Progress Database Design Guide

P P S C .
T . T , ,
, , , -
, , P S C .

T , P S C
.

T .

P , P R , P W S P S
C U S . A , A S , P V P , S O ,
I S , P P S C .

S MQ S S C U S .

P S C R I T
S S 1993-1997 IBM XML P J E .

IBM C 1998-1999. A . U.S. G U R R U ,
GSA ADP S C IBM C .

P P S C IBM C
P /400 . P /400 AND 400 IBM C
P S C .

J J - S M , I .
U S .

A / .

.

M 2001

P C : 4511
I N : 81079;9.1C

Contents

Figures

Tables

Procedures

Preface

Purpose

T *Progress Database Design Guide* . U P 4GL . T , , , .

Audience

T - .

Organization of This Manual

C 1, I

T . I : , ( ), ( ). I P .

C 2, T R N

T . I .

C 3,

C 4, D I

T P .

C 5, P 4GL I U

T P 4GL .

C 6, P 4GL W I

T P 4GL .

C 7, C I U SQL-92

T .

C 8, P 4GL T

T 4GL .

C 9, J S P T

T J .

Typographical Conventions

T :

- **Bold typeface** :
  - C
  - T
- *Italic typeface* :
  - P
  - N
  - T
- `Monospaced typeface` :
  - C

- S
- O

T :

- S P .

END-ERROR, GET, GO
ALT, CTRL, SPACEBAR, TAB

- W , . Y ;.

CTRL-X

- W ; ; .

ESCAPE H
ESCAPE CURSOR-LEFT

Syntax Notation

T :

- U . A ,

I , ACCUM :

SYNTAX

```
ACCUM aggregate expression
```

- I . T . I ACCUM , *aggregate expression* ACCUM *Progress Language Reference*.

- Y (DO, FOR, FUNCTION, PROCEDURE, REPEAT) . DO, FOR, FUNCTION, PROCEDURE, REPEAT ,

```
FOR EACH Customer:
   DISPLAY Name.
END.
```

- S ([ ]) , , .

  I , STREAM *stream*, UNLESS-HIDDEN, NO-ERROR :

  SYNTAX

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

  I , ,

  F , INITIAL . I , ([ ) :

  SYNTAX

```
INITIAL [ constant [ , constant ] ... ]
```

  NOTE: T (...) , .

- B ({ }) , , .

  I , BY *expression* *DESCENDING*, :

  SYNTAX

```
{ BY expression [ DESCENDING ] }
```

  I , ,

F , . I , ( ) :

SYNTAX

```
{ &argument-name }
```

- A (|)

  I , EACH, FIRST, LAST , :

SYNTAX

```
PRESELECT [ EACH | FIRST | LAST ] record-phrase
```

I , *logical-name* *alias*:

SYNTAX

```
CONNECTED ( { logical-name | alias } )
```

- E (. . .) . I . I .

I , , . N :

SYNTAX

```
MAXIMUM ( expression , expression [ , expression ] . . . )
```

I , MESSAGE, *expression* SKIP, *expression* SKIP :

SYNTAX

```
MESSAGE { expression | SKIP [ (n) ] } . . .
```

I , *include-file*, *argument* &*argument-name* = "*argument-value*", :

SYNTAX

```
{ include-file
    [ argument | &argument-name = "argument-value" ] ... }
```

- I , . I , **optional** , . **Required** . I , ( ) .

I , WITH :

SYNTAX

```
WITH [ ACCUM max-length ] [ expression DOWN ]
  [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
  [ STREAM-IO ]
```

I , ASSIGN : *field*, *record*. O *field* *record* . T :

SYNTAX

```
ASSIGN {   { [ FRAME frame ]
               { field [ = expression ] }
               [ WHEN expression ]
           } ...
         | { record [ EXCEPT field ... ] }
       }
```

Progress Messages

P

- E

- C

- S

A

- C

- R

- H

- T

P 200:

```
** Unknown table name table. (200)
```

U P P . O W , P H :

- C H → R M P .
- C H → M , P . (I P , .)
- I P E , **HELP** (**F2** **CTRL-W**).

O UNIX , P PRO P .

1 ♦ S P P E :

```
install-dir/dlc/bin/pro
```

2 ♦ P **F3** , H → M .

3 ♦ T , **ENTER**. D .

4 ♦ P **F4** , **F3** P E , F → E .

Other Useful Documentation

T P S C .
U , W **and** C
CD-ROM.

Getting Started

Progress Electronic Documentation Installation and Configuration Guide (H)

A P EDOC UNIX
W .

Progress Installation and Configuration Guide Version 9 for UNIX

A P V 9.1 UNIX
.

Progress Installation and Configuration Guide Version 9 for Windows

A P V 9.1
W C M F .

Progress Version 9 Product Update Bulletin

A . T

Progress Application Development Environment — Getting Started (W)

A P A
D E (ADE). T ADE
P S O ,
S O ADE
.

Progress Language Tutorial for Windows *Progress Language Tutorial for Character*

P P . T
- - P
4GL.

Progress Master Glossary for Windows *Progress Master Glossary for Character* (EDOC)

P - P . T

Progress Master Index and Glossary for Windows *Progress Master Index and Glossary for Character* (H)

P - P -

Progress Startup Command and Parameter Reference

A P

Welcome to Progress (H)

A P , A - . *Welcome to Progress* P S C .

Development Tools

Progress ADM 2 Guide

A A D M , V 2 (ADM 2) P . I P S O .

Progress ADM 2 Reference

A A D M , V 2 (ADM 2) . I ADM 2 .

Progress AppBuilder Developer's Guide (W)

A ' P A B . A B R A D (RAD) P .

Progress Basic Database Tools (C ; W)

A P D A , D D .

Progress Basic Development Tools (C ; W)

A P , P P E A C .

Progress Debugger Guide

A P A D . T D .

Progress Help Development Guide (W)

A P .

Progress Translation Manager Guide (W)

A P T M P .

Progress Visual Translator Guide (W)

A P V T .

Reporting Tools

Progress Report Builder Deployment Guide (W)

A R B P R E .

Progress Report Builder Tutorial (W)

A - R B .

Progress Report Builder User's Guide (W)

A P R B .

Progress Results Administration and Development Guide (W)

A R . T , , R . I , - R R .

Progress Results User's Guide for Windows *Progress Results User's Guide for UNIX*

P -

R .E

R

.

4GL

Building Distributed Applications Using the Progress AppServer

A P A S . T

, 4GL , .

Progress External Program Interfaces

A -P P . T

, UNIX , W , W

, W A X , P H L C

I -P P

.

Progress Internationalization Guide

A P . T

(, ,)

.

Progress Language Reference

A -

, , , , , , , P

.

Progress Programming Handbook

A - P .

Database

Progress Database Administration Guide and Reference

T P P .

DataServers

P D S G

T D S -P . T D S ,
, . E D S , , , ,
Progress DataServer for ODBC Guide, *Progress DataServer for ORACLE Guide*, *Progress/400 Product Guide*.

MERANT ODBC B D R

T E D S ODBC MERANT ODBC . F , MERANT , PDF *installation-path*\odbc. T . A A R V 3.1 . I A A R , A W : http://www.adobe.com/prodindex/acrobat/readstep.html.

SQL-89/Open Access

Progress Embedded SQL-89 Guide and Reference

A P E SQL-89 C, - ESQL-89 E SQL-89 P . T ESQL-89 ANSI .

Progress Open Client Developer's Guide

A J A X P A S. T A S J A X .

Progress SQL-89 Guide and Reference

A P /SQL-89. I SQL-89 , SQL-89 D M L , SQL-89 D D L , P .

SQL-92

Progress Embedded SQL-92 Guide and Reference

A P E SQL-92 C, - ESQL-92 E SQL-92 P . T ESQL-92 ANSI .

Progress JDBC Driver Guide

A J D C (JDBC) P SQL-92 JDBC . I ’ JDBC .

Progress ODBC Driver Guide

A ODBC P SQL-92 ODBC . I ’ ODBC .

Progress SQL-92 Guide and Reference

A P SQL-92. I SQL-92 , SQL-92 D M L , SQL-92 D D L , P . T P SQL-92 J .

Deployment

Progress Client Deployment Guide

A .

Progress Developer’s Toolkit

A D ’ T . T P T .

Progress Portability Guide

A P , , , P .

WebSpeed

Getting Started with WebSpeed

P W S W W .
I W S W I .

WebSpeed Installation and Configuration Guide

P W S W UNIX . I W S , W S B , W S A , N S , .

WebSpeed Developer's Guide

P W S W S W .

WebSpeed Product Update Bulletin

A . T .

Welcome to WebSpeed! (H)

A W S . *Welcome to WebSpeed!* P S C .

