



Welcome to the Virtual IMS *CONNECTION* newsletter. Virtual IMS *CONNECTION* at www.virtualims.com is an independently-operated vendor-neutral site run by and for the IMS user community.

Virtual IMS *CONNECTION* IMS presentation

The latest webinar from Virtual IMS *CONNECTION* was entitled, “The logical answer: IMS Logical Relationships”, and was presented by Aurora Emanuela Dell’Anno, an Engineering Services Architect with CA. Aurora is an IMS and DB2 Specialist Systems Engineer, with a past in application development. Aurora has spent the past 14 years specializing in various aspects of the IBM software world, with particular attention to database management and data warehousing. Keeping pace with the times, the past years have been dedicated to in-depth work on, and study of, data warehousing realities, trouble-shooting systems installations, and application performance issues – this included getting her hands dirty with distributed platforms, customers’ implementations, and problem-solving needs in this field, all from the perspective of systems engineering and database administration. Aurora works for CA in the highly-focused Mainframe Solutions Center.

Aurora started her presentation by explaining exactly what a hierarchical database was and what relational data structures were. She then asked why IMS needed logical relationships and suggested the reason was to access segments by a field other than the one chosen as the key, or to associate segments in two different databases or hierarchies. Aurora informed the user group that IMS provides two very useful tools to resolve these data requirements: secondary indexes logical relationships.

Thinking about database design, it can be useful to normalize the data. This helps to break data into naturally associated groupings that can be stored collectively in segments in a hierarchical database. It also breaks individual data elements into groups based on the processing functions, and groups data based on inherent relationships between data elements.

The database types that support logical relationships are HISAM, HDAM, PHDAM, HIDAM, and PHIDAM. There are no logical relationships with Fast Path DEDB or MSDB.

Logical relationships are formed by creating special segments that use pointers to access data in other segments. The path between the logical child and the segment to which it points is called a logical relationship.

Contents:

Virtual IMS <i>CONNECTION</i> presentation	1
Virtual IMS <i>CONNECTION</i> meeting dates	7
Recent IMS articles	7
Arcati Mainframe Yearbook 2010	7
Virtual IMS on Twitter and Facebook	8
About Virtual IMS <i>CONNECTION</i>	8

