more autonomy and flexibility in heterogeneity, partial schema integration is done repeatedly on-the-fly and the user needs master help. Multi-DB language approach favors autonomy over heterogeneity such that a common multi-DB language that enables accessing and manipulating disparate DBs (Elmagarmid et.al., 1999).

PPDB_T is a non-spatial relational DB containing information about populated places. The relationships among entities resemble the hierarchical relationship among populated places in the real world. In addition to conventional use of databases in different applications, PPDB_T is usually used for topographic map production by GCM, Turkey. Although, thematic correction is finished, geometric enhancement of populated places is being continued. PPDB_T is being designed as a central DB. Integration with State Statistics Institute (SSI) DB and other spatial DBs are accomplished by developing programs in Application Programming Interface (API) of Microsoft Access, ESRI Arc/Info and ArcView coding languages. In the following chapters, concepts on database integration and DB integration strategy applied for PPDB_T are given.

## 2. DATABASE INTEGRATION

A distributed DB is developed from scratch by using a top-down methodology. However, in most of the cases the tendency is management of legacy systems and re-use of data in recent years. The DBs, which are distributed geographically, in different models, in different environment and with different semantics are needed to be integrated to use the available data. This problem is called as ‘DB integration’. This type of problem yields more difficulty compared to distributed DBs design. DB integration is a bottom-up work contrary to distributed DB design, which is a top down work.

The problems caused by semantic heterogeneity and design autonomy (heterogeneity) should be resolved to achieve DB integration. Semantic heterogeneity is defined as perceiving the same real world phenomena differently with respect to different viewpoints of applications. Design autonomy is a matter of schema and syntax of DBs. Schema defines how the features are expressed within the DB. The discrepancy within the type of DBMS, API used in development and the data model that the DBMS uses to express the features defines syntax autonomy.

Database integration is the process by which information from participating (component) databases are conceptually integrated to form a single cohesive multi-DB -in other words Global Conceptual Schema (GCS)-. Therefore, DB integration is unifying existing data stores into a single framework (Figure 1).

Here are some concepts, which are frequently used in database integration to make the discussion clear. Local Internal Schema (LIS) is the schema of individual internal schema of DB. Local Conceptual Schema (LCS) is logical organization of data at each site. External Schema (ES) is the schema for a user application. Global Conceptual Schema (GCS) is enterprise view of the data-logical structure of the data at all the sites. Federated Conceptual Schema (FCS) is combination of a set of export schemas of individual DBs. Data dictionary is information (or a DB) about the DB schema, relations, attributes, domains of attributes, relationships of a DB. A data dictionary can be implemented as a metadata. GES, LES stand for Global External Schema and Local External Schema respectively (Figure 1). LES and GES are the views for local and external users respectively (Bobak 1996, Ozsu et.al. 1999).

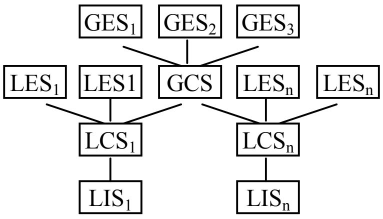

Figure 1. Schema Hierarchy in Database Integration and relationship within local and global schemas

## 2.1 Database Integration Strategies

DB integration implies uniform and transparent access to data managed by multiple DBs. The mechanism is based on an integrated schema which involves all or part of the schema of component DBs. Heterogeneous distributed DBs are integrated and re-designed as a whole by means of global schema integration, federated DBs and multi DB language approaches.

There are three types of DB integration strategies; firstly integrating the data, secondly schema integration and thirdly developing software (Devogele et.al. 1998).

### Data Integration

A very basic approach is providing a global catalogue of accessible information sources to user to allow him to do integration by himself without attempting any integration. Each data source may be described as a metadata containing information about representation model, scale, last update date, data quality etc. Most of the data warehouses on the web are working in this way. The user should search, find, decide and find tools to select, extract, integrate and query the data.