Reference

Pocket Progress (H)

A P .

Pocket WebSpeed (H)

A S S W S .

SQL-92 Reference

(T -P .)

A Guide to the SQL Standard

D , C.J., H D . 1997. R , MA: A W .

Understanding the New SQL: A Complete Guide

M , J (D E C) A R. S . 1993. S F : M K P .

1

Introduction

T :

- W
- B
- T

T . I ; .

1.1 What Is a Database?

A *database* . A . S , , , . T , :

- T ' , . I , , , ' . T . Y , , ,

- T ' , . T ' , ' , . Y , , ' . F , $1,000,

- T , , , ' , . Y ID , . O , .

T , , , , . E , , ,

1.2 Computerized Databases

L ’ . W , . F , . T . A , . F . I , . A :

- **Centralized and shared data** Y . T . O T ’ .
- **Current data** S .
- **Speed and productivity** Y , , , , , .
- **Accuracy and consistency** Y . F , OD OH O , . I .
- **Analysis** D , , . Y . F , , , . A - , , .
- **Security** Y . T . F , .

- **Crash recovery** S . W , . T .
- **Transactions** T . T ; . T .

N , , .

1.3 Elements of a Relational Database

R . T *relational model* E. F. C ( ); . T ( ), ( ), ( ). T .

1.3.1 Tables

A *table* [illegible] .F [illegible]1 1 [illegible] C [illegible].

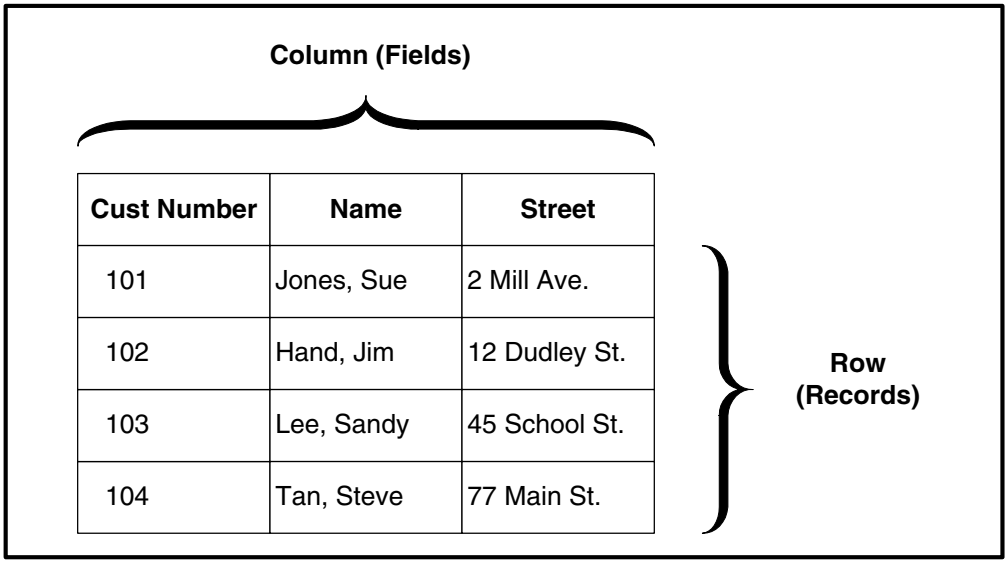


Figure 1–1: Columns and Rows in the Customer Table

E [illegible] O [illegible], A [illegible], S [illegible] S [illegible].

1.3.2 Rows

A [illegible] ([illegible]). A *row* [illegible]. E [illegible] .I [illegible]C [illegible] F [illegible]1 1, [illegible], [illegible].

S [illegible], [illegible] [illegible]O [illegible], [illegible] A [illegible], [illegible]S [illegible] S [illegible].

1.3.3 Columns

R [illegible] *columns* ([illegible]). A [illegible] .I [illegible]C [illegible], [illegible] [illegible]C N [illegible], N [illegible], S [illegible].

1.3.4 Keys

T : . A *primary key* ( ) . B , . A :

- I **mandatory**; , - . I ,
  .
- I **unique**. F , E S . T C N C . I , . A , (, C C , J J). F , (,).
- I **stable**; , . A , ,
  , , , .
- I **short**; , . S , . F , 9 30 .

NOTE: Y - .

A *foreign key* . I , . I , . I . C 2, T R N .

A *composite key* .

Indexes

A *index* . I , , , .

W . I , , .I

Y

A :

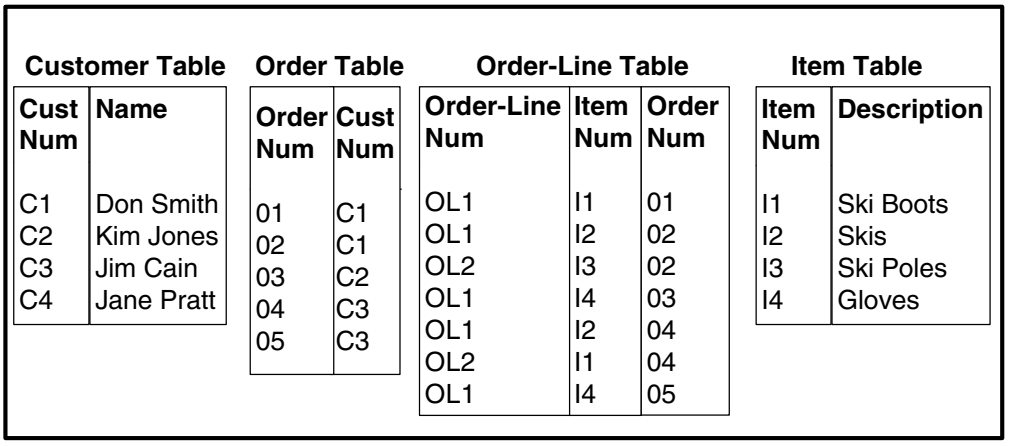
- F
- I 4GL
- W
- A
- E
- W

1.3.5 Applying the Principles of the Relational Model

T

M

N



Customer Table

Cust Num	Name
C1	Don Smith
C2	Kim Jones
C3	Jim Cain
C4	Jane Pratt

Order Table

Order Num	Cust Num
01	C1
02	C1
03	C2
04	C3
05	C3

Order-Line Table

Order-Line Num	Item Num	Order Num
OL1	I1	01
OL1	I2	02
OL2	I3	02
OL1	I4	03
OL1	I2	04
OL2	I1	04
OL1	I4	05

Item Table

Item Num	Description
I1	Ski Boots
I2	Skis
I3	Ski Poles
I4	Gloves

Figure 1–2: Example of a Relational Database

T _ _ _ _ _ _ _ F _1 2:

- T _C _ _ , _ _ _. E : C N N _. E _ _ _, C3 J C . T _ C N .
- T _O _ _ _ _ _ _ _ _C _ _. E O _ : C N (_C _ _) O _ N . T _ O _ N . T _C N _ _ _ _ _ _ _. T _ _ _ _ _ _ _ _, _ _ _.
- T _O _-L _ _ _ _ _-_ _ _. E _-_ _ : O _ N (_O _ _), O _-L _N , I _ N (_I _ _). T _ O _ N , O _ L _. T _ (O _ N I _ N) _ _ _C _, O _, I _ _ _ :
  - A _ _-_ _ _
  - I _ _ _-_ _
  - T _ _ _ _-_ _
  - I _
- T _I _ _ _ _ _ _. E I _ : I _ N D _ . I _ N _ _.

Y . T , . T I C , (/). F I O (O -L) O C . F 1 3 .

NOTE: T . A .

