



The Oracle Magician

November 2002

Volume I, number 3

From the Editor

Welcome to the third issue of the magazine, *The Oracle Magician!* This quarterly newsletter focuses on various “tricks of the trade” in the Oracle world --from DBAs, architects, developers, designers, and report writers.

Thank you to the many notes and suggestions from readers. They are most appreciated!

In this issue we present an overview written by Gavin Powell on the confusing subject of database normalization. Gavin explains and gives examples for each form of normalization.

The new book, “The Art and Science of Oracle Performance Tuning,” will be in bookstores soon! This issue contains a brief book update.

We are also looking for writers to submit articles that explore interesting ideas of use to DBAs or designers.

Please send all ideas to Editor@OracleMagician.com

Chris Lawson

The Mysterious World of Database Normalization

By Gavin JT Powell

What is Normalization?

Application of the Relational Database model to a data set involves the removal of duplication. Removal of duplication is performed using a process called *Normalization*.

Normalization is comprised of a set of rules called *Normal Forms*. Normalization is applied to subsets of data or tables in a database. Tables are for placing directly associated data into. Tables can be related or linked to each other through the use of index identifiers. An index identifier identifies a row of data in a table much like an index is used in a book. The index is used to locate an item of interest without having to read the whole book.

Five Levels of Normalization

There are five levels or layers of Normalization called 1st, 2nd, 3rd, 4th

and 5th Normal Forms. Each Normal Form is a refinement of the previous Normal Form. 4th and 5th Normal Forms are rarely applied.

In designing tables for performance it is common practice to ignore the steps of Normalization and jump directly to 2nd Normal Form. 3rd Normal Form is often not applied either; unless many-to-many joins cause an absolute need for unique values at the application level.

Over-Normalization

Over-Normalization can lead to poor performance in both OLTP and data warehouse type databases. Over-Normalization is common in top-down designed Java object applications. In this situation an object structure is imposed onto a relational database. Object and relational data

Continued on page 2

Table of Contents

From the Editor	Page 1
The Secret World of Database Normalization	Page 1
Update: The Art & Science of Oracle Performance Tuning....	Page 5

structures are completely different methodologies.

Too much jargon

Let us make the understanding of Normalization very simple. *Forget about it!* Normalization is for academics and in its strictest form is generally impractical due to its adverse effect on performance in a commercial environment, especially 3rd, 4th and 5th Normal Forms.

Normalization is for academics and in its strictest form is generally impractical

The simplest way to describe what Normalization attempts to achieve can be explained in three ways.

1. Divide the whole into smaller more manageable parts.
2. Removal of duplicated data into related subsets.
3. Linking of two indirectly related tables by the creation of a new table. The new table contains indexes from the two indirectly related tables. This is commonly known as a many-to-many join.

These three points are meaningless with-

out further explanation of Normalization. So let us go through the rules and try to explain it in a non-academic fashion.

1st Normal Form

1st Normal Form removes repetition by creating one-to-many relationships. Data repeated many times in one table is removed to a subset table. The subset table becomes the container for the removed repeating data. Each row in the subset table will contain a single reference to each row in the original table. The original table will then contain only non-duplicated data.

In the example in Figure 1, a 1st Normal Form transformation is shown. The purchase order table on the left contains customer details, purchase order details and descriptions of multiple items on the purchase order. Application of 1st Normal Form removes the multiple items from the purchase order table by creating a one-to-many relationship between the purchase order and the purchase order item tables. This has three benefits.

Continued on page 3

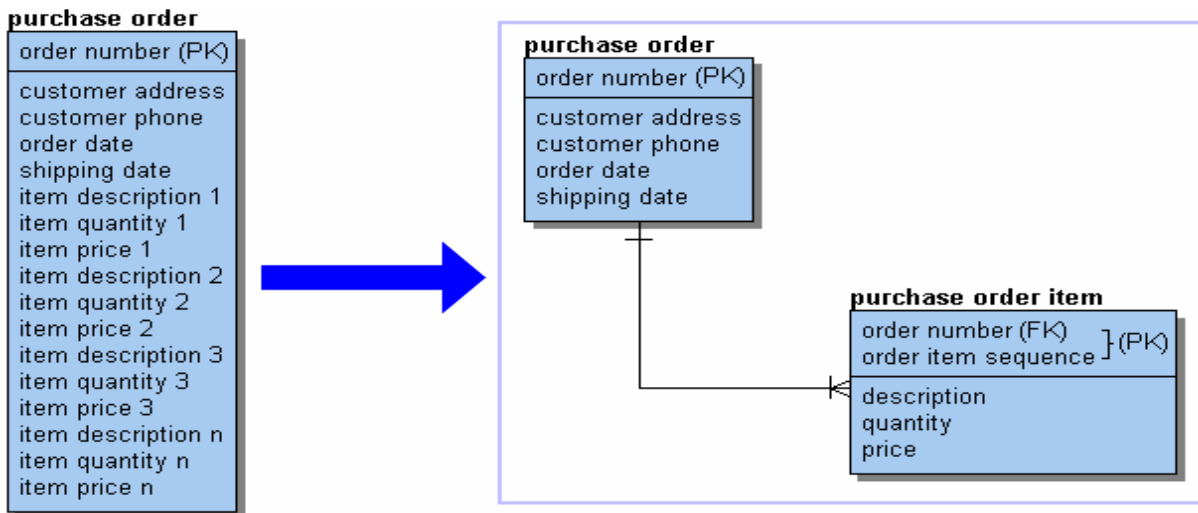


Figure 1. First Normal Form

- Saves space.
- Reduces complexity.
- Ensures that every purchase order item will belong to a purchase order.