Virtual IMS CONNECTION

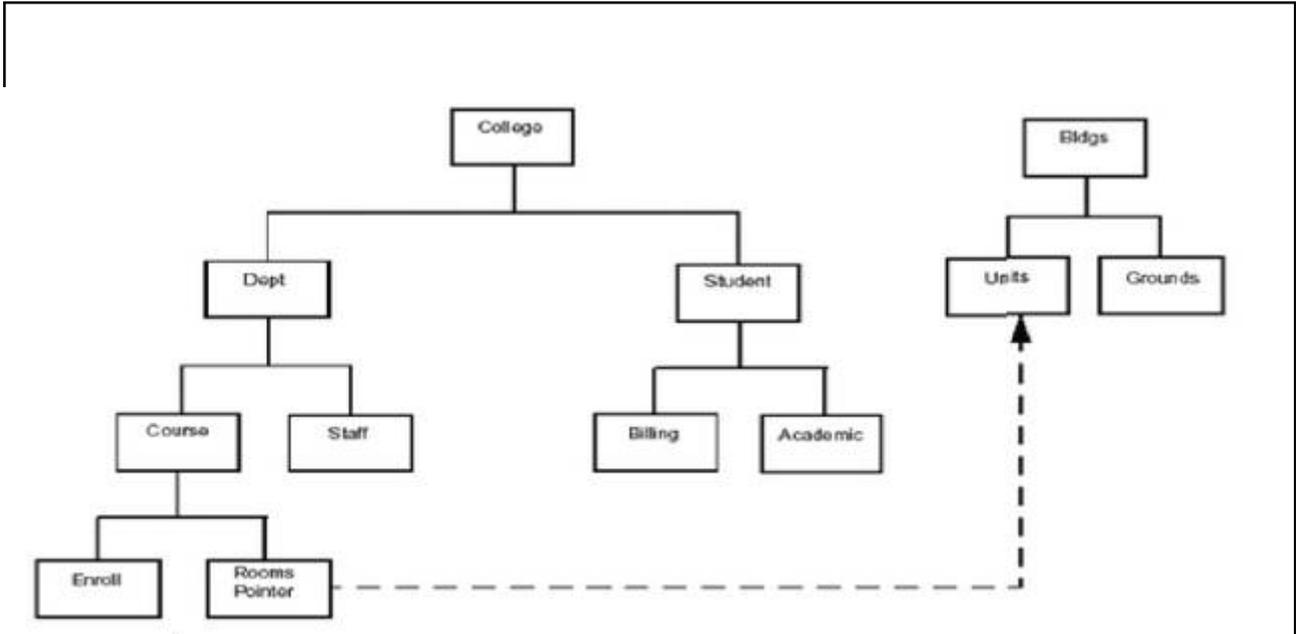


Figure 1: Logical relationships

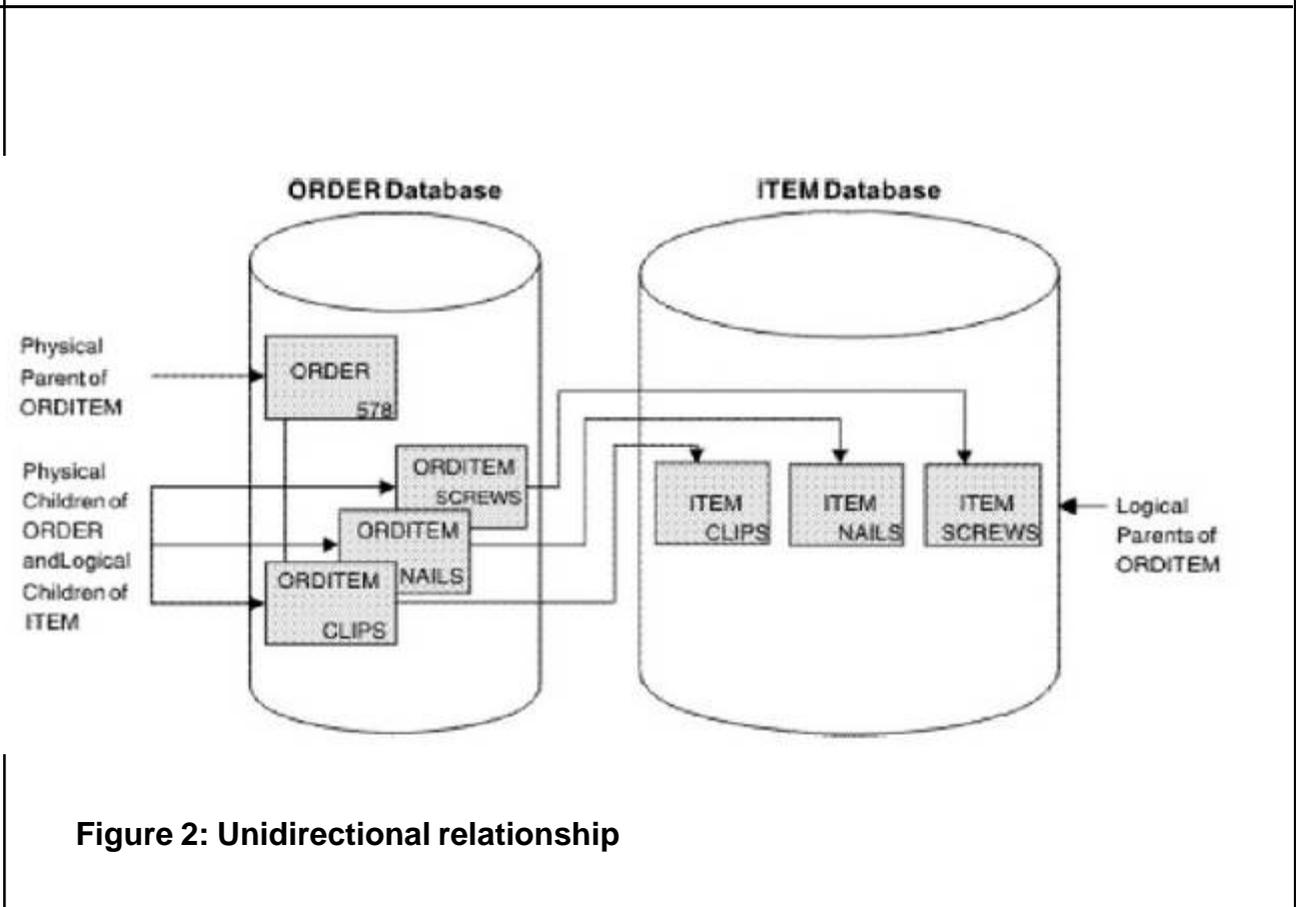


Figure 2: Unidirectional relationship

Virtual IMS CONNECTION

This is illustrated in Figure 1.

There are three types of logical relationships:

- Unidirectional
- Bidirectional physically paired
- Bidirectional virtually paired.

A unidirectional relationship links two segment types (logical child and its logical parent) in one direction. A one-way path is established using a pointer in the logical child – this is illustrated in Figure 2.

A bidirectional physically paired logical relationship links two segment types, a logical child and its logical parent, in two directions. A two-way path is established using pointers in the logical child segments – this is illustrated in Figure 3.

A bidirectional virtually paired logical relationship is like a bidirectional physically paired relationship. It:

- Links two segment types, a logical child and its logical parent, in two directions, establishing a two-way path
- Can be established between two segment types in the same or different databases.

A logical child segment exists in only one database. When going from database A to database B, IMS uses the pointer in the logical child segment. When going from database B to database A, IMS uses the pointer in the logical parent, as well as the pointer in the logical child segment. This is illustrated in Figure 4.

To establish a logical relationship, three segment types are always defined – the physical parent, the logical parent, and the logical child.

Aurora explained that the pointers used in logical relationships fall into two categories – direct and symbolic. We can implement logical relationships using both types, however, a direct pointer is a true pointer.

There are four types of pointers:

- Logical parent (LP)
- Logical Child (LC)
- Logical Twin (LT)
- Physical Parent (PP).

LP pointers point from the logical child to the logical parent. An LP pointer is in the prefix of the logical child and consists of the 4-byte direct address of the logical parent.

Logical child pointers are used only for logical relationships that use virtual pairing. With virtual