A further step is integrating the source DBs as one single DB by putting the data together. This approach violates the basic rules of distributed DBs such as data distribution, data autonomy (independence). The data and applications should be converted into the integrated DB. Integration is accomplished in two ways; either integrating existing data by hand or extracting relevant data from data sources in suitable form to be integrated centrally.

For the former method, the application is split into small pieces each of which accesses only a single data store. Then, the local data is carried to the site of global application. For the latter method, data is extracted from the data source to make a single DB such as EuroGlobalMap (former MEGRIN project) or SABE of EuroGeographics. DB integration based on data integration is lack of scalebility. Every time DB is enlarged, the same integration processes should be followed from the beginning. Moreover, it yields duplication. The same data resides at its local DB and at the integrated DB. Therefore, maintaining the consistency becomes difficult.

### Schema Integration

Source DBs can be integrated by means of schema integration or using standardized data or schema models. Schema integration is a practical method such that the data is integrated as a logical DB with a global schema. With a given set of local DB schemas belonging to an individual DBMS, an integrated schema which subsumes those local schemas is created by synthesizing the schemas. There are difficulties in schema integration, because of different structures and semantics among local DBs. The user is not aware of the underlying systems, the whole system imitates as if one single DB. Those integrated DBs of this type are known as multi-DBs.

Standardization of data and models provides a base for integration. Standardization of data modeling and manipulation features enables exchange of these data and methods among heterogeneous systems.

Data model standards define which abstract modeling concepts that are to be used for modeling the real world. For non-spatial data, there are standards for relational and Object Oriented data models. Several spatial data modeling standards are defined by ISO (TC 211), by CEN(TC 287) and OpenGIS. Each of these standards provides a view for different aspect of data.

There are schema standards consisting of data and process design for specific application areas, which are defined by OpenGIS. These standards define a common target (canonical model) for data conversion and interoperability. But they don’t define how to convert existing data into standards and how to integrate data.

These standard data and design models provide a common understanding to squeeze the interoperability – schema, format and semantics- problem into semantic problems.

### Software Developing

Software can be developed to support data interoperability among diverse DBs. The software may work in three different ways; as a gateway, as a persistent view and as a tool to construct a Federated DB. A gateway interface connects different DBs and allow accessing data in other systems. Although, defining a persistent view over several DBs allow access to distant data, consistency constraints can not be defined. Connection to distributed data is limited with the definition in the view. Federated DB allows combining and scaleable integration of distributed data. Site autonomy is not violated within a federated DB.

Federated (virtual) DB is defined by creating a schema representing component DBs. However, data instance is at its original local site and is created temporarily on-the-fly when required. Local DB administrators specify a subset of DB for FDB users. Import/export into/from FDB is managed by federated system on the basis of standard data model and manipulation language.

Federated Schema is a less loosely coupled multi-DB through a federated schema. There is no global schema that comprises component schemas. A user defines his own schema based on the export schemas of component DBs that are allowed for sharing. The Component DBMSs may have different models creating heterogeneity, which must be solved during integration. Federated schema is a union of export schemas of Component DBs, $Federated\ \ Schema\ = \bigcup ES_{Component\ \ DB}$

Gateway and View approach provide the user with a multi-DB access language (SQL) without any unification of the semantics of data from various sources. Federated DB (FDB) is an integrated view of data which is managed by the FDB. Users access FDB like a centralized DB without worrying the localization of data or without worrying about the type of DBMS.

### 2.2 Integration Methodology for PPDB_T

A hybrid integration technique is applied in order to develop Populated Places DB of Turkey (PPDB_T) (Torun, 2000). Data integration method –integrating several DBs into one unique DB- and schema integration method are used for integration. Data insertion from the component DBs is done by running the programs written in API of PPDB_T. Schema and Data integration is accomplished by using ladder technique in which first of all, schema of newly designed PPDB_T of GCM is integrated with the data collected for populated places which are stored in a plain table. Then, schema of PPDB_T is extended to integrate it with SSI DB schema.

