The Simplification of Data Warehouse Design

Relational Data Cubes and Sybase® Adaptive Server® IQ Multiplex

by Frank Teklitz
# Table of Contents

Relational Data Cubes and the Simplification of Data Warehouse Design......4

Evolution of Data Warehouse Design: The Last 15 Years ..........................5

- 3NF Schema .........................................................................................................5
  - Limitations of 3NF ............................................................................................6

- Star Schema .........................................................................................................7
  - Limitations of the Star....................................................................................8

Relational Data Cubes – An Evolutionary Approach to
Data Warehouse Design ..................................................................................10

Rcubes and Adaptive Server IQ Multiplex .......................................................12

  - Overcoming the Limits of Current Relational Technology ...........................12

  - Adaptive Server IQM's Bit-Wise Indexes for
    Faster Ad Hoc Queries...................................................................................14

Success Story – Compudigm Seepower and Crown Ltd.:
Catching the Eye of the Business Intelligence World...............................16
Relational Data Cubes and the Simplification of Data Warehouse Design

This paper explores the evolution of data warehouse design that has occurred over the last 15 years and the recent emergence of Relational Data Cubes (Rcubes) as an evolutionary design methodology.

To facilitate the comparison of data warehouse design methodologies, we will first define a model for a data warehouse to evaluate the different design strategies. The main issue of this paper is whether or not a collection of data, using relational storage models – 3rd Normal Form (3NF), Star schema and now Rcubes – provides real business value. In this discussion, we refer to the popular definition of a data warehouse as described by Bill Inmon, commonly known as the father of data warehousing. He defined a data warehouse as a “collection of subject-oriented, integrated, time variant, non-volatile data in support of management decisions.” Below are key components to help evaluate the value of a data warehouse:

**Subject oriented** – Does the data warehouse apply to the subjects the business is most interested in? Does the data warehouse have the data elements needed to analyze the business and is the data organized in a manner that is usable to solve real-world business problems? Also, are the proper data elements available for each subject area (such as marketing, finance distribution, inventory)?

**Integrated** – Can I view the data within and across subject areas? Any functional data warehouse (or data mart or OLAP) must navigate in at least three directions (sometimes called dimensions): geography (sales territories, market sectors, cost centers, production areas, distribution channels), objects (parts, products, shipping units, subassemblies, accounts, service types, etc.) and time (hours, days, weeks, years). The ability to navigate within a subject area is obviously critical, but to enable navigation across subject areas is also important for understanding complex business relationships.

**Time variant** – Can I see changes over time? The ability to measure incremental improvement (or the lack of it) over time is the single most important function of a data warehouse. If we are making poor decisions over time, without visibility into the impact of those decisions, we will continue to repeat our mistakes, costing our organizations significantly. But if we have visibility into the data, we can see the results of successful best practices, which can be expanded to improve overall organizational performance, and the results of poor practices also can be evaluated and corrected.

**Non-volatile data** – Can I depend on the data to be accurate and insulated from the turbulence of my transaction-based systems? Transaction-based, online transaction processing (OLTP) systems are volatile by their very nature. OLTP-based systems run the business – sales orders come in and are processed by the order entry system; the order entry system, in turn, generates information to have the product manufactured or pulled from inventory. The finished product is shipped, and the sale is recorded in the general ledger. And information is generated all along the way. It is this generated information that serves as the source data that incrementally (hourly, daily, weekly, etc.) feeds the data warehousing process. (We will go through the data warehouse process further in the next section.) It is the incremental nature of the data warehouse process that creates non-volatile environments for analysis.
Now that we have laid out a framework for comparison, we need to use these criteria to determine the most effective way of storing the data with regard to ease of development, maintenance and ease of end-user access. What is the most effective design schema for a data warehouse?

**Evolution of Data Warehouse Design: The Last 15 Years**

In the last 15 years, data warehouse design has gone through two stages of evolution: 3NF and the Star schema. By default, the first data warehouses used the 3NF method of design. The limitations of the 3NF schema for data warehousing design led to the development of the Star schema in the early 1980s. At that time, the Star schema represented a monumental breakthrough in data warehouse design. Today, data warehouses have become larger and larger and user access more demanding with the near ubiquitous Web front end. A new data warehouse design methodology is needed to meet these demanding new pressures.

