Strategy for Using Aliases

runtot2.p

```
/* runtot2.p - connect to two databases and create an alias part 2 */

define var baltot as decimal format ">>,>>>,>>9.99"
                              column-label "Subsidiary!Customer!Balance".
define var grandbal as decimal.
define var curphydb as char format "x(30)" column-label "Physical!Database".
define var curlogdb as char column-label "Logical!Database".
define var i as int.

connect sports1 -ld subsid1 -1.
connect sports2 -ld subsid2 -1.

do i = 1 to num-dbs:

  create alias nowdb for database value(ldbname(i)).
  run custtot.p (output baltot).
  grandbal = grandbal + baltot.

  display pdbname("nowdb") @ curphydb
          ldbname("nowdb") @ curlogdb
          baltot with down.
  down.

end. /* do i = 1 to num-dbs */

display "Conglomerate Total" @ curphydb
        grandbal @ baltot.

disconnect subsid1.
disconnect subsid2.
```

Physical Database	Logical Database	Subsidiary Customer
sports1	subsid1	1,378,202.00
sports2	subsid2	1,378,202.00
Conglomerate Total		2,756,404.00

Strategy for Using Aliases

■ For example, suppose you have two databases, **subsid1** and **subsid2**. Further, suppose that you want run a consolidated customer report for both databases called **custtot.p**. The program **runtot.p** is going to use the **nowdb** alias before running **custtot.p** with each database.

1) Connect to one database using the logical name of **nowdb**.

2) Compile the program **custtot.p** and save it. The **nowdb** logical database reference will then be stored in **custtot.r**.

3) Disconnect from the **nowdb** database.

4) Connect to the databases with logical names **subsid1** and **subsid2**.

5) Run **runtot.p** program.

■ In **runtot.p**, we introduce the functions **PDBNAME** and **LDBNAME**, used to display the current physical and logical names, respectively.

■ In **runtot2.p**, we introduce an additional function, **NUM-DBS**, and use **ldbname(i)** in the **CREATE ALIAS** statement to use the same alias to connect to different logical databases.

PROGRESS Syntax

PDBNAME function

```
PDBNAME(integer-expression|logical-name|alias)
```

LDBNAME function

```
LDBNAME(integer-expression|logical-name|alias|BUFFER bufname)
```

NUM-DBS function

```
NUM-DBS
```