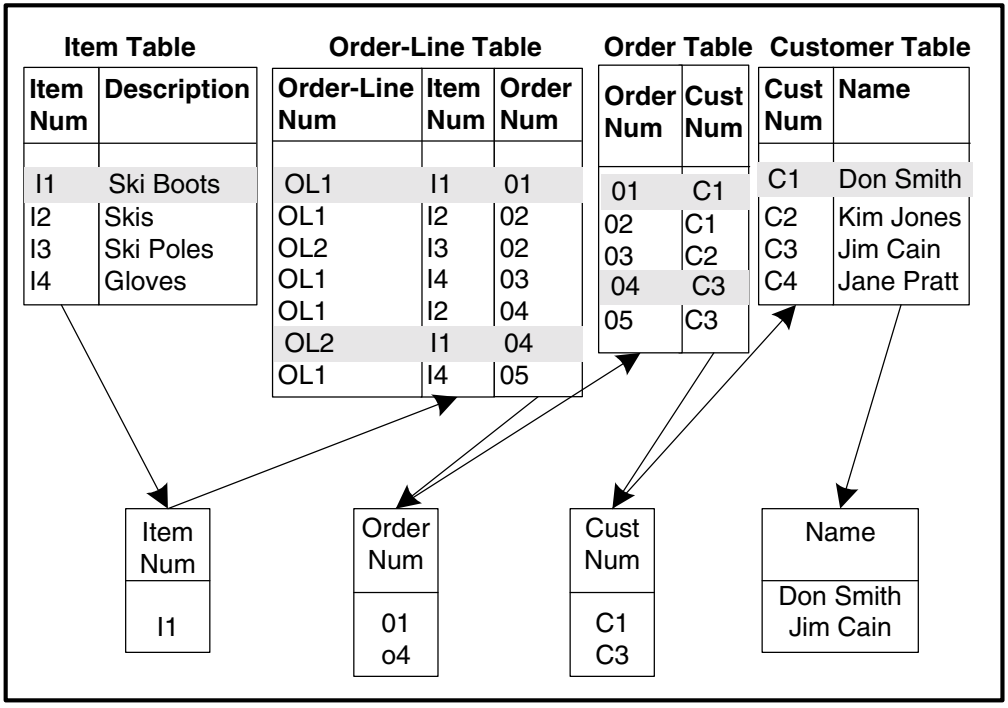


Figure 1–3: Selecting Records from Related Tables

1 ♦ F , I D . T I N I1.

2 ♦ N , O I I1. B O ’ I . I O -L . S O -L I1, O O -L . O 01 04 I N I1.

3 ♦ N O N ,
. S 01 04 . T
C1 C3.

4 ♦ F , C C1 C3, C
C1 C3. D S J C .

B , .
. T
, .

1.3.6 The Progress Database and the Relational Model

T P (RDBMS).Y ,
, , .

Database Schema and Metaschema

T P : , , . T ’
, , , , .
schema data definitions.

T
metaschema. T , .A
( ). T .

NOTE: T . T ,
, .

The sports Database

T ; . T 1 1 .

Table 1–1: The Sports Database

Table	Description
C	N , , , ,
I	F
I	S , ,
L -D	F
O	S
O -L	I
R -C	C .
S	N , ,
S	U.S.

T , , , .

1.4 Key Points to Remember

- A , , .
- A . A , , .
- A , .

- A , .
- A , , .
- A ( ) . B , , . I .
- A . I .
- A ( ) .

2

Table Relationships and Normalization

T :

- T
- O , , ,
- N
- F ,

2.1 Table Relationships

I , . T , *joined*. W . (T ; , .)

I F 2 1, C O , C N .

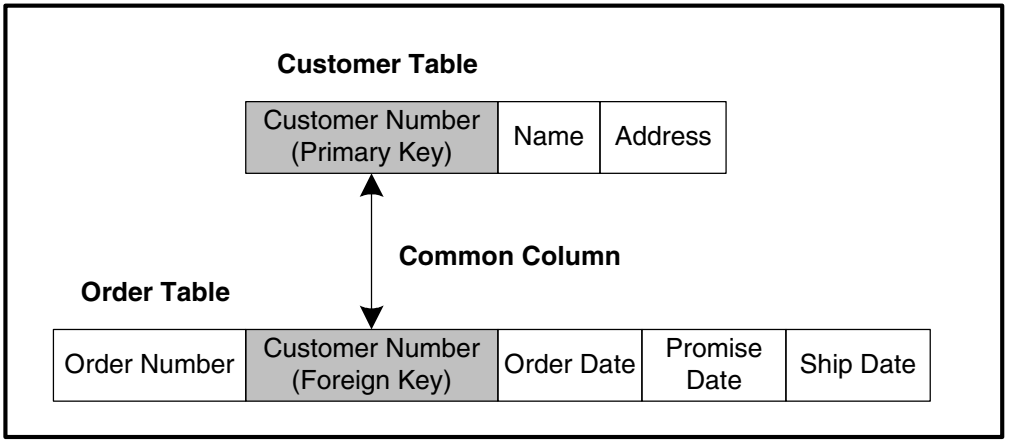


Figure 2–1: Relating the Customer and Order Tables

I C N , :

- F , (, , ,) .
- F ( ) ’ .

NOTES

- I SQL , DATE, TIME TIMESTAMP BIGINT, INT, SMALLINT TINYINT.
- I SQL , - , BINARY VARBINARY, NUMERIC, CHAR VARCHAR. H , D REAL FLOAT - .

F 2 2 C O .

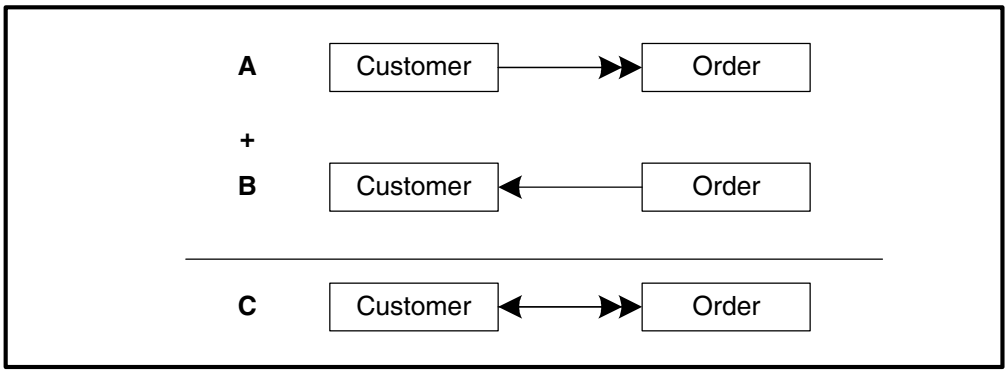


Figure 2–2: Relationship Between the Customer and Order Tables

F A, C ’ , C O - - ( ).

F B, O ’ , O C - - ( ).

C C O O C .

T - , - - ,
- - - .

2.1.1 One-to-one Relationship

A *one-to-one* . F , . T . T , . F 2 3 .

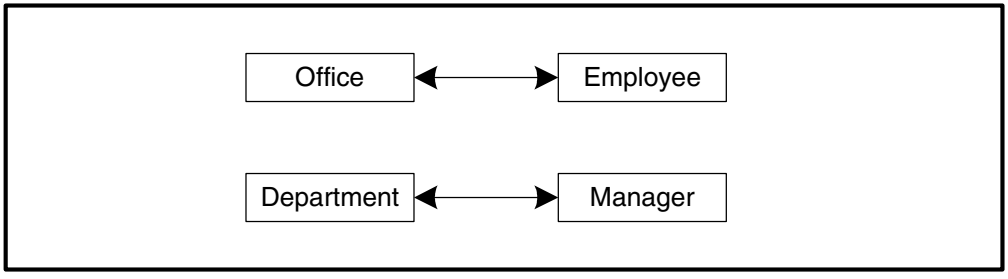


Figure 2–3: Examples of a One-to-one Relationship

H , ;

.

2.1.2 One-to-many Relationship

A *one-to-many* . F 2 4 : . A

.

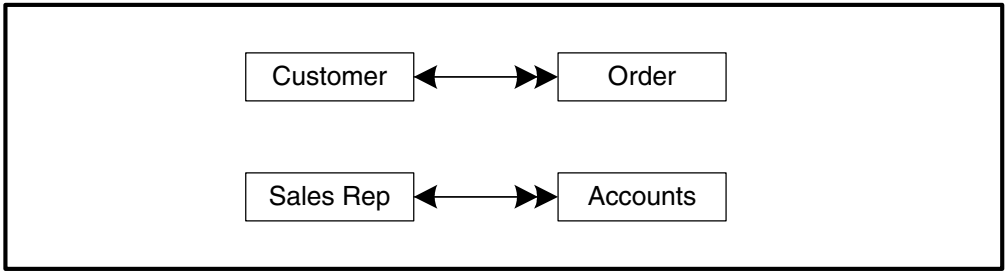


Figure 2–4: Examples of a One-to-many Relationship

H , - - , . T .

2.1.3 Many-to-many Relationship

A *many-to-many* . L , . F 2 5 : , ; , .