To understand the need for a new data warehouse design methodology, let’s first look at the evolution of data warehouse design.

**3NF Schema**

Originally used to build database schemas for OLTP applications, 3NF often became the data warehouse schema out of expediency – it is easier to use a schema that is already available than to create one specifically for data warehousing.

---

*Diagram 1: 3NF Schema – Complex and Difficult to Use*
Using the popular definition of a data warehouse as a collection of subject-oriented, integrated, time variant, non-volatile data in support of management decisions, the 3NF example above is certainly a collection of data, but is it functional for business analysis? Let’s take a look:

**Subject oriented** – The 3NF schema above does not provide an easy way to see the various subject areas contained within the tables. Can you tell which tables apply to finance, sales or distribution? Also note that some of the tables look alike. For example, there are two tables that pertain to region. One is for sales regions, the other is for manufacturing. Would a user make the right choice?

**Integrated** – The 3NF schema does provide for integration, but this integration is designed for OLTP – not data warehousing. 3NF tables used for ad hoc data access analysis are inherently slow because of the numerous data paths into the data and the expense of indexing 3NF for ad hoc analysis. Also, 3NF does not provide for simplified navigation. I cannot easily navigate a 3NF data warehouse by geography, product or time.

**Time variant** – Can I see changes over time with 3NF? 3NF is designed for OLTP. OLTP is a world of individual transactions that flow from process step to process step at one point in time (like the last three minutes) and are not a collection of transaction events over time. Because of this inherent limitation, trend analysis can be difficult. In the case of the example on the prior page, there are three tables that define time: ship date, sales date and entry date. Which tables, if any, do I apply to a given query situation? Also, is the history for each timetable for the same amounts of time? It is difficult to gain this level of visibility with 3NF schemas.

**Non-volatile data** – Because of the inherent complexity of 3NF schemas, it can be very difficult to incrementally load information in a data warehouse and create a history. Typically, operational 3NF data is loaded as a “snapshot” from scratch from the source production systems. Without a mechanism to manage the incremental load of new data, it is difficult to create a stable, non-volatile data warehouse.

**Limitations of 3NF**

Although expedient, what many IT managers discovered very quickly was that the 3NF schema was difficult for end users to use and provided poor query performance. As can be seen from the example, 3NF is a very complex way of representing information. These schemas were designed to support the needs of operational OLTP applications where the access paths into the data are known in advance and the indexes are tuned for these known paths. For ad hoc analysis, each table and column and combination of columns represents a possible path into the data and a possible place to add an index. To index every column and combination of columns using conventional B-tree indexes would cause a huge data explosion, and still would not eliminate the complexity of a 3NF schema for ad hoc analysis. On top of these issues, the more tables used in an ad hoc query, the slower the query becomes because current relational technology is not designed to support numerous tables being accessed in an ad hoc fashion.
Star Schema

The need to work within the limitations of conventional relational databases, 3NF schema models and B-tree indexes has led to the development of the Star schema. The Star schema is generally credited to Ralph Kimball, who developed it while with Metaphor Computer Systems in the early 1980s. Below is an example of a very simple Star schema:

As can be seen from the diagram above, the Star schema eliminates the large number of paths into the data, reducing the number of indexes needed to support the data warehouse. For much of the late 1980s and 1990s, the Star schema proved to be a reasonable approach for ad hoc end user data access. The Star schema successfully meets all of the key elements for a complete data warehouse:

**Subject oriented** – Each star represents its own subject area and the data elements are organized for access by end users. For example, the information can be organized by the key subject areas (e.g., material, finance, manufacturing, etc.) needed to analyze the business and with the proper data elements for each subject area.

**Integrated** – Any functional data warehouse (or data mart or OLAP) must navigate in at least three directions: geography (sales territories, market sectors, cost centers, production areas, distribution channels, etc.); objects (parts, products, shipping units, subassemblies, accounts, service types, etc.); and time (minutes, hours, days, weeks, years). The ability to navigate within a subject area is an obvious strength of the Star schema. In the example above, there is a table for each path into the database (period table for time, market table for geography and product table for the object). These tables (called dimension tables) provide a reasonable way to navigate through one subject area. But how easy is the Star schema for navigating across subject areas?