For integrating different data sets describing the same phenomena, first a correct understanding of the semantics of the existing data should be developed. Schemas are transformed into a common data model. For instance SSI DB schema and DB instance are converted into SQL and dbf respectively. Then, an accurate correlation among structures (schemas) is established to avoid comparing different type of objects within the same category. The inter-schema correspondence at meta level and inter-DB correspondence at data level are identified and described. Finally, integration is described precisely to prevent merging irrelevant data together. The conflicts are solved semi-automatically. Integrated schema is generated on top of contributing data sources (Figure 2). Schema integration is done by using modified 5 level integration architecture for the purpose of single DB generation (Abel et.al. 1998, Ozsu et.al. 1999).

**Modified 5 Level Schema Integration Architecture**

1. Local Schema: Local schema of plain table, SSI DB and DCT are made available by the Component DBs.

2. Component Schema: Representation of Local schema in canonical (standard) data model is defined by using SQL. Canonical data model is employed for unifying divergent local schemas in a single schema.

3. Export Schema: Export Schemas are defined from component schemas based on integrated global schema of PPDB_T.

4. Schema Integration: Export Schemas are integrated to form a single schema. Since the desired final DB is not a federated one, the integrated schema is not a Federated Schema, either. Firstly, two component schemas are integrated. Then, the third one is added to the integrated schema. This method is called as ladder technique. Data integration follows schema integration (Figure 3).

5. External Schema: Upon the integrated schema different conceptual/external schemas are defined for different purposes and usage.

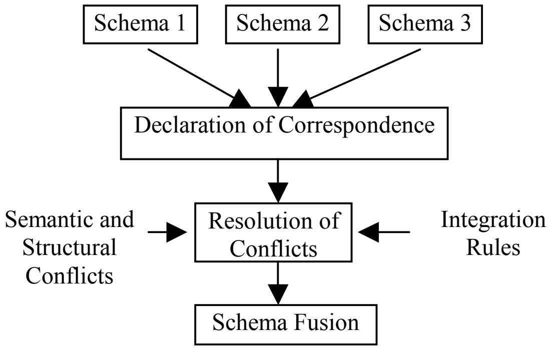

Figure 2: Integration methodology for constructing PPDB_T (After Devogele et.al, 1998)

## 3. DEVELOPING POPULATED PLACES DATABASE OF TURKEY (PPDB_T)

Information about populated places is used in topographic map production for typography and feature placement in GCM. Population information and relevant statistics are collected and stored by SSI. In order to maintain population information up-to-date for map production, DB integration is necessary to share the legacy data by SSI instead of re-constructing the same content. Content, definition and thematic granularity and accuracy of both non-spatial DBs are considering the administrative hierarchy of Turkey (Torun 2000).

PPDB_T contains information about inhabited places such as its name, its former names, its administrative status, its hierarchy within administrative system, location information of its center, its population etc.

Data is gathered from three different sources. Initially, the information about populated places except population was collected and stored in a plain table on computer. Population information is extracted from State Statistics Institute (SSI) DB.

Finally, administrative boundaries are taken from Digital Chart of Turkey (DCT) at scale 1:1000000.

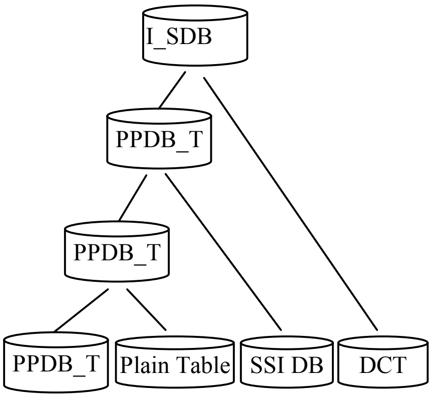

Figure 3: PPDB_T is developed by using ladder method with three steps

### 3.1 Defining Associations

Turkey has a hierarchical administrative system, which looks like a balanced tree structure. The sequence of residential entities (populated places) from top to bottom is province, district, sub-district, village and suburb respectively. Data about populated places that are being on topographic map series of Turkey at scale 1:25 000, are maintained in a conventional data store –a paper form for each populated place that is clustered for each province- in Cartography Department. Some of the information collected for a populated place is a unique ID number, nationally authorized name of the place, location (title of the 1:25 000 map sheet and grid numbers), population, old names of the place and height of the center of the town information. Initially, information on the forms is transferred in a plain table on computer (Torun 2002).