Virtual IMS CONNECTION

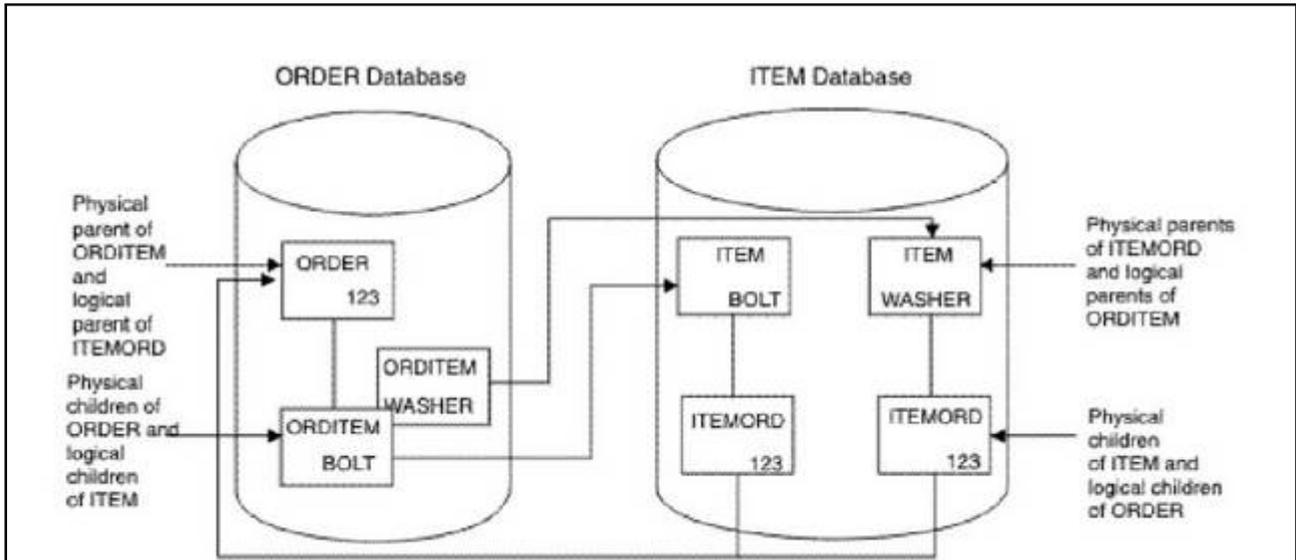


Figure 3: A bidirectional physically paired logical relationship

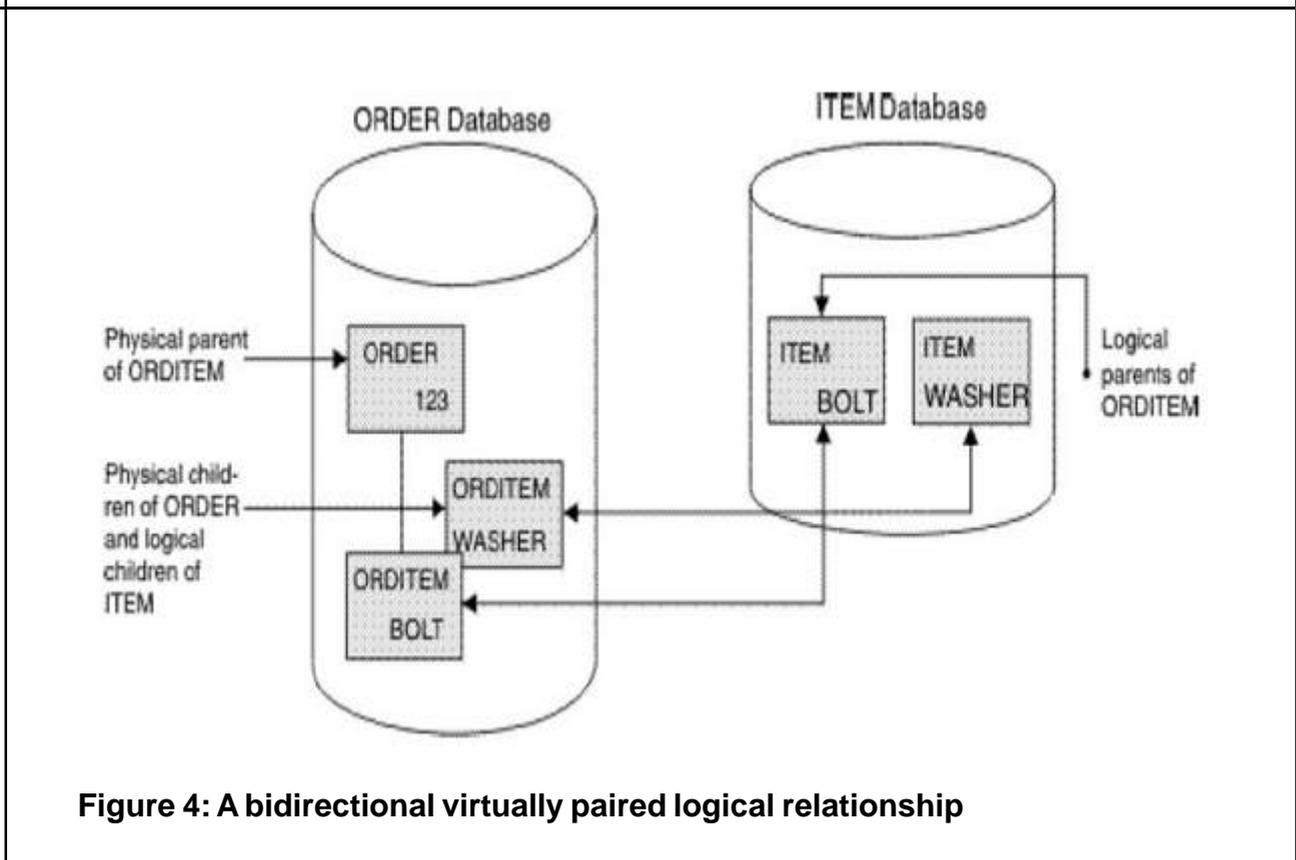


Figure 4: A bidirectional virtually paired logical relationship

Virtual IMS CONNECTION

pairing, only one logical child exists on DASD, and it contains a pointer to a logical parent. The logical parent points to the logical child segment.

Two types of logical child pointers can be used:

- Logical Child First (LCF)
- A combination of LCF and Logical Child Last (LCL) pointers.

Because LCF and LCL pointers are direct pointers, the segment they are pointing to must be in an HD database. The logical parent (the segment pointing to the logical child) must be in a HISAM or HD database. If the parent is in a HISAM database, the logical child must use a symbolic pointer to point to the parent.

A symbolic LP pointer consists of the Logical Parent's Concatenated Key (LPCK). It can be used to point into a HISAM or HD database.

Physical Parent (PP) pointers point from a segment to its physical parent. IMS generates PP pointers automatically for HD databases involved in logical relationships.

Logical Child Pointers are used only in logical relationships with virtual pairing. When virtual pairing is used, there is only one logical child on DASD, called the real logical child. This logical child has an LP pointer. The LP pointer can be symbolic or direct.

Two types of logical child pointers can be used:

- Logical Child First (LCF) pointers
- A combination of logical child first (LCF) and Logical Child Last (LCL) pointers.

CA's Aurora Dell'Anno went on to inform the user group that logical twins are multiple occurrences of logical child segments that point to the same occurrence of a logical parent segment. Two types of logical twin pointers can be used:

- Logical Twin Forward (LTF)
- A combination of LTF and Logical Twin Backward (LTB).

HALDBs (PHDAM, PHIDAM, and PSINDEX databases) use direct and indirect pointers for pointing from one database record to another database record. The use of indirect pointers prevents the problem of misdirected pointers that would otherwise occur when a database is reorganized. The repository for the indirect pointers is the indirect list data set. The misdirected pointers after reorganization are self-healing using indirect pointers.