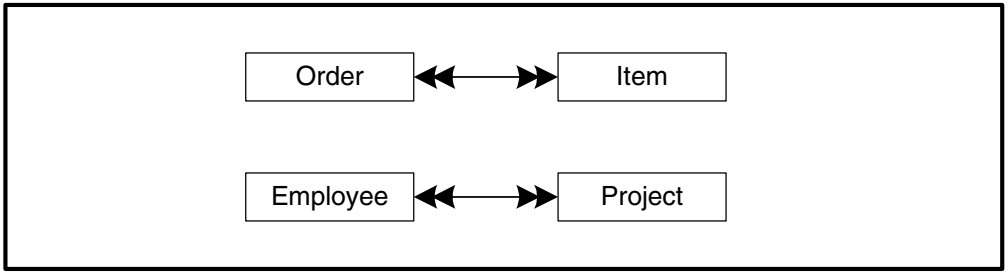


Figure 2–5: Examples of the Many-to-many Relationship

A

. F

two

F

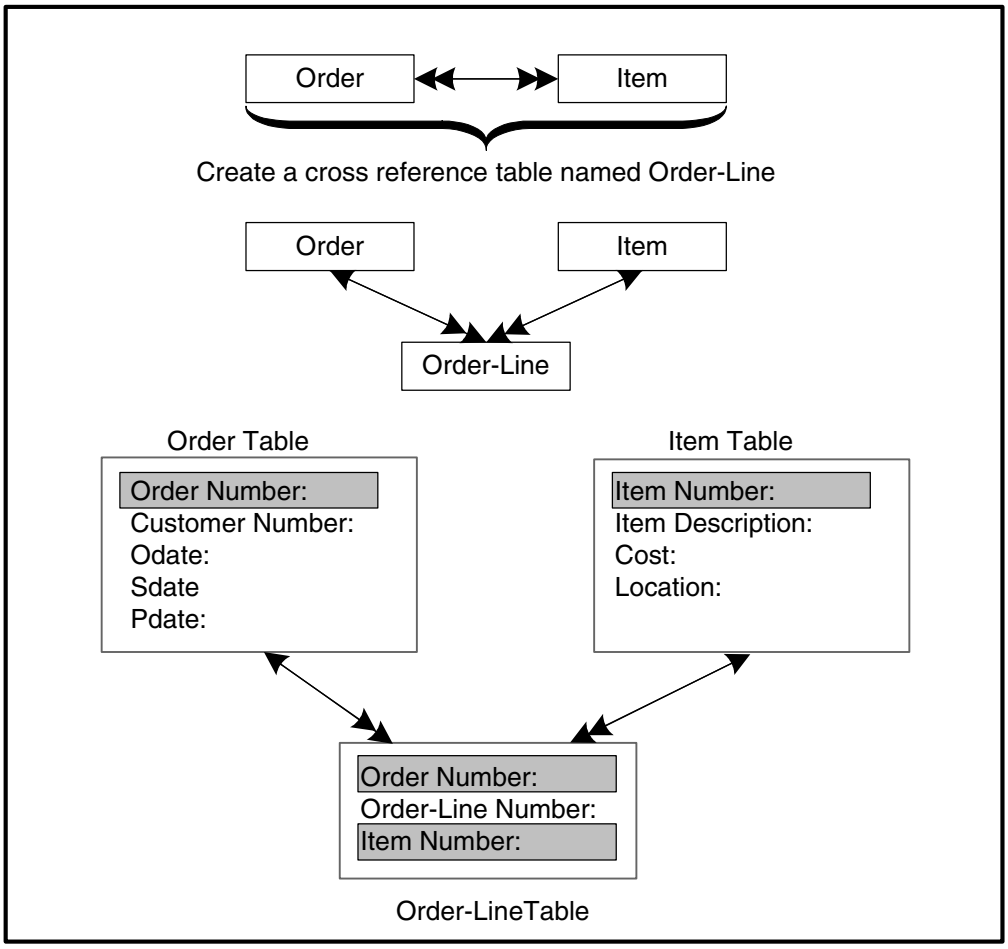


Figure 2–6: Using a Cross-reference Table to Relate Order and Item Tables

2.2 Normalization

T .
Normalization . D . Y - , - . T .

N . I .
W :

- C
- D
- D
- C
- C

T , .

2.2.1 The First Normal Form

T :

-

F , C , T 2 1.

Table 2–1: Unnormalized Customer Table with Several Values in a Column

Cust Num	Name	Street	Order Number
101	J , S	2 M A .	M31, M98, M129
102	H , J	12 D S .	M56
103	L , S	45 S S .	M37, M40
104	T , S	67 M S .	M41

H , O N . T M56 . T O N . I , , .

T 2 2 C .

Table 2–2: Unnormalized Table with Multiple Duplicate Columns

Cust Num	Name	Street	Order Number1	Order Number2	Order Number3
101	J , S	2 M A .	M31	M98	M129
102	H , J	12 D S .	M56	N	N
103	L , S	45 S S .	M37	M140	N
104	T , S	67 M S .	M41	N	N

H , O N , O N (O N 1, O N 2, O N 3). T . W ? Y . I . I , 200 O N . A 10 , 190 .

F _ _, _ _ _ _ _ _ . F
_ _, _ _ _ _ _ O _ N _ M98,

A :

- I
- I O N
- I

2.2.2 The Second Normal Form

A

T ,

T 2 5 C

Table 2–5: Customer Table with Repeated Data

Cust Num	Name	Street	Order Number	Order Date	Order Amount
101	J , S	2 M A .	M31	3/19/91	$400.87
101	J , S	2 M A .	M98	8/13/91	$3,000.90
101	J , S	2 M A .	M129	2/9/91	$919.45
102	H , J	12 D S .	M56	5/14/90	$1,000.50
103	L , S	45 S S .	M37	12/25/90	$299.89
103	L , S	45 S S .	M140	3/15/91	$299.89
104	T , S	67 M S .	M41	4/2/90	$2,300.56

H , , I :

- F , : C N , N , S . T *redundant data*.
- S , , S J , , , . I , . A *inconsistent data*. T , *integrity* .
- T , N 1, 1990, , J H S T . T *anomaly*.

S , ? N , C N , N , S , , . I , O N , O D , O A C N . T .

T C , , T 2 6 T 2 7.

Table 2–6: Customer Table

Cust Num (Primary Key)	Name	Street
101	J , S	2 M A
102	H , J	12 D S .
103	L , S	45 S S .
104	T , S	67 M S .

Table 2–7: Order Table *(1 of 2)*

Order Number (Primary Key)	Order Date	Order Amount	Cust Num (Foreign Key)
M31	3/19/91	$400.87	101
M98	8/13/91	$3,000.90	101

Table 2–7: Order Table *(2 of 2)*

Order Number (Primary Key)	Order Date	Order Amount	Cust Num (Foreign Key)
M129	2/9/91	$919.45	101
M56	5/14/90	$1,000.50	102
M37	12/25/90	$299.89	103
M140	3/15/91	$299.89	103
M41	4/2/90	$2,300.56	104

N C , O N . T O , C N , O C .

A :

- I
- I
- I

2.2.3 The Third Normal Form

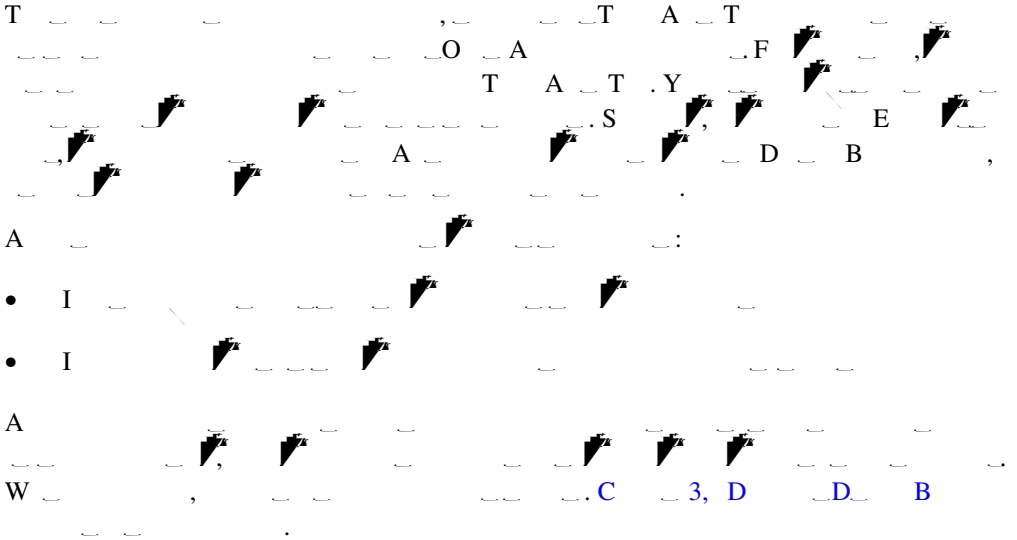
A , ,
.

T .

T 2 8 O T A T
10% O A .

