

Database
Management
System **LINTER**®

Version 5.9

Interactive SQL

Relational Expert Systems



Table of Contents

Introduction	4
Program Execution Requirements	5
Program Execution	6
Launch	6
Terminating INL	6
Commands	7
HELP.....	9
DBINFO	9
SHOW.....	9
Change User	10
Suspend Execution	10
Toggle Commands.....	10
TIME	11
PAGE.....	11
FORMAT	11
COUNT.....	11
BROWSE	12
OPTIMISTIC.....	12
EXCLUSIVE	13
Output Formatting Commands.....	13
HEADER	13
UNLOAD	13
File Handling Commands.....	13
OUTFIL.....	14
Underscore ' _ '	14
EDIT.....	15
Query Visualization & Execution Commands.....	15
Comment.....	15
LIST	16
EXEC	16
SQL Query Commands.....	16
Transaction Completion Commands	16
Working with BLOB Data	16
Shell Commands	17
INL Messages	18
Informational Messages	18

Error Messages	18
User Errors.....	18
System Errors	18
Examples	19
Example 1	19
Example 2	20
Example 3	20
Example 4	21
Example 5	22
Example 6	22
Example 7	23
Example 8	23
Example 9	24

Introduction

The document describes the user interface for Linter's interactive SQL. Control, informational, and interface adjustment commands are included. Examples of INL sessions are provided in section "Examples". This document is intended for programmers and users who know how to work with SQL.

In addition to writing queries, INL is used to debug SQL queries when developing applications, to train users in SQL language query composition and to obtain query execution timing characteristics.

Program Execution Requirements

Prior to launching INL, the Linter kernel:

- must be active;
- must have at least one free channel.

Program Execution

Launch

To start INL from the command line, use the following syntax:

```
INL [-u <User_Name> [ / <User_Password > ] ]  
[-n <Server_Name>]  
[-f <File_Name>] [-t]
```

where the options are:

- u** user name and password separated by a slash and without spaces. If the password is omitted, will treat it as a string of 18 spaces.
- n** node name for the system on which the needed Linter server is installed. Not applicable if using local host.
- f** file name of SQL script intended for execution;
- t** time option to switch off reporting execution time statistics.

Defaults:

- u** if omitted, you will be prompted for name and password.
- n** the default server is assumed.
- f** none. You will have the SQL prompt and be in interactive mode.
- t** execution time statistics will not be displayed after queries have been processed.

Microsoft users can access INL from the ldba and desktop utilities, see documents “Administration: The LDBA Utility” & “The Linter Desktop Utility”. All operating systems can launch INL from the command line by just typing its name:

```
INL<ENTER>
```

Without using any options, the INL program begins with a logon dialogue.

```
Interactive Language v.5.9 for RDBMS Linter SQL v.5.9  
Copyright (C) 1995-2006 Relex, Inc. All rights reserved.  
  
Enter user name:SYSTEM  
Enter password :  
SQL>
```

Screen 1 – Logon to INL

In this example, the first password entered contained an error. Note that password characters aren't echoed.

Terminating INL

INL is closed by entering the EXIT command or pressing CTL+C.

Commands

In this document INL and SQL commands are shown in CAPITALS. In use, these commands are case insensitive. Commands are entered at the INL or SQL prompt and consist of the command name and any appropriate argument to the command.

The following table lists the INL commands.

Since the purpose of INL is the creation of SQL queries, INL recognizes SQL statements. SQL statements, but not INL commands, must be terminated with a semi-colon.

<u>Command and Argument or Toggle</u>	<u>Description</u>
CTL+C	Exit INL and close all buffers; same as EXIT.
!<comment>	In either INL or a script, an exclamation point in the first position of a line marks the entire line as a comment..
_<scriptname>	To run a script, precede it with an underscore. scriptname must have an .sql or .inl suffix.
BROWSE<tog>	Optional freeform browsing mode .
CODEPAGE<page>	For Microsoft, set code page to 866, 1251, or KOI8.
COMMIT	SQL write changes to DB.
COUNT<tog>	Append number of rows processed to output.
DBINFO	Display information about the DB's settings.
EXCLUSIVE	Lock/unlock rows during transaction.
EDIT	Load commands and arguments of most recently executed SQL request and switch to edit mode.
EXEC	Repeat execution of the most recently run SQL statement or execute the just-edited SQL statement.
EXIT	Shut down and close all buffers; same as CTL+C.
FORMAT<tog>	ASCII / binary output.
HEADER<text>	Insert <text> as first line of output.
HELP	List all commands with brief descriptions.
LIST	Print current INL settings.
OPTIMISTIC<tog>	Optimistic/pessimistic concurrency control.
OUTFIL<filename>	Output to <filename> instead of screen.
PAGE<tog>	Display 20 rows of output and prompt for next 20.
SH <space><command>	Execute shell command. On completion, returns to INL.
SHOW<table name>	Display structural information about a specified table.
SLEEP<seconds>	Suspend execution for specified number of seconds.

<u>Command and Argument or Toggle</u>	<u>Description</u>
TIME<tog>	Timestamp query execution and display results.
UNLOAD<char>	Specifies output column separator. <char> may be any printable or unprintable character.
USERNAME	Logon with another user name.

HELP

HELP displays a list of INL commands and the more frequently used SQL statements.

DBINFO

DBINFO returns information about the DB's runtime and other parameters.

Command syntax

DBINFO

The following table shows the information is displayed by DBINFO. This example is from the DEMO database.

```
Version                : 5.9
Kernel memory         : 200
Sort memory           : 100
Channels queue        : 100
Table queue           : 100
Column queue          : 100
Files queue           : 30
Users queue           : 100
Transaction control   : on
Log file              : off
```

SHOW

SHOW returns information about either a named database table or all all tables in the DB.

Command syntax

```
SHOW {<Table_name> | %}
```

In constructing <Table_name>, you may use the wildcards %, for multiple character substring matches and the underscore, _, for single substring matches.

To get information about all tables, use SHOW %.

The following table shows the table information returned. The example is from the DEMO database.

```
Description of table "AUTO"
* ROWID limit                