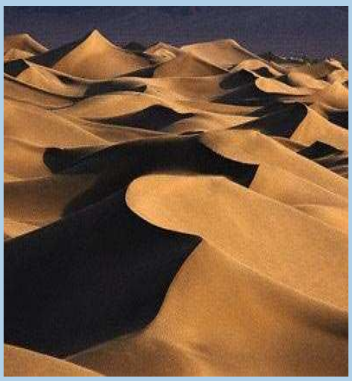
Birdstep's database products combine

the strength of two proven database technologies: Relational and Network

Data management in real-time and embedded systems is a rapidly changing landscape.



Together we can conquer the new challenges of the next generation systems…

as we already have done in the past with 20,000 developers...

who have created an installed base of over 17 million installations worldwide.

It's a pedigree we are proud of, one that few other database vendors can match.

RDM Embedded
For deeply-embedded and real-time systems

RDM Server
For shallow-embedded and packaged applications

RDM Mobile
For deeply-embedded and mobile devices

www.birdstep.com

Relational Database Model

The relational model stores data in tables composed of columns and rows. A join operation is used to relate data in separate tables using a duplicate column. *Figure 1* shows a typical relational database structure. While the relational model is flexible, performance is limited by the need to create new tables to hold results from relational operations, and storage requirements are increased by storing redundant columns in related tables. Join processing is done through an index which consumes valuable system resources, and requires storage and maintenance of index files.

Network Database Model

In contrast, the network model lets you define relationships between records using "sets", where pointers are used to relate objects directly and navigate between them. Figure 2 shows a typical network database structure. Compared to the relational model, the network model is faster (you can find a related object in a single logical disk access instead of relational's three or more), more reliable (since pointers to "member" objects are maintained directly in the "owner" object), more efficient (disk space is saved when sets are used since no indexes and duplicate fields are required), and better at expressing very complex database designs.

Combined Models for Optimum Performance

Since the Birdstep database products support both the network and relational models, you can use either one separately. But for real performance, use Birdstep database products to design a system combining the network and relational models. Figure 3 demonstrates how to implement a combined model database. Records that require fast random or sorted access are related through an index, while information that falls naturally into one-to-many relationships is organised into network sets.

"RDM Server had the highest performance of all databases we tested."

Andreas Schwegmann
HP Product Marketing Manager

Proven Data Management Solutions

Birdstep products provide a solid foundation for managing data in real-time and embedded systems. Products first released in 1984 with a proven track record.

Data Management for the Next Generation Devices

Many forecasts predict that within a decade, up to a trillion new devices will be using the internet to access and transfer data - a sobering thought, given the difficulties of managing data:

- Hardware and software platforms vary widely.
- Data must be 'dockable', i.e. edited off-line, then synchronized to on-line data stores.
- Operating conditions can be brutal - users may power down the device during a transaction.

Consulting Services

Our consulting division is able to provide specific ports or functionality by arrangement.

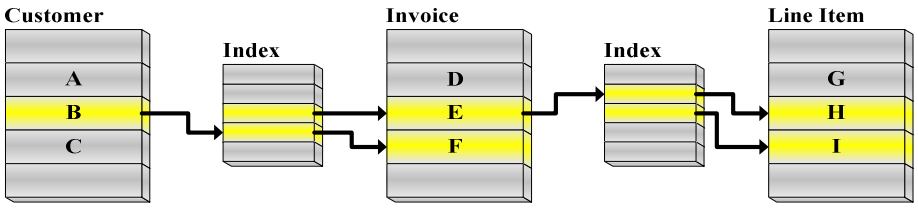


Figure 1: **Record types in a typical accounting system based on the relational model.** There are three record types: customer, invoice, and line type records. Each customer will have one customer record and several invoices, with several line items per invoice. To relate invoices to a customer, a customer code is stored in each invoice record as a key (foreign key). To find the invoices for a customer, the application searches the index for the customer code.

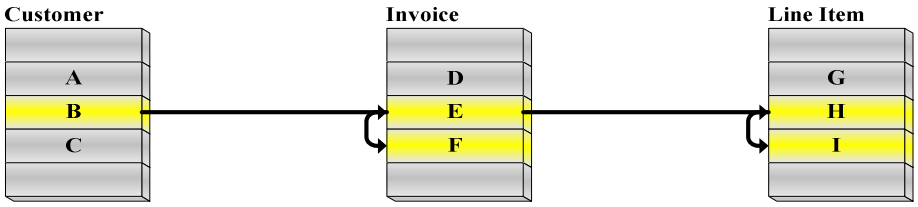


Figure 2: **The same accounting system based on the network model, using sets to define one-to-many relationships.** One customer has many invoices; one invoice has many line items. No redundant data is required because pointers relate one object to another. Access is direct and therefore fast. If you need to change the object, you make only the one change, as no other objects are affected. This means that referential integrity is assured.

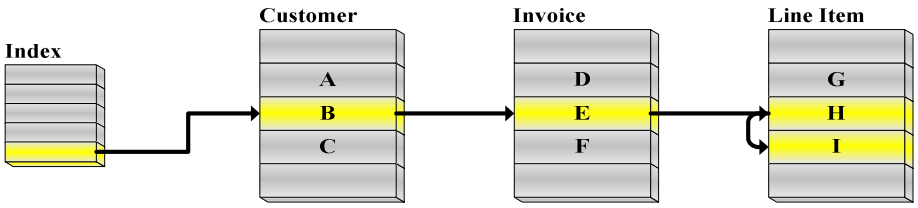


Figure 3: **The same accounting system based on the network model, using sets to define one-to-many relationships.** In this example a customer object contains a keyed field. Sets connect the customer object to the invoice object, and the invoice object to the line items. This index permits fast random access to a customer object, and the sets permits rapid retrieval of invoices and line item information.

Americas

Birdstep Technology Inc
2101 4th Ave
Suite 240
Seattle, WA 98121
Tel: +1 206 748 5353
Fax: + 1 206 748 5200
Email:
americas@birdstep.com

EMEA

Birdstep Technology
Birdstep Technology ASA
Bryggegata 7,N-0250 Oslo
Norway
Tel: +47 90628326
Fax: +1 206 748 5200
Email:
lpagni@birdstep.com

www.birdstep.com