Population and population growth data are organized as a spreadsheet by SSI. The provinces are ordered due to plate number sequence. Rest of the populated places are ordered considering alphabetic order at each level within its parent unit. Cumulative data are added into the list representing for higher level populated places. Suburbs, which are not considered as a unit, are not existing in SSI DB.

### 3.2 Removing Schematic and Semantic Conflicts

Organization and hierarchy of populated places in data stores of both GCM and SSI are almost the same. Main difference comes out of schema and format. SSI DB is modified and DB schema is enhanced in order to prepare the data for integration. Data is transferred into a common schema and format (dbf format) for which a middleware is developed to transfer population data (Figure 4).

Semantic heterogeneity is resolved by using single semantics and preservation of data consistency and integration is provided by resolution of naming conflicts. With schema integration, the problems of name conflicts, schematic differences, missing data are resolved. A multi-word name in one DB might be a combined word in the other or different names –old and new names or multiple names for a place- are assigned for the same place. These are partially resolved with computer programs. A populated place might be clustered in a false class in one of the DBs. After resolving those conflicts given above, two data sources are integrated considering integrated PPDB_T schema.

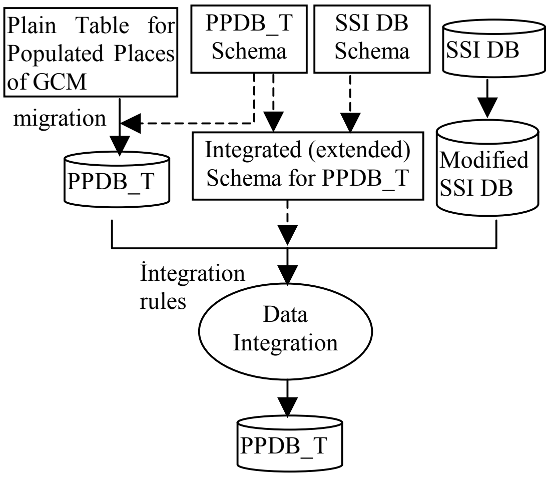

Figure 4: Integration procedures to construct PPDB_T

### 3.3 Database Integration to build PPDB_T

**Schema Integration**

First of all, database schema of PPDB_T is designed by means of actual and future needs. DB schema of PPDB_T is designed considering the administrative binding of populated places. Then, relevant attributes of SSI DB are added to the PPDB_T schema in addition to a foreign key, which provides PPDB_T to connect SSI DB.

A subset of PPDB_T is exported to generate a spatial DB consisting of points of populated places. This export schema is integrated with DCT schema to do spatial analysis and to do corrections on spatial and non-spatial attributes of populated places. When correction is finished, information in the spatial DB is transferred back to PPDB_T for improvement. There happens no schema integration.

Schemas of two DBs are different. DB schema of GCM for Populated Places is designed by using relational model. However, Population DB of SSI is a table containing all kind of quantitative values and statistics. DB schema of SSI is migrated into schema of PPDB_T of GCM. This process prevents scalebility of the resultant schema and DB instance. Each time there happens a change, the integrated schema and integrated DB should be updated.

Data for spatial DB is extracted from PPDB_T in a common model for both DBMS and GIS software –ESRI Arc/Info- to transfer data back and forth. These are done by developing a tiny software which extracts data from PPDB_T into a common model, transforms the common model into a spatial DB and visa-versa. Different languages are employed for spatial and non-spatial definitions and manipulations.

**Data Integration**

Data integration process is done in four major steps. Data integration is accomplished by applying ladder technique. Firstly, DB schema of PPDB_T is designed considering the available non-spatial digital information sources and further needs. PPDB_T is a non-spatial relational DB containing location information, which is used to produce spatial data. Location of a populated place is represented at the center of gravity of the settlement area. Secondly, PPDB_T is populated with the data from plain table. Therefore, the plain table is mapped into PPDB_T. Thirdly, PPDB_T schema is extended in order to import population information from exported data of SSI DB.