**Time variant** – Can I see changes over time? The ability to measure the business over time to see incremental improvement (or the lack of it) is a key advantage of the Star schema. The Star schema is designed for easy trend analysis (at least within one subject area).
Non-volatile data – Can I depend on the data to be accurate and insulated from my transaction-based systems? The Star schema is designed to insulate the user from the volatility of transaction-based, OLTP systems. Through the use of extract-transformation-load tools (like Informatica, Ardent Data Stage or Sagent) it is possible to incrementally manage a Star-oriented data warehouse process and create a non-volatile environment for analysis.

Limitations of the Star

Along with the strengths of the Star schema are many limitations. The first is that it can be very inflexible. With a 3NF schema, we theoretically have unlimited access to any data path – even though that access is slow, costly and inefficient. To reduce the inefficiencies of 3NF, the number of paths (number of tables and columns) is reduced with the Star schema. A drawback of Star schema is that it can limit the number of data element choices. If a data element needs to be added at a later time, there is likely to be a significant cost in data management and maintenance. Because relational engines are inherently slow for doing calculations "on the fly," much of the numeric information must be precalculated and aggregated through the transformation process to gain reasonable query response time.

The second limitation is the data explosion that results from aggregations needed to support the Star schema in a traditional RDBMS. For every gigabyte of raw data, a Star schema will require at least an additional 2 gigabytes for aggregations and 2 gigabytes of B-tree (or bitmap) indexes. It is not unusual to have data and index explosion in the range of four to seven times the original raw data size to support the Star schema.

The third limitation is the amount of development and maintenance effort needed to manage a Star-oriented data warehouse. The level of effort needed to develop the Star schema, determine the aggregations and indexes, build a transformation process to extract data from source production systems and load the data warehouse can be tremendous. And once the data warehouse is built, the maintenance needed to add tables, columns, aggregates and indexes is an ongoing, time-consuming and extremely expensive process.
The fourth limitation is the difficulty of doing crossfunctional analysis with the Star schema such as is seen in Diagram 3, three Stars joined together (sometimes called a constellation of Stars) for finance, sales and distribution. What started out as one simple Star schema now becomes significantly more complex when a constellation of Stars are introduced to perform cross-functional analysis. Given the increased number of query paths into the data, cross-functional queries will have similar response times as 3NF schemas – slow. This will also add to the number of indexes and aggregates needed to support the crossfunctional analysis.

Diagram 3: Cross-Functional Star Schema Looks Like a 3NF Schema and will have the Same Limitations
Relational Data Cubes –
An Evolutionary Approach to Data Warehouse Design

For years, database administrators have wanted to develop schemas like the one pictured in diagram 4 below – one table with simple definitions. But given the state of the art of traditional relation technology through the 1980s and 1990s, this flat schema using Rcubes was not feasible. For example, the data element “Category” may only have three values, such as “Blue Line Pens,” “School House Pencils” and “Executive Desk Sets.” If this is a 10 million row table with a traditional relational database, the three values will be repeated 10 million times, causing extensive data explosion. This problem was the primary reason for developing the Star schema. Also, using a traditional relational database, every column and combination of columns (all permutations) would need a B-tree index for this flat schema, causing a huge index explosion. It is easy to see, given the high disk resource costs, why Rcubes and flat schemas were infeasible.

Diagram 4: Rcube from Sales Star Schema
Let’s also look at the power of Rcubes for cross-functional analysis. Below is an example of three Rcubes for cross-functional analysis:

Notice from a usability point of view the simplicity of the tables now needed for cross-functional analysis compared with diagram 3. There are eight fewer tables (which also means eight fewer extract, transformations and load operations) and nine fewer joins. This simplicity is the result of the “flatness” of the schema for each subject area, and this simplicity carries through to the crossfunctional queries and delivers superior performance.