Table 2–8: Order Table with Derived Column

Order Number (Primary Key)	Order Date	Order Amount	Total After Tax	Cust Num (Foreign Key)
M31	3/19/91	$400.87	$441.74	101
M98	8/13/91	$3,000.90	$3,300.99	101
M129	2/9/91	$919.45	$1011.39	101
M56	5/14/90	$1,000.50	$1,100.55	102
M37	12/25/90	$299.89	$329.87	103
M140	3/15/90	$299.89	$329.87	103
M41	4/2/90	$2,300.56	$2,530.61	104



3

Database Design Basics

O , . D . T . T :

- D
- L
- P
- P

3.1 Database Design Cycle

F 3 1 .

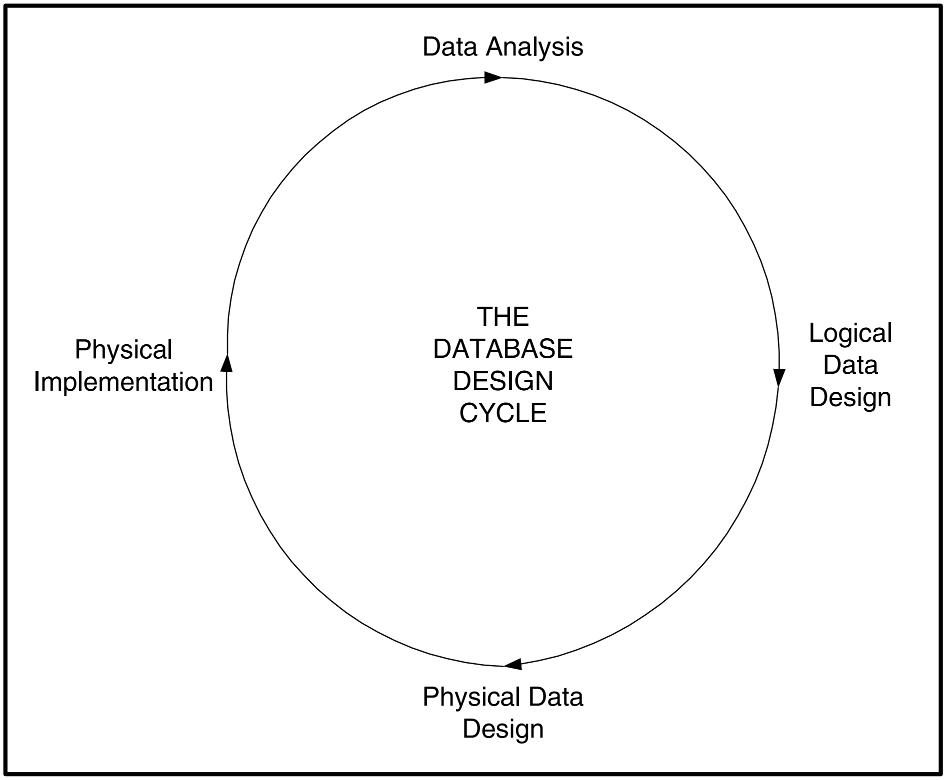


Figure 3–1: Database Design Cycle

Y . T
; ,
.

3.2 Data Analysis

T .
A :

- W ? W
?

- W I ? W I
?

- W I ?
- W ?
- W ?

T , . F :

- I , ,
- A , ,
- S ,
- L $1,000
- L - -
- L (,)
- L 200 ,
- L
- T
- T . P
- T , , , ,
- T ,

T , , .

- *Kernels* . T . S : , , , , ,
.
- *Associations* . F , .
- *Characteritics* . T , , .
F , - ; , - .

T , , , , ,
. W , , , .

3.3 Logical Database Design

Logical database design ,
. W , , , ,
.

O , , (,) (,).

C - , , . I :

- D .
- D .
- D ( ) .
- N .
- D . A *domain* . F , , .

A

.

3.4 Physical Database Design

T *physical database design* . I , D :

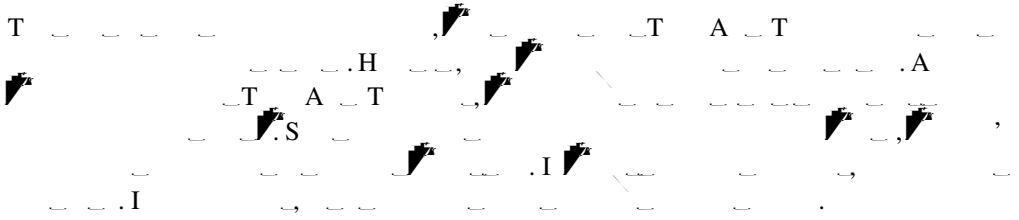
- W I ?
- W ( ) I ?
- W I ?
- S I ?

I . *Denormalizing* -

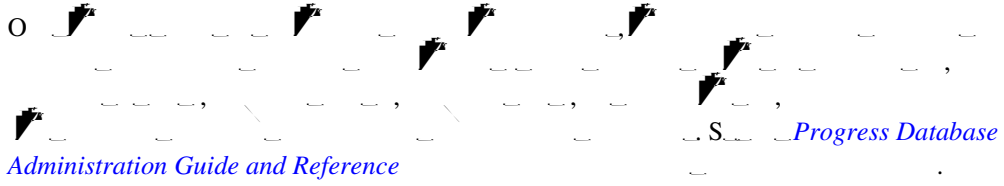
L ’ I C 2, T R N , O , T 3 1.

Table 3–1: Order Table with Derived Column

Order Number (Primary Key)	Order Date	Order Amount	Total After Tax	Cust Num (Foreign Key)
M31	3/19/91	$400.87	$441.74	101
M98	8/13/91	$3,000.90	$3,300.99	101
M129	2/9/91	$919.45	$1,011.39	101
M56	5/14/90	$1,000.50	$1,100.55	102
M37	12/25/90	$299.89	$329.87	103
M140	3/15/91	$299.89	$329.87	103
M41	4/2/90	$2,300.56	$2,530.61	104



3.5 Physical Implementation


Progress Database Administration Guide and Reference

4

Defining Indexes

T

4.1 Overview

4.2 Indexing Databases

T

4.2.1 How Indexes Work

A . T , , . T ; W , , S , , . A *index key* *row pointer* . F 4 1 .

T , , U , , .

E , *primary index*. W 4GL , P

P . F 4 1 O -N ( )

O -D O .

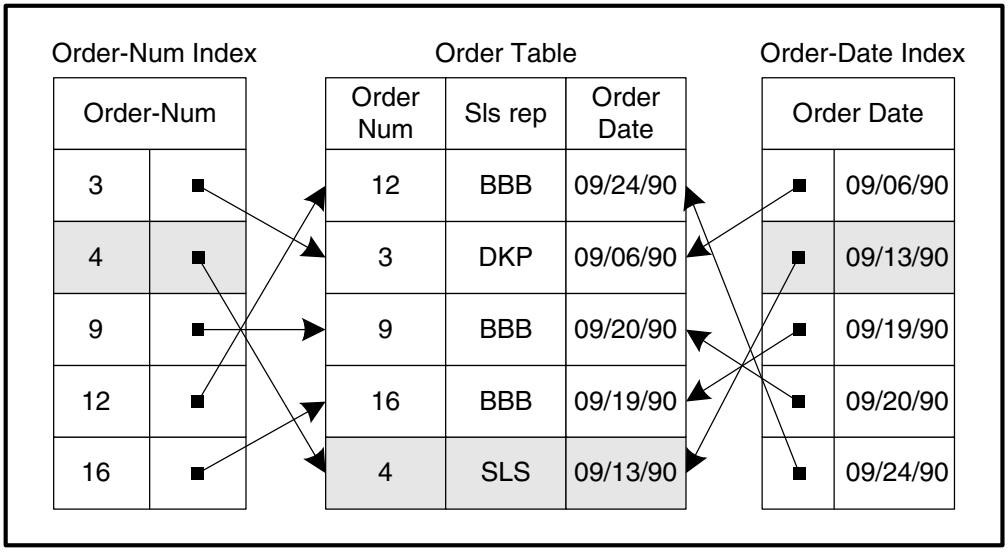


Figure 4–1: Indexing the Order Table

4.2.2 Why Define an Index?

T :

- D .

  I F 4 1, O . I . S .

  H , O N , P (, 1, 2, 3, .).

  F , 4, P . I , **directly** O -N . P . B , , . S , (, 9/13/90). T O . A , , .

- A .

  A . S ( ), *range* . F , , F 09/6/90 09/20/90", .

NOTE: A , . S , , .

- E .

  W , , . F , - 4 - 4, 4 . T - .

- R - .

T ( ) . I , . T .

4.2.3 Indexes Used in the Sports Database

T 4 1 , .