Each populated place in SSI DB is searched in PPDB_T by tracing the path through the hierarchical administrative path by using type of the feature. As the populated place is located, value of population attribute is updated and a unique ID value generated from the traced path for the populated place in SSI DB is stored as a foreign key. If the populated place is not met in DB, user resolves the conflict.

The new spatial DB namely derived Spatial DB (d_SDB) is generated by means of a non-spatial predicate, which has spatial meaning when transformed into spatial world. For instance, a predicate 'those places that are bound to province Ankara' yields a non-spatial set of populated places. However, each place has a location on earth. The coordinates of those places help to transfer the set of tuples into spatial world.

Derived Spatial DB (d_SDB) is based on a common schema for both PPDB_T and spatial data processor –for the time being ESRI-ArcInfo- that will import the data. Export schema is created by means of a non-spatial predicate that cuts the DB both vertically –a subset of attributes- and horizontally –a subset of tuples-. The exported non-spatial data is mapped into spatial format to generate d_SDB (Figure 5).

However, primary key is repeated in every fragment in vertical fragmentation. Thus, disjointness is valid only on non-primary key attributes in vertical fragmentation.

DCT contains a set of spatial data classes such as administrative boundaries, hydrology, transportation, elevation, populated places (only provinces and sub-provinces), physiography.

The relationship among PPDB_T and d_SDB is preserved by keeping the same primary key –Populated Place ID- in corresponding relations of both DBs.

The first level analysis of geometric correction is to search for 'Are the populated places bound to province Ankara being inside the administrative boundary of it' or 'Are the populated places located within the map sheet K being inside the boundary of mapsheet'

The final step is to design an integrated spatial DB (I_SDB) with the three component DBs; PPDB_T, DCT and spatial DB derived from a subset of PPDB_T.

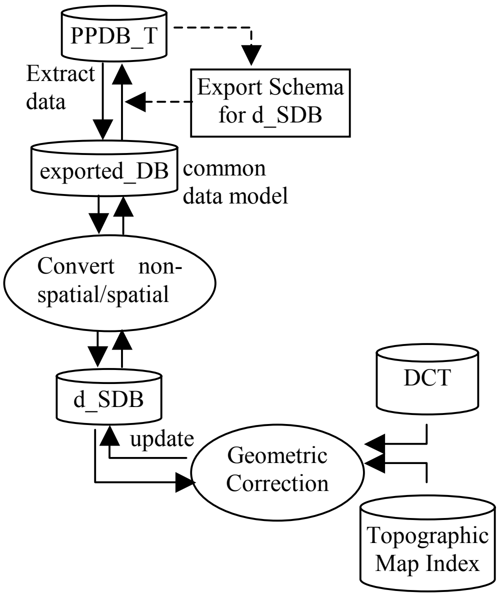

Figure 5: Procedures for geometric correction of PPDB_T.

### 3.4 Validating Database Integration

The application has tools for entering, manipulating and updating data in addition to intelligent query generator based on administrative hierarchy and standard topographic map indexes.

Since, mapping from data sources into PPDB_T is initially done automatically, the tools for mapping are available for further data injection into the DB, provided that schema of new data set is integrated with schema of PPDB_T. Moreover, there are tools to check consistency considering administrative hierarchy and to compare different versions of PPDB_T.

**Problems of Integration**

Constructing an integrated DB from existing DBs yield some problems due to lack of interoperability among DBs.

GCM and SSI have different non-spatial data to an extent for the same context. The data differs because of partly semantic but mainly schematic and format (syntax) discrepancy. For instance, in sub-district and village level of administrative hierarchy, a sub-district might be classified as village or visa versa in two non-spatial DBs. Updateness is the main reason for this kind of anomalies. If one of the DBs stores a different name from the current one for a populated place the corresponding populated places can not be matched till the mistake is removed or association is built.