The relationship between physical parent and logical child in a physical database and the LP pointer in each logical child creates a physical parent to logical parent path. For a physical parent to logical parent path, the logical parent is the destination parent in the concatenated segment.

For a logical parent to physical parent path, the physical parent is the destination parent in the concatenated segment.

Virtual IMS CONNECTION

When use of a physical parent to logical parent path is defined, the physical parent is the parent of the concatenated segment type. When an application program retrieves an occurrence of the concatenated segment type from a physical parent, the logical child and its logical parent are concatenated and presented to the application program as one segment. When use of a logical parent to physical parent path is defined, the logical parent is the parent of the concatenated segment type. When an application program retrieves an occurrence of the concatenated segment type from a logical parent, an occurrence of the logical child and its physical parent are concatenated and presented to the application program as one segment.

In both cases, the physical parent or logical parent segment included in the concatenated segment is called the destination parent.

There must be one physical DBD for each of the databases in a logical relationship. All statements are coded with the same format used when a logical relationship is not defined, except for the SEGM and LCHILD statements. The SEGM statement includes the new types of pointers. The LCHILD statement is added to define the logical relationship between the two segment types. The pointers for use with HD databases must be explicit in the PTR= parameter.

When a logical relationship is used, you must define the physical databases involved in the relationship to IMS. Also, you often need to define the logical structure of IMS since this is the structure the application program perceives.

With HALDB databases, bidirectional logical relationships must be implemented with physical pairing. When loading a new partitioned database with logical relationships, the logical child segments cannot be loaded as part of the load step. IMS adds logical children by normal update processing after the database has been loaded. HALDBs use an Indirect List Data Set (ILDS) to maintain logical relationship pointers when logically related databases are reorganized.