Table 4–1: Reasons for Defining Some Sports Database Indexes *(1 of 3)*

Table	Index Name	Index Column(s)	Primary	Unique
C	-	-	YES	YES
	Why the Index Was Defined: 1. R ’ . 2. R . 3. E ( ). 4. R , .			
			NO	NO
	Why the Index Was Defined: 1. R ’ . 2. R .			
			NO	NO
	Why the Index Was Defined: 1. R . 2. R .			

Table 4–1: Reasons for Defining Some Sports Database Indexes *(2 of 3)*

Table	Index Name	Index Column(s)	Primary	Unique
I	-	-	YES	YES
	Why the Index Was Defined: 1. R . 2. R . 3. E ( ). 4. R - , - - .			
O -	-	- -	YES	YES
	Why the Index Was Defined: 1. E - . . T . 2. R - , .			
	-	-	NO	NO
	Why the Index Was Defined: 1. R - .			

Table 4–1: Reasons for Defining Some Sports Database Indexes *(3 of 3)*

Table	Index Name	Index Column(s)	Primary	Unique
O	-	-	YES	YES
	Why the Index Was Defined: 1. R . 2. R . 3. E ( ). 4. R - , - - .			
	-	- -	NO	YES
	Why the Index Was Defined: 1. R . W , - . 2. E / ( ). 3. R ’ .			
	-	-	NO	NO
	Why the Index Was Defined: 1. R .			

4.2.4 Disadvantages of Defining an Index

E , :

- I . (S C I S .)
- I . W , P . A , P .

D , . F , ( ), .

4.3 Choosing Which Tables and Columns to Index

I , , , . H , , . Y , , . Y ' (19,000 20,000). I . H , (, 100 20,000). T P .

4.4 Indexes and ROWIDs

A ID (ROWID). *ROWIDs* . ROWID . A , , ROWID.

A . E 1024, 2048, 4096, 8192 . B . M . T . P . O , ROWID . P . I . T , .

4.5 Calculating Index Size

Y :

N * (6 + +) * 2

F , 21 (T 4 2) 500 , :

500 * (6 + 1 + 21) * 2 = 29,000

T :

- T
- T
- T , .
- T

H , P . I , 20% 60% .

T . P . T -

NOTE: A , ,

T 4 2 .

Table 4–2: Column Storage *(1 of 2)*

Data Type	Value	Column Storage in Bytes
C	1 + *number of characters*, excluding trailing blanks. If the number of characters is greater than 240, add 3 to the number of characters instead of 1.	
D	3	

Table 4–2: Column Storage *(2 of 2)*

Data Type	Value	Column Storage in Bytes
D_	_	1
	_	2 + (*number of significant digits* + 1)/2
I _ _	_	1
	1	2
	128	3
	32768	4
	8	4
	2,147,483,647	5
L	_	0
	_	1

F 4 2 P .

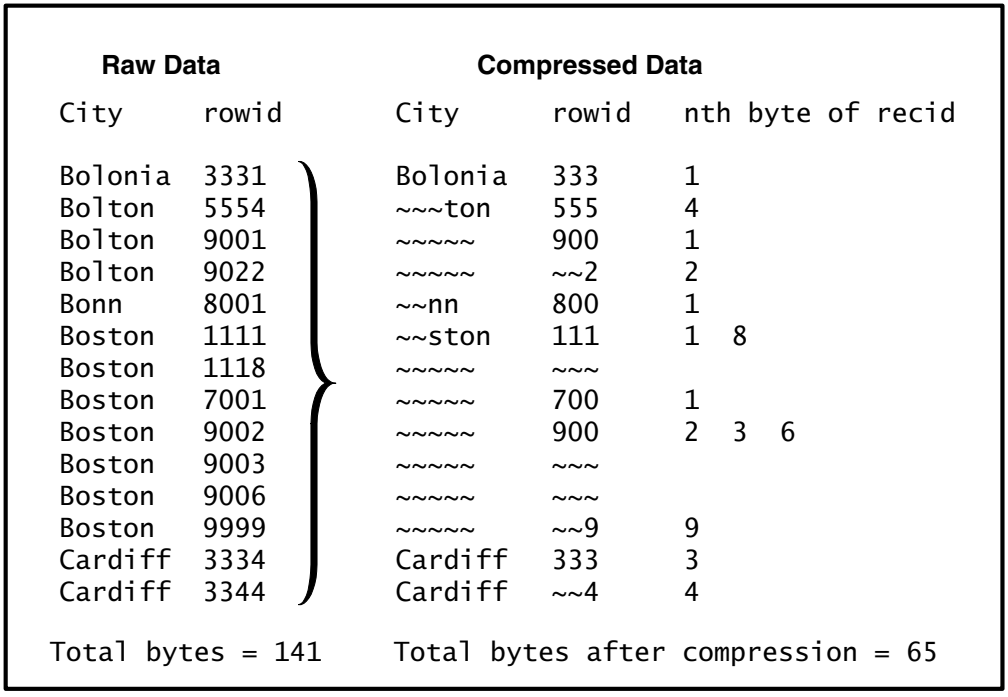


Figure 4–2: Data Compression

T C ROWID . T . F , P . B . T , B , B . I , B . S , P . B . S , B B (B) .

F ROWID , P . I ROWID ROWID . F , P ROWID 333 ROWID, . G B ROWID 1111, ROWID 1118. S (111) ROWID ; (8) .

B , P . I , 65 141 . T ' 54%. A , . Y -.

4.6 Eliminating Redundant Indexes

I , . R .

4.7 Deactivating Indexes

I . I , - , . D . F *Progress Database Administration Guide and Reference*. T SQL-92, *Progress SQL-92 Guide and Reference*.

NOTE: Y . Y .

5

Progress 4GL Index Usage

T _ _ P _ 4GL _ _ _.

5.1 Finding Out Which Indexes Are Used

T P XREF
COMPILE . T SEARCH XREF

T 5 1 XREF :

Table 5–1: XREF tags

Tag	Meaning
SEARCH	I . T . W
SEARCH . . . WHOLE-INDEX	I
SORT-ACCESS	I . A
ACCESS	I
CREATE	I
DELETE	I
UPDATE	I

A .
T C 4 *Progress Database Administration Guide and Reference*.

5.2 Maintaining Indexes Through the 4GL

A , P . P **creates** :

- A P
- A 4GL
- W P VALIDATE
- W P
- A P

P **updates** . B P (UPDATE), P FIND , . P .

NOTE: P .

Y . Y . H , , P .

I , . Y . I , .

R P , .

5.3 Using the 4GL ASSIGN Statement

W 4GL , ASSIGN . T I/O . O , P - .

T ASSIGN :

r-sgn2.p

```
DEFINE VARIABLE neword LIKE order-line.order-num
    LABEL "New Order".
DEFINE VARIABLE newordli LIKE order-line.line-num
    LABEL "New Order Line".
REPEAT:
  PROMPT-FOR order-line.order-num line-num.
  FIND order-line USING order-line.order-num AND line-num.
  SET neword newordli.
  FIND order WHERE order.order-num = neword.
 ASSIGN order-line.order-num = neword
        order-line.line-num = newordli.
END.
```

T - . (I - .) I ASSIGN *field=expression*. S . B - - , .

5.4 Indexes and Unknown Values

I (?), P . W , P . F , - , - 10, P - 10. H ,P . Y .

EXAMPLES

E 1:

```
FOR EACH cust WHERE cust-num >10 AND cust-num <= ?
```

H ZERO AND FALSE:

```
FOR EACH cust WHERE cust-num >10 AND cust-num <= ? AND zip >0 AND zip <?
```

E 2:

```
FOR EACH order WHERE order-date >= 1/1/97
```

H NO :

```
FOR EACH order WHERE order-date >= 1/1/97 AND cust-num = 1
```

5.5 Indexes and Case Sensitivity

T

P JOHN , J , A , JOHN

F , P , J , A , S , JOHN , J , N ,

C , . T ,
.
F ; , ,
I P D PHONE
.
O , P (SMITH = S =).
H , , . F ,
. C P . H , ANSI SQL ,
. O . I
, , - .
C , - .W - , JOHN , J ,
. H , . N
( ) - -
. F , ANSI SQL (-Q)
Progress Startup Command and Parameter Reference.

5.6 How Progress Chooses and Brackets Indexes to Satisfy Queries

K . H , P .
E , . T , P
4GL . T , P
, , .