The Rcubes successfully meet all elements for a complete data warehouse. Here is the value of Rcubes:

**Subject oriented** – Each Rcube represents its own subject area and the data elements are organized for easy access by end users. For this example, the information is organized by the key subject area (sales, finance, distribution) needed to analyze the business and with the proper data elements for each subject area.
Integrated – Any functional data warehouse (or data mart or OLAP) must navigate in at least three directions: geography, object and time. The ability to navigate within a subject area is an obvious strength of the RCubes. In the example from diagram 5, there is a column in the table for each path into the database (period table for time, market table for geography and product table for object). These columns provide the simplest way to navigate through a subject area. The major advantage of RCubes is that they are as easy to navigate for cross-functional analysis as a Star is for one subject area.

Time variant – Can I see changes over time? The ability to measure over time incremental improvement, or the lack of it, is a key function for RCubes. RCubes are designed for easy trend analysis (even across subject areas) and incremental update.

Non-volatile data – Can I depend on the data to be accurate and insulated from my transaction-based systems? RCubes are designed to insulate the user from the volatility of transaction-based, OLTP systems. Through the use of ETL tools (like Informatica, Ardent Data Stage or Sagent), it is possible to incrementally manage an RCube-oriented data warehouse process and create a non-volatile environment for analysis.

RCubes and Sybase Adaptive Server IQ Multiplex

Sybase® Adaptive Server® IQ Multiplex (IQM) is the only relational database that can take advantage of this leap in data warehouse architecture represented by RCubes. IQM, in conjunction with RCubes, represents a significant advance in database technology for building a new generation of easy-to-use, high-performance data warehouses. In this section, we will explain how IQM can help improve end-user performance while it can significantly reduce development and maintenance costs. We will discuss RCubes and how IQM changes the entire paradigm for developing data warehouse schemas.

Overcoming the Limits of Current Relational Technology

One of the major issues facing data warehouse developers today is that query performance is directly related to the design and maintenance of the Star schema. With the Star schema, if the database is designed correctly and is properly tuned with the right B-tree indexes, the performance will be acceptable. But as soon as there is a change in the way the user looks at the data, which can happen very rapidly, new indexes, aggregates and often new schemas need to be created. This leads to an environment where both development and maintenance costs are unpredictable and high. The database administrator is always tuning for yesterday’s query.

Database administrators have been limited in their ability to deliver flexible, easy-to-use and economical data warehouses because of the limits of traditional relational database technology and the need for B-tree indexes. In a traditional relational database, the data is stored separately from the index map. With IQM, the data and the index map are the same. As the user navigates the indexes, he is navigating the data. Because of this integration, only one map (index) is generally needed per column per table. (For details about this, see Colin White’s white paper on IQM titled, “Sybase Adaptive Server IQ: A High Performance Database for Decision Processing”.)
IQM is optimized for flexible, ad hoc business analysis and eliminates the need for complex development and maintenance procedures, while providing a robust, easy-to-use environment for end user analysis. IQM can take advantage of 3NF and Star schema, but is not limited to it. What IQM delivers is as follows:

- IQM is the only relational database that can take advantage of Rcubes to deliver data warehouses that are easy for ad hoc access, even for crossfunctional analysis. Diagram 5 shows a three-table join for Finance, Sales and Distribution that is significantly simpler than the cross-functional Star schema shown in Diagram 3.

- IQM uses Rcubes to enable data warehouses that are easy to design and maintain. With IQM, the database administrator can now develop the simple schemas he has wanted to develop for years because IQM can perform aggregation on the fly (eliminating preaggregated data and extra indexes) and does index reduction automatically as part of the index creation process. With IQM, there is less need for the Star schema because IQM does index reduction automatically and does not need the data preaggregated. The two most difficult chores for a database administrator today using the current generation of relational database technology (designed for OLTP) are designing and developing Star schemas for index and aggregation reduction, while still optimizing query performance.

- IQM is very flexible to design and develop versus 3NF or Star schema. The entire concept of dimension and fact tables becomes obsolete with IQM and Rcubes. Fact and dimension tables can be integrated together with IQM in ways that are easy for the user to visualize. These simpler IQM tables are, in turn, easier to join together for cross-subject queries.

- IQM is easy to maintain. Because of IQM’s bit-wise indexes and column store, it is easy to add columns to a table as they are needed. There is no need to rebuild indexes, aggregates or tables (as with traditional relational databases) because with IQM, it is easy to add and load columns to a table, and aggregation is done on the fly, eliminating the need to preaggregate or reindex the table.