In Figure 1, the *crows-foot* pointing to the purchase order item table indicates that for a purchase order to exist, the purchase order has to have at least one purchase order item. The line across the pointer to the purchase order table signifies that at least one purchase order is required in this relationship. The *crows-foot* is used to denote an inter-entity relationship. Inter-entity relationships can be zero, one or many to zero, one or many. The relationship shown in Figure 1 between the purchase order and purchase order item table is that of one-and-only-one to one-or-many.

2nd Normal Form

2nd Normal Form creates not one-to-many relationships but many-to-one relationships, effectively separating static from dynamic information. Static information is potentially repeatable.

This repeatable static information is moved into separate tables. In Figure 2, the customer information is removed from the purchase order table. Customer information can be duplicated for many purchase orders or have no purchase orders; thus the one-and-only-one to zero-one-or-many relationship between customer and purchase order.

3rd Normal Form

3rd Normal Form is used to resolve many-to-many relationships into unique values. In Figure 3, a student can be enrolled in many courses and a course can have many students enrolled. The point to note is that it is impossible to find a unique course-student item without joining every student with every course. Thus each unique item can be found with the combination of values.

Thus the *coursestudent* entity in Figure 3

Continued on page 4

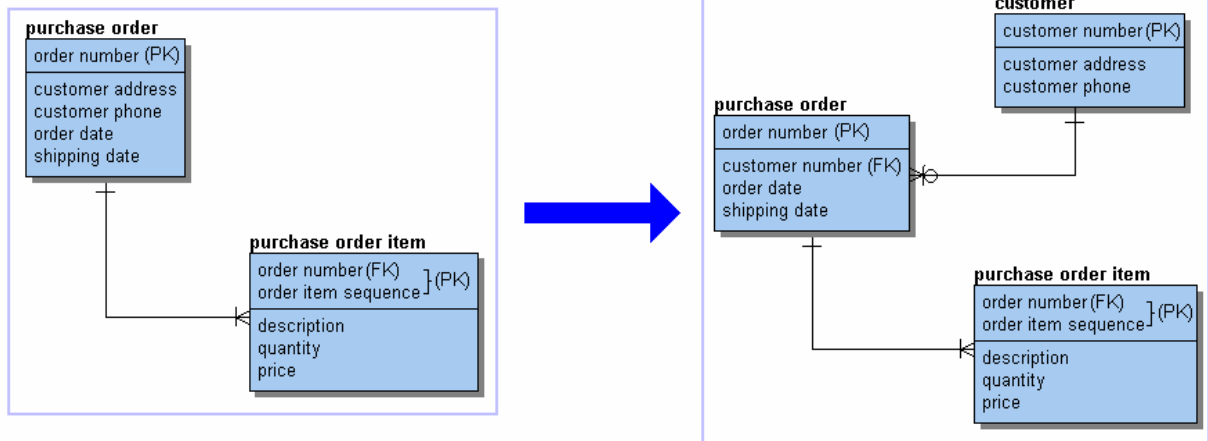


Figure 2. Second Normal Form

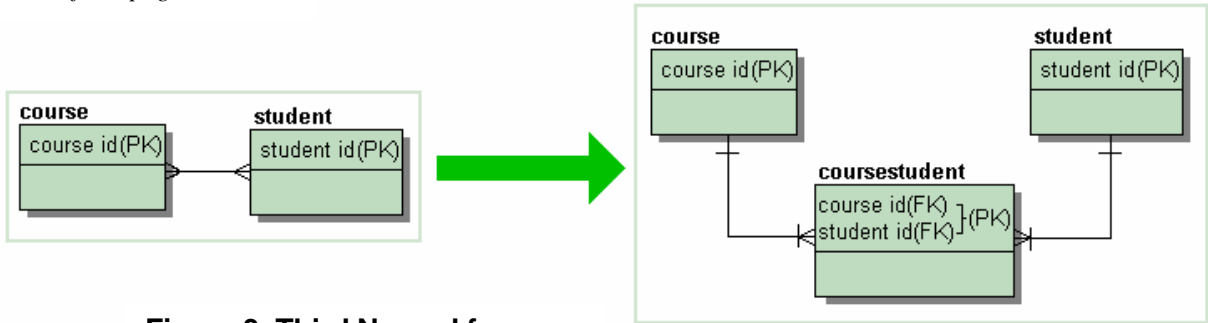


Figure 3. Third Normal form

is a many-to-many join resolution entity. In a commercial environment it is very unlikely that an application will ever need to find this unique item, especially not a modern-day Java object web application where the tendency is to drill-down through list collections rather than display individual items.