Aurora recommended a White Paper called *Converting Logical Relationships to Support Partitioned Databases* written by William N Keene – who some of you will have heard speak at user group meetings and who is a regular attendee. The White Paper provides guidance and examples for converting a set of logically related databases from bidirectional, virtual pairing using direct pointers to bidirectional, physical pairing using symbolic pointers.

The presentation contained a lot more detailed information, including some thoughts on performance. Aurora concluded by stating that logical relationships resolve conflicts in the way application programs need to view segments in the database. With logical relationships, application programs can access segment types in an order other than the one defined by the hierarchy, and have a data structure containing segments from more than one physical database.

A copy of Aurora's presentation is available for download from the Virtual IMS CONNECTION Web site at www.virtualims.com/presentations/IMS_Logical_Relationships.pdf.

Virtual IMS CONNECTION meeting dates

The following meeting dates have been arranged for the Virtual IMS CONNECTION user group:

- April 6, 2010 – Gregg Willhoit, Chief Architect, Progress DataDirect.
“Bringing zIIP offload to the IMS platform”
- June 8, 2010 – Patrick Fournier, SysperTec Communication.
“Improving the performance, efficiency, and TCO of SOA integration for IMS applications”
- August 10, 2010 – TBA, IBM.

All meetings start at 10:30 Central Time (4:30 GMT in the winter and 3:30 GMT during daylight saving time). All members will be e-mailed the WebEx URL, access code, and password for each meeting. Members need to log in about five minutes before the meetings starts in order for the WebEx connection to complete, and to phone into the freephone number provided so they can join in with the audio component of the meeting.

Recent IMS articles

IMS version 11: Reviewing features and functions of IBM's new release in TechTarget's CICS newsletter. You can find the article at searchdatacenter.techtarget.com/tip/0,289483,sid80_gci1378687,00.html.

Smarter Application Access to IMS Resources by Kenny Blackman in IBM Systems Magazine. You can find the article at www.ibmssystemsmag.com/mainframe/eneewsletterexclusive/31380p1.aspx.

Arcati Mainframe Yearbook 2010

The latest edition of the *Arcati Mainframe Yearbook* is available for download from the Arcati Web site at www.arcati.com/newyearbook10. The *Yearbook* is described by Mark Lillycrop as the the *de facto* reference work for IT professionals working with z/OS. It includes an up-to-date directory of vendors and consultants, a media guide, a strategy section with papers on mainframe trends and directions, a glossary of terminology, and a technical specification section.

One of the most eagerly-awaited section each year is the user survey. This has responses from 75 mainframe users, and examines how they currently use their mainframes and their plans for the future.

The Yearbook is available in PDF format. And it is completely **free**.

The Yearbook can only be free to mainframers because of the support given by sponsors. This year's sponsors were CA, DataKinetics, OpenMainframe.org, Progress:DataDirect, and Software Diversified Services (SDS).

Virtual IMS on Twitter and Facebook

The exciting news is that you can now stay up-to-date with all that's happening in the IMS world and with what's happening with the Virtual IMS CONNECTION user group by following us on Twitter or becoming a Fan on Facebook – as well as by checking the Web site regularly.



Find us on Twitter at
twitter.com/VirtualIMS

Become a fan on Facebook at
[www.facebook.com/#/pages/
Virtual-IMS-Connection-user-group/
282385116069](https://www.facebook.com/#/pages/Virtual-IMS-Connection-user-group/282385116069)



You could also add the Twitter and Facebook widgets to your Web sites, so not only do you stay in touch, but IMS professionals visiting your site will also be able to see the latest information. Examples of the widgets are shown below:



So, follow us on Twitter and become a fan on Facebook.

About Virtual IMS CONNECTION

Virtual IMS CONNECTION is a user group established as a way for individuals using IBM's IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.virtualims.com provides a central point for coordinating periodic meetings (which are technically-oriented topics presented in a webinar format), and will provide articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join Virtual IMS CONNECTION and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual IMS CONNECTION user group is free to its members. Various sponsorship opportunities are available to vendors of IMS-related products. Full details of these opportunities and can be found [here](#).