- IQM eliminates data explosion. Because IQM does not need preaggregated tables, it is extremely economical. As stated earlier, using conventional relational technology and B-tree indexes, 200 gigabytes of raw data would turn into a 1-terabyte data warehouse (200 gigabytes of raw data + 400 gigabytes of aggregates + 400 gigabytes of index). With IQM, 200 gigabytes of raw data equal at most a 200-gigabyte data warehouse (probably more like a 120-gigabyte data warehouse with IQM’s automatic data compression). IQM accomplishes this by storing the data as columns (versus rows), eliminating aggregates.

- IQM dramatically reduces the transformation design and development effort. Much of the metadata developed for data transformation is used to preaggregate data and maintain complex 3NF and Star schemas. Because of the simplicity of Rcubes and the elimination of preaggregation with IQM, the transformation logic is greatly simplified.
• IQM dramatically reduces load time, reducing the number of load steps through a parallel load/indexing facility. Typically loading a data warehouse is a three-step process: stage the data from the transformation step; load the data into the data warehouse; and index the data. Each of these steps can be long and involved. With IQM, the load and index steps are combined because the data and index are a single structure, significantly reducing processing overhead and time. In addition, this load/index process is done in parallel and is multi-threaded. Typical load times can be as high as 40 gigabytes an hour depending on hardware configuration.

• IQM, like all Sybase server products, is open and takes advantage of EnterpriseConnect™ to directly access heterogeneous data sources (DB2, VSAM, Oracle, etc). With regard to end user tools, IQM uses ODBC for easy, standardized access. This guarantees “plug-and-play” integration between client tools and enterprise resources.

Sybase IQM and Rcubes dramatically reduce and simplify the design, development and maintenance effort needed to create a data warehouse. We are delivering to our customers an environment that can manage warehoused data, user volume and load requirements.

**Adaptive Server IQM’s Bit-Wise Indexes for Faster Ad Hoc Queries**

IQM is faster for ad hoc business analysis for a number of reasons. First, Sybase IQM’s patent pending index structures index and optimize each column of data based on the characteristics of the column. Today there are four types of indexes:

• Low cardinality indexes are for those fields that have less than 1,500 unique values (state description – CA, OH, etc.; gender code – male, female; product or part descriptions)

• Bit-wise indexes for high cardinality fields use calculations and range searches (sum of dollars and units, where price is less than $50)

• High group indexes are for key fields and grouping information (group dollars and units by product (high cardinality group by index) and plant location (low cardinality)

• Column stores are useful in “where” clauses and for projection of information onto the actual report or query result set.
Using IQM, the database administrator assigns an index to each column depending on the characteristics of the column (gender code – low cardinality, dollars – high cardinality, etc.). Using conventional B-tree indexes, the database administrator would have to make an educated guess as to which columns would be used alone (that column would get a B-tree index) or as a group (these groups of columns would also get a concatenated B-tree index). The more B-tree indexes the database administrator adds, the faster the database, but also the more disk space that is consumed. With traditional relational databases, as long as the user creates queries that fall within the B-tree index maps created, query performance will be acceptable. But what most customers have discovered is that this tuning process is a continuing and ongoing process because the database administrator is basically tuning for every new query, consuming significant development, maintenance and hardware resources. With IQM, this tuning process is fundamentally eliminated. The database administrator assigns IQM indexes per column generally just once. IQM will automatically compress the data as it is loaded, inserted or updated. There is little tuning needed after the initial column/index assignment.

The second reason that IQM is faster and easier to manage is that the column orientation of the indexes is carried through to the I/O buffers and memory. We like to say IQM zips through bits, while our competition chugs through bytes. In addition, IQM uses very large page sizes and disk I/O (typically 32- to 256-kilobyte page size), designed to pass large amounts of detail data into memory for fast, on-the-fly access.

The third reason for IQM’s speed and versatility is because it is a column store and not a row store. For example, if two columns out of 20 are used from a table, with IQM, only those two columns are projected from disk whereas a traditional RDBMS would have had to project the entire row off disk. The column store makes the data explosion of an Rcube manageable whereas a traditional RDBMS would use too much storage to achieve the same effect. In addition, IQM offers an enumerated column store where storage of duplicate values is eliminated on disk, further reducing data explosion and disk I/O.