Many-to-many join resolutions should *only* be created when they are specifically required by the application. It can sometimes be better to resolve these joins in the application to improve database performance. Be very careful using 3rd Normal Form and beyond.

4th Normal Form

4th Normal Form is intended to separate multi-valued facts in a single table into multiple tables. In Figure 4,

employee skill and certification lists are removed into separate entities. An employee could have skills or certifications, or both.

5th Normal Form

5th Normal Form divides related columns into separate tables based on those relationships. In Figure 5, product, manager and employee are all related to each other. Thus three separate entities can be created to explicitly define those inter-relationships, resulting in information that can be reconstructed from smaller parts.

A Summary of Normalization

Continued on page 5

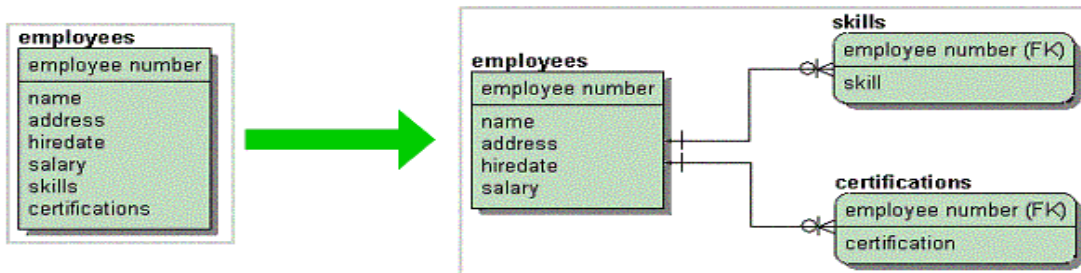


Figure 4. Fourth Normal form

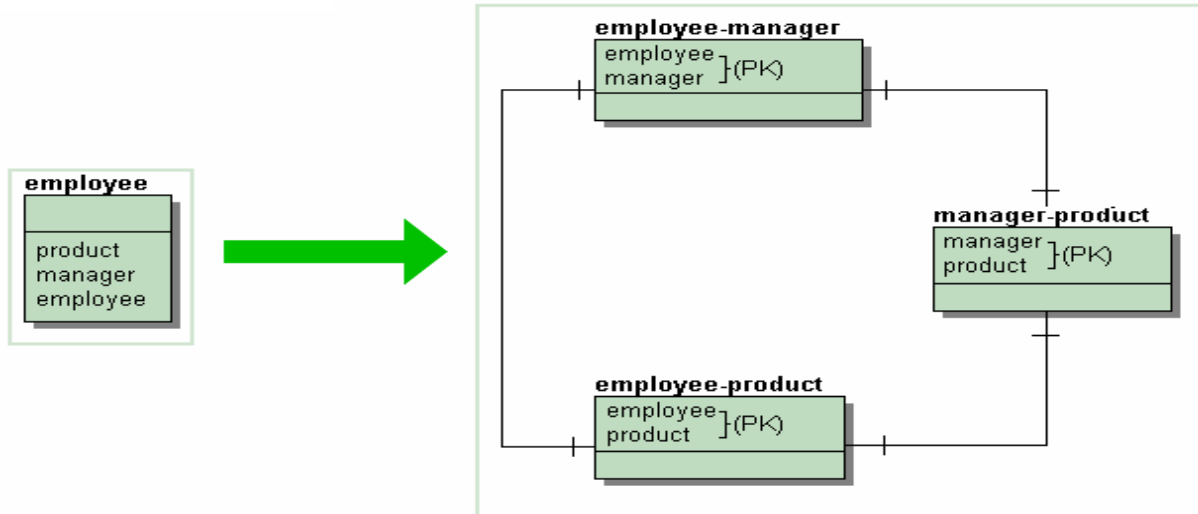


Figure 5. Fifth Normal Form

- 1st Normal Form removes repetition by creating one-to-many relationships.
- 2nd Normal Form creates not one-to-many relationships but many-to-one relationships, effectively separating static from dynamic information. 2nd NF removes items from tables independent of the primary key.
- 3rd Normal Form is used to resolve many-to-many relationships into unique values. 3rd NF allows for uniqueness of information by creation of additional many-to-many join resolution tables. These tables are rarely required in modern day applications.
- 4th Normal Form is intended to separate multi-valued facts in a single table into multiple tables. 5th Normal Form divides related columns into separate tables based on those relationships. 4th and 5th NF minimizes nulls and composite primary keys by removing null capable fields and subset composite primary key dependent fields to new tables. 4th and 5th Normal Forms are rarely useful.

Gavin JT Powell is a Senior Oracle DBA consultant who works in the San Francisco Bay area. Gavin can be reached at gjtpowell@earthlink.net

Book to Hit the Shelves Soon!

The Art and Science of Oracle Performance Tuning will be in bookstores soon! The author is working with the Curlingstone editors to finalize the manuscript.

Publication is expected in February!

The Oracle Magician

Editor: Chris Lawson



Contributing Editor:
Gavin Powell

Layout: Wizard Press