5.6.1 Background and Terminology

T , , P . F
, WHERE :

SYNTAX

```
WHERE searchExpr [ BY field ]
```

F :

```
WHERE Cust-num > 6
WHERE Name "D"  BY Sales-Rep
WHERE Country = "Zimbabwe"
WHERE Comments CONTAINS "Com*"
```

SYNTAX

```
WHERE searchExpr AND searchExpr [ BY field ]
```

F :

```
WHERE Cust-num > 6 AND comments CONTAINS "ASAP"
WHERE Name = "Rogers" AND Postal-Code BEGINS "017"
```

SYNTAX

```
WHERE searchExpr OR searchExpr [ BY field ]
```

F :

```
WHERE Cust-num > 1000 OR Sales-Rep BEGINS "S"
WHERE Postal-Code =< "01500" OR Postal-Code >= "25000"
```

T BY *field* *sort match.*
A *searchExpr* :

BY *field*	**Sort Match**
field = *expression*	**equality match**
field < / =< / > / >= *expression*	**range match**
field BEGINS *expression*	**range match**
wordIndexedfield CONTAINS *stringExpression*	**equality** ( ) **range** ( ) **none** (>1 ,)

F , R P FOR
Progress Language Reference.

B *search conditions*. C , , *field* . A ,
searchExpr *searchExpr's* AND OR ,
.

T C AND
OR, . P , , (OR)
. T . T .I , ,
XREF , .
SEARCH XREF .

T .

5.6.2 Case 1: WHERE *searchExpr*

I *field* *searchExpr*, *field* - ,P . O , P :

Sample WHERE Clause	Indexes Used
`WHERE Customer.Name BEGINS "B"`	N
`WHERE Customer.Postal-Code BEGINS "01"`	C -N ( )

I *searchExpr* - , P .

I BY *field* , *field* , P
WHERE . I *field* , P
.

5.6.3 Case 2: WHERE *searchExpr* AND *searchExpr*

F WHERE , P
AND. W FOR EACH , AND
- , .
W FIND , AND
, . N
; :

Sample WHERE Clause	Indexes Used
WHERE Customer.Name = "Mary" AND Customer.Sales-Rep = "Higgins"	N S -R
WHERE Comments CONTAINS "small" AND Country = "USA" AND Postal-Code = "01730"	C C -P

I , G R
C S I :

NOTE: I P , , USE-INDEX BY
. I , BY
C -N .

Sample WHERE Clause	Indexes Used
WHERE Customer.Country = "USA" AND Customer.Sales-Rep = "Higgins" BY Cust-Num	S -R

5.6.4 Case 3: WHERE *searchExpr* OR *searchExpr*

F WHERE , P OR. I , OR , , , P :

Sample WHERE Clause	Indexes Used
`WHERE Customer.Comments CONTAINS "to*"` `  OR Customer.Name = "Carlin"`	C N
`WHERE Name > "Beaudette"` `  OR Country > "Zambia"`	N C -P

I , OR CONTAINS (,), P OR:

`WHERE Comments CONTAINS "credit"` `  OR Postal-Code > "01000"`	C C -N

I , OR , P -C C -P , . P , , CONTAINS :

`WHERE Comments CONTAINS "credit"`
`  OR Postal-Code < "01000"`
`  BY Sales-Rep`
	C
	S -R

I , G R C S I .

NOTE: I , OR , P .

5.6.5 General Rules for Choosing a Single Index

W , P ( ) :

1. I CONTAINS ( ), :

Sample WHERE Clause	Indexes Used
`WHERE Customer.Comments CONTAINS "big" AND Customer.Country = "Canada"`	C

2. I , , . I 0 1 :

Sample WHERE Clause	Indexes Used
`WHERE Customer.Cust-Num = 10 AND Customer.Sales-Rep = "DR"`	C -N

3. U . E :

- T ,

AND

- T AND (OR NOT)

T , , 2 3 , 1 2, OR:

Sample WHERE Clause	Indexes Used
`WHERE Customer.Country = "Costa Rica" AND Customer.Postal-Code > "3001" AND Customer.Sales-Rep BEGINS "S"`	C -P

Sample WHERE Clause	Indexes Used
WHERE Customer.Name = "Harrison" AND Customer.Sales-Rep BEGINS "S"	N
WHERE Customer.Name = "Harrison" AND (Customer.Country = "Finland" OR Customer.Country = "Denmark")	N

4. U . F AND . I ,
:

- T
- A

Sample WHERE Clause	Indexes Used
WHERE Customer.Sales-Rep = "ALH" AND Customer.Country = "Italy" AND Customer.Postal-Code BEGINS "2"	C -P
WHERE Customer.Contact = "DLC" AND Customer.Sales-Rep BEGINS "S"	S -R
WHERE Customer.Contact = "Ritter" AND Comments CONTAINS "compute*"	C

5. U . (A .)

Sample WHERE Clause	Indexes Used
WHERE Customer.Country BEGINS "EC" AND Customer.Sales-Rep BEGINS "S" BY Country	C -P
WHERE Customer.Contact = "Wilson" AND Customer.Credit-Limit > 2000 BY Name	N
WHERE Name = "Wilson" OR Customer.Credit-Limit = 2000 BY Sales-Rep	S -R

6. U . T , / :

Sample WHERE Clause	Indexes Used
WHERE Customer.Name = "Samali" AND Customer.Sales-Rep = "BCW"	N
WHERE Customer.Country BEGINS "EC" AND Customer.Sales-Rep BEGINS "B"	P -C

7. U :

Sample WHERE Clause	Indexes Used
WHERE Customer.Contact = "MK" AND (Customer.Sales-Rep BEGINS "S" OR Customer.Sales-Rep BEGINS "B")	C -N
WHERE Customer.Postal-Code >= "01000" AND Customer.City = "Boston"	C -N
WHERE "meaningless expression"	C -N

5.6.6 Bracketing

H , P . T *bracketing*. C , P . T :

- B .
- B , .

T :

Sample WHERE Clause	Indexes Used	Brackets
`WHERE Contact = "DLC" AND (Sales-Rep BEGINS "S" OR Sales-Rep BEGINS "B")`	C -N	N
`WHERE Postal-Code >= "01000" AND City = "Boston"`	C -N	N
`WHERE Name = "Harrison" AND Sales-Rep BEGINS "S"`	N	N
`WHERE Contact = "DLC" AND Sales-Rep BEGINS "S"`	S -R	S -R
`WHERE Country BEGINS "EC" AND Sales-Rep BEGINS "S" BY Country`	C -P	C -P
`WHERE Comments CONTAINS "big" AND Country = "USA" AND Postal-Code = "01730"`	C C -P	C P -C

T . T , .

- A AND.
- A OR OR ( ), P .

- W , AND , (WHERE CONTAINS * & *) (WHERE CONTAINS * AND CONTAINS *).
- A WHERE OR - (WHERE CONTAINS OR 2 = B).

5.7 Index-related Hints

- A
- A ;
- M , , P ,

6

Progress 4GL Word Indexes

P . A *word index* . S . Y D D . T D D .

6.1 Word Index Support

T , CONTAINS WHERE
R . T FOR EACH . F ,
C . T
H :

```
FOR EACH customer WHERE address CONTAINS "Homer":
```

T WHERE CONTAINS :

SYNTAX

```
WHERE field CONTAINS string-expression
```

field

R

string-expression

C *field:*

SYNTAX

```
"word [ [ & | | | ! | ^ ] word ] ... "
```

word

A . A
(*). Y
. F , *
,
.

& | | | ! | ^

W . T (&) AND; ( ),
(!), ( ) OR. S
. Y
AND OR
.

EXAMPLE

T :

T AND (&) OR ( ). F , . T .

EXAMPLE

T . I (,), AND :

B

:

```
FOR EACH accident WHERE city = "Boston" AND
        description CONTAINS "milk (truck ^ trailer)":
```

NOTE: U S A (-) . F , S A (-) *Progress Startup Command and Parameter Reference*.

6.2 Word Delimiters

P . P . Y P .

P :

- **LETTER** A .
- **DIGIT** A .
- **USE_IT** A .
- **BEFORE_LETTER** A LETTER . I , .
- **BEFORE_DIGIT** A DIGIT .

- **BEFORE_LET_DIG** A
  LETTER DIGIT .
- **IGNORE** A . S .
- **TERMINATOR** A .

6.2.1 Progress Defaults

T P 4GL :

- T LETTER . I E , A-Z - - .
- T DIGIT 0-9.
- T USE IT : ($), (%), (#), (@), ( ).
- T BEFORE DIGIT : (.), (,), (-). T , , 12.34" , .
  .
- T IGNORE (’). T , , , J ’ J .
- T TERMINATOR .

T CONTAINS . F , J ’ CONTAINS , J . I 8:30 : 8 30 . T , 8:30 , 8 C S . 7:30 . A , CONTAINS . T CONTAINS .