Although, formats of both DBs are different, the format of SSI DB (plain text or spreadsheet) can be transformed into a common format such as dbf that is an open format for API of DBMS of PPDB_T.

**Future Work**

As PPDB_T is a central DB, communication, conversion, transfer, integration among PPDB_T and other spatial and non-spatial DBs are succeeded by using developed software. However, all these DBs have temporal behaviors changing in time. In order to maintain PPDB_T up-to-date, all the processes should be done for every change in one of the DBs. Therefore, a global schema for permanent part of the DBs and federated schemas for the user needs supported by a software or distributed DBMS may ease the cumbersome processes and may improve scelebility for extensions, updateness and maintenance. While PPDB_T and DCT are local databases, derived SDB, which is providing the interface between PPDB_T and DCT, is being generated on-the-fly. For the time being, integration with population DB of SSI is assumed to be immigrated into PPDB_T. However the model might be extended taking SSI DB as a component local DB (Figure 6).

The federated model (Federated Spatial DB (F_SDB)) defined above is going to be used for two purposes. Firstly, improving spatial and thematic accuracy of the PPDB_T that yield spatial and non-spatial information about populated places is used for the purpose of producing topographic maps and others in the organization. This target is accomplished by means of DB integration defined in this paper. Secondly, the F_SDB is going to be available for all users for the purpose of planning, analysis, thematic and statistical map generation etc.

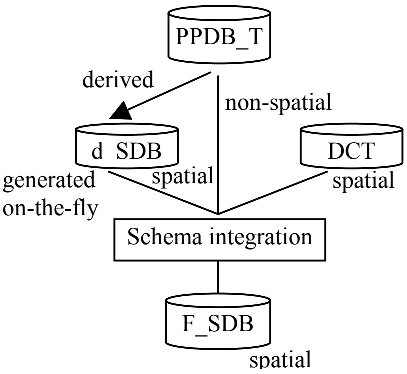

Figure 6: Federated Database model having three components.

## 4. CONCLUSIONS

In this paper, after introducing database integration strategies proposed in the literature, schema and data integration problems of PPDB_T are expressed. In real world applications, no modular model is fulfilling the needs. Therefore, a hybrid strategy comprising schema integration, data integration and software development in a good balance is defined for integration spatial and non-spatial databases. Schema and data are prepared for integration by transforming them into defined common models. Schema integration is done by hand. However, data integration is mostly done automatically by written software. Data transformations from non-spatial to spatial world and visa versa are accomplished by written programs. Consistency of PPDB_T is maintained considering application specifications and administrative hierarchy of Turkey. PPDB_T is going to be re-engineered to deploy the DB into distributed environment with a federated architecture.

**References** :

Abel., D.J., B.C.Ooi, K.L.Tan, S.H.Tan, 1998. Towards integrated geographical information Processing, International Journal of Geographic Information Science, 12(4), pp. 353-372.

Bobak, A.R. 1996. Distributed and Multi-Database Systems, Artech House, Boston, pp 121-138

Devogele, T., C.Parent, S.Spaccapietra, 1998. On Spatial Database Integration. International Journal of Geographic Information Science, 12(4), pp. 335-352.

Elmagarmid, A. et.al., 1999. Management of Heterogeneous And Autonomous Database Systems, Morgan Kaufmann Publishers, San Francisco, pp.2-32.

Hepner, P. 1995. Integrating Heterogeneous Databases: An Overview, http://citeseer.nj.nec.com/cs (accessed Dec. 2001)

Ozsu, M.T., P. Valduriez, 1999. Principles of Distributed Database Systems, Prentice Hall, New Jersey, pp. 75-101

Torun, A., 2000. Populated Places DB Project, General Command of Mapping. Internal Report, Cartography Department, General Command of Mapping, Turkey.

Torun, A. 2002. Designing Populated Places Database of Turkey (PPDB_T) by Using Relational Model, Harita Dergisi, 128 (on publish).

**Acknowledgements**

Populated Places Database of Turkey is designed and developed in Cartography Department of General Command of Mapping, Turkey.