What does this all mean to the users? First, query results will come back significantly faster, 10% to 100% faster, and with Rcubes 10 to 100 times faster because of IQM’s bit-wise indexing capabilities. This performance information is based on results from the hundreds of customers who own and use IQM. Second, the database schemas that the user access is significantly simpler. 3NF and Star schemas are difficult for users to understand and are very inflexible for business analysis. Database administrators have tried over the years to develop simpler schemas for the users but have been stymied by the need to do index reduction because of limitations of B-tree indexes and traditional relational databases. Because of IQM’s unique bit-wise indexes, the database administrator can build simpler Rcubes designed for easier end user access, enabling IQM’s bit-wise indexes to do data compression automatically.
Success Story – Compudigm Seepower and Crown Ltd.: Catching the Eye of the Business Intelligence World

The Crown Entertainment Complex comprises over 500,000 square meters of excitement and entertainment on the banks of Melbourne's Yarra River. This complex combines gaming, hotel, convention, restaurant, retail and comprehensive entertainment facilities with state-of-the-art technology to create one of the most innovative and technologically advanced sites in the world.

Crown Ltd.'s main business is gaming and in the gaming business, there are two basic goals: getting players into the casino and maximizing their overall customer experience. The first goal is achieved through successful marketing operations. The second is met by a number of factors, including top notch entertainment, fine restaurants, world-class accommodations, generous free “comps,” and, of course, the opportunity to strike it rich – all combining to make casinos one of the number one getaway destinations.

In gaming, as in any business, success or failure is based on the ability to make the right business decisions. In the gaming industry, decisions used to be made with a little trial and error, intuition and finally a best guess. Today the right decisions are built on hard data. In 1998, Crown was in a good position to make the right decisions – it had a lot of data. Like many casinos, Crown had been collecting both customer and gaming operations data for years through the use of customer loyalty cards and computerized gaming machines and tables. Unfortunately, even with all this data it was struggling to make the right decisions, and that year, Crown Melbourne Casino posted an AUS$140 million plus loss.

The good news for Crown was that it was one step away from making the right business decisions. Crown had the data, lots and lots of transactional data. What it lacked was an easy way to clearly see the information the data contained. Crown needed an easy way to see the key information contained in the data, based on selected key performance indicators (KPIs). Clayton Wheeler, gaming machine analyst for Crown Casino, knew there had to be an answer. He heard about a gaming solution offered by Compudigm International Ltd. that would let Crown see directly how its business operates and interact with customers.

Compudigm builds business intelligence solutions for the gaming and other industries. Its flagship product, SeePower®, integrates operational and customer data into a simple-to-understand visual presentation, empowering business management to easily spot trends and opportunities. Using the latest in easy-to-understand imaging technology and an easy-to-use Windows® wizard-based interface, SeePower provides immediate insight into marketing and business operations. The SeePower Analysis Suite for Gaming consists of three components: the Profiler for marketing, the Visualizer for business operations, and the Planner for managing floor layout. Working with Compudigm, Wheeler was confident he could help turn Crown in the right direction.
Going back to the basics, Crown first needed to get more patrons into its casino. To achieve this cost-effectively, it needed to greatly increase the return on its marketing campaigns. The first requirement for producing more effective campaigns is to clearly understand and identify customers. Who are they? Where do they live? What are their gaming interests? How much do they spend? How often? Using SeePower, Crown was able to immediately answer these questions and others to create a clear customer profile. SeePower’s Profiler module, Customer Loyalty Analysis Manager, enabled Crown to analyze existing customer data to a depth never seen before. “Now we can clearly identify exactly where our customer base resides, how much customers spend, how often they visit and so much more,” Wheeler remarked. “Using this information, we began to achieve a remarkable return on our marketing campaigns. In one campaign to 50,000 people, we achieved the same total response as we had in a mailing to all of our 800,000 Melbourne customers. And we spent less than 10% of what we had previously to accomplish this.”