6.2.2 Defining Your Own Delimiters

T , , *word-break rules file*. W , . T -:

```
#define    AT_SIGN    64
#define    PERIOD     46
#define    COMMA      44

word_attr  =
{
PERIOD,      BEFORE_DIGIT,
COMMA,       BEFORE_DIGIT,
0x2D,         BEFORE_DIGIT,      /*hyphen */
39,          IGNORE,            /*single quote */
'$',          USE_IT,
'%',          USE_IT,
'#',          USE_IT,
AT_SIGN,      USE_IT,
'_'          USE_IT
};
```

N # . W ('), ASCII , (0 2D,). E .

Y - . B , LETTER , 0 9 DIGIT , TERMINATOR . Y .

NOTE: T (*), ( ), (!), ( ), CONTAINS . Y **must** TERMINATOR . Y , , , OR .

A ' - , PROUTIL :

Operating System	Syntax
UNIX Windows	`proutil` *`database`* `-C wbreak-compiler` *`src-file rule-numb`*

I , *database* , *src-file* - *rule-numb* 1 255 .

T PROUTIL .*rule-numb*. F , - 34, .34. T , $DLC PROWD*rule-numb* . F , , PROWD34 .34 . (N PROWD .)

T - , , - PROUTIL :

Operating System	Syntax
UNIX Windows	`proutil database -C word-rules rule-numb`

T *rule-numb* . T , 0 *rule-numb*.

I - , . T , , P . Y . Y PROUTIL . F PROUTIL , PROUTIL *Progress Database Administration Guide and Reference*.

P (CRC) - . I , P . T . Y . N .

6.3 Word Indexing External Documents

I , P . Y .

6.3.1 Indexing by Line

T

Y

T

```
DEFINE VARIABLE words AS CHAR FORMAT "x(60)"
        LABEL "To find document lines, enter search words".

REPEAT:
    UPDATE words.
    FOR EACH line WHERE line_text CONTAINS words:
      DISPLAY line.
    END.
END.
```

T

6.3.2 Indexing by Paragraph

I

600 10

W 60

Y

6.4 Word Indexing and Non-Progress Databases

W P ; ,
. T -P
P 4GL -P
P .

Y . D
-P . Y
.

6.5 Word Indexing and SQL-92

C , SQL-92. T ,
CONTAINS .

NOTE: SQL-92
. T
4GL .

7

Constraints and Indexes Using SQL-92

T

7.1 Constraints

Y , , SQL-92 .
C .

7.1.1 Keys

T :

A *primary key* ( ) . B , . Y . P . T . F C 1, I .

A *foreign key* ( ). I . C 2, T R N , .

W SQL-92, . F SQL CREATE TABLE *Progress SQL-92 Guide and Reference*.

7.1.2 PRIMARY Constraint

U . T .

7.1.3 UNIQUE Constraint

U . T .

7.1.4 FOREIGN Constraint

T . E .

7.1.5 NOT NULL Constraint

U NOT NULL .

7.1.6 CHECK Constraint

T CHECK . Y , . T CHECK .

7.2 Indexes

A - . F . R .

T . I .

A . A . A .

A :

- F . I .
- R . N , .
- W , . T .
- A (, P **and**).
- E .

T SQL-92 CREATE INDEX . T . F CREATE INDEX *Progress SQL-92 Guide and Reference*.

Y . A .

A .

7.2.1 SQL-92 Notes

- T
- I UNIQUE ,
- A SQL-92 .

8

Progress 4GL Triggers

A 4GL . A

T P 4GL .

8.1 Trigger Definition

A 4GL 4GL . A . F , WRITE .

B , . F , , . T ( ) ( ). A , . O .

F P 4GL , *Progress Language Reference* *Progress Programming Handbook*.

8.2 4GL Database Events

D . W , . P . F , , DUMP , P DUMP .

H , P - . R - . F - , *Progress Programming Handbook*.

T P 4GL .

8.2.1 CREATE

W P CREATE INSERT , , P , , CREATE , , REPLICATION-CREATE .

8.2.2 DELETE

W P DELETE , P DELETE , , REPLICATION-DELETE , , , .

8.2.3 FIND

W P FIND GET FOR EACH , P FIND . FIND , WHERE . FIND CAN-FIND .

N FIND , P . I FIND FOR EACH , P . I BREAK PRESELECT (P), P FIND FIND, FIND . T , P , FIND .

8.2.4 WRITE

W P , P WRITE , REPLICATION-WRITE . P . Y VALIDATE . I , WRITE (WRITE). P WRITE ( ).

8.2.5 ASSIGN

W P , P ASSIGN . U , . ASSIGN . T - . I (, UPDATE), P ASSIGN . I , P (NO-UNDO). F - , *Progress Database Administration Guide and Reference*.

8.3 Schema and Session Database Triggers

P 4GL : . A . , D D , . S .

A , .

8.3.1 Schema Triggers

Y T F P D D . W D D , , . P . T . F D D , , *Progress Basic Database Tools* (C) , , D D . F , , *Progress Programming Handbook*.

8.3.2 Differences Between Schema and Session Triggers

A , . T ; . B , , .

S . S , , , , .

S , , . U . F , , - .

U .
B ERRORS . F ERROR RETURN , , *Progress Language Reference*.

8.3.3 Trigger Interaction

Y / / . H .

O , , (FIND , FIND). I , . F WRITE, DELETE, CREATE, ASSIGN , . F FIND , , .

8.4 General Considerations

W 4GL , .

8.4.1 Metaschema Tables

P ( ). Y .

8.4.2 User-interaction Code

P 4GL , . H , .

F , , , , . A , ESQL, .

NOTE: 4GL - (ESQL-89) .

8.4.3 FIND NEXT and FIND PREV

Y (FIND NEXT FIND PREV) .

8.4.4 Triggers Execute Other Triggers

A . F , ASSIGN ,

ASSIGN . Y . F , A , B . T B , .

8.4.5 Triggers Can Start Transactions

B , CREATE, DELETE, WRITE, ASSIGN . (T .)

T , . H , FIND . T , FIND ;

8.4.6 Where Triggers Execute

D (-) , P A S 4GL . F P A S , *Building Distributed Applications Using the Progress AppServer*.

8.4.7 Storing Trigger Procedures

Y ’ P 4GL - .

8.4.8 SQL Considerations

B P 4GL SQL, . F SQL , *Progress SQL-92 Guide and Reference*.

NOTES

- SQL-92 4GL .
- 4GL SQL .
- T , , SQL , SQL .

9

Java Stored Procedures and Triggers

A J SQL-92 . S .

A . T .

T .

9.1 Java Stored Procedures

A J SQL-92 . S . T , / . S .

J P 4GL .

9.1.1 Advantages of Stored Procedures

S SQL-92 J .

S .

9.1.2 How Progress SQL-92 Interacts with Java

P J SQL-92 .

W , SQL-92 J , J , .

W , SQL-92 J V M (JVM) .

F *Progress SQL-92 Guide and Reference*.

9.2 SQL-92 Triggers

A . T .

A J J SQL-92 (INSERT, UPDATE, DELETE) ’ .T P J .

T . L ,

D J . F , . I , . J

9.2.1 Typical Uses for Triggers

Cascading Deletes

A

9.2.3 Trigger Types

T J : .

S SQL-92 . T SQL-92 . R .

EXAMPLE

T :

```
CREATE TRIGGER trig1 BEFORE DELETE ON customer
FOR EACH STATEMENT
BEGIN
   SQLIStatement upd_stmt
     ("UPDATE summary SET trans_tot = trans_tot + 1");
   upd_stmt.execute ();
END
```

NOTE: T UPDATE .

EXAMPLE

T :

```
CREATE TRIGGER trig1
BEFORE INSERT ON customer REFERENCING NEWROW
FOR EACH ROW
BEGIN
  String new_name; Integer new_num;
  new_num = (Integer) NEWROW.getValue(1,INTEGER);
  new_name = (String) NEWROW.getValue(2,INTEGER);
  SQLIStatement ins_stmt ("INSERT INTO customer2 (?,?,?,?,?,?) ");
  "UPDATE summary SET trans_tot = trans_tot + 1");
  ins_stmt.setParam (1, new_num);  ...
  ins_stmt.execute ();
END
```

NOTES

- T INSERT .
- I , FOREIGN KEY .

9.2.4 Differences between 4GL and Java Triggers

T P J 4GL .

- SQL 4GL-
- 4GL J
- 4GL SQL

A , 4GL SQL , .

Restrictions

T J . J :

- DDL
-
- J
- OS
- ASSIGN FIND

9.3 Triggers versus Stored Procedures

T . T :

- T .

  W (INSERT, UPDATE DELETE) , J . A .

- T .

  S , .

- T .

  T .

9.4 Triggers versus Constraints

T . I ,

T :

- T .

  C . W ; .

- T .

  B , . T . H , .

Index

I

K

M

N

O

P

R

S

T