SeePower also provides customers’ performance and characteristics over time, enabling Crown to identify the 4% of patrons who were responsible for 30% of Crown’s income. With this information, Crown is able to offer generous free offers or “comps” to its most loyal customers. These programs have resulted in significant returns, validating the generous “comps” strategy and ensuring its continuance.
Besides being able to clearly identify and characterize current and prospective customers, Crown now can directly characterize and target opportunities for marketing programs. Using SeePower’s clear data visualization, Crown management was able to target low-use periods and implement appropriate marketing programs. A campaign to drive up patronage during a recent seasonal low resulted in a 3.5% incremental revenue increase during the period of the offer.

Crown’s second business goal was to maximize its gaming operation. This is accomplished by making the most out of every square meter of casino space. For Crown, this turns out to be no small undertaking as its complex offers one of the largest gaming facilities in the Southern Hemisphere, with 350 gaming tables and 2,500 slots, spanning half a kilometer in length. Managing and maximizing a gaming operation of this size offer many logistical challenges. What is the right ratio of tables to slots? What machines are the most popular? What machine co-locations and adjacencies are most effective? What denomination works best for a particular machine? Or table? In a particular location? The layout and makeup of a casino floor can make or break casino revenues.

Using SeePower Planner for managing casino floor layout, Crown now can carry out spatial analysis using complex algorithms and data visualization techniques to track the usage and profitability of machines and tables over time. Crown uses SeePower Planner to determine a gaming machine’s best position on the gaming floor, track floor configuration effectiveness and immediately test out “what if” location changes. By providing a big picture of the floor layout, SeePower Planner reduces the amount of time Crown needs to manage its gaming floor layout. “Within an hour, one week’s worth of data is analyzed and available for display in a simple-to-understand format,” said Wheeler. “Previously we used spreadsheets and other reports, which took as long as two days to generate and have an expert make sense of it all.”

Diagram 9: SeePower’s extensive use of easy-to-understand imaging technology makes it easy to spot exactly which slots are hot performers.
Recently Crown implemented a major change to one of its room layouts, resulting in an increase in the percentage of low-revenue players. “We were able to detect the negative impact in a matter of weeks,” said Wheeler. “Previously it would have taken us months to see the subtle changes and ultimately realize the impact. With SeePower, we were able to turn it around very quickly.”

SeePower Visualizer tracks the performance of each gaming machine, then turns that data into a visual format so the user is able to clearly see the KPIs. Now Crown can easily analyze its operations from the point of the entire casino, a room, a bank of slots or even drill down to a single slot machine or table.

What sets SeePower apart from other business intelligence solutions is the ability to quickly analyze massive amounts of data and produce intuitive visualizations. Analysis of this scale requires large volumes of data to be stored and accessed very rapidly. To achieve this, Compudigm relies on the technology of Sybase’s Adaptive Server® IQ Multiplex (IQM) to deliver fast ad hoc business analysis, unmatched scalability and 24/7 reliability. IQM is a relational database offering patented indexing technologies designed specifically for data analysis. IQM’s technology takes advantage of the latest in Storage Area Network (SAN) technology. IQM’s unique independent Read-node/Writer-node architecture, stand-alone configuration (no shared CPUs or memory) and shared SAN access to redundant disks and fiber networks allow IQM to economically deliver the scalability and availability SeePower needs.

An additional advantage of IQM is its capability to deliver a new generation of data warehouse design using Rcubes. With Rcubes, Seepower delivers a tenfold performance improvement versus a traditional Star schema architecture. Combined with IQM’s typical tenfold or larger query performance improvement over a traditional relational database, SeePower achieves a significant one hundredfold performance improvement by using Rcubes and IQM together. According to Andrew Cardno, president of Compudigm, “SeePower offers highly advanced analytics, creating an extreme demand on the database environment. We really needed IQM, Rcubes and the 100 times performance we experienced. All the other relational databases we tried, which included Oracle 8i, Microsoft SQL Server and IBM DB2, just could not handle the query load. Additionally, Rcubes made the data warehouse much easier to design, manage and maintain.”

Today Crown, through its heavy use of the SeePower suite of tools, is realizing real improvements in growing its domestic slot and table business. SeePower, having already paid for itself multiple times over, will continue to drive patron frequency and length of stay, further contributing to Crown’s overall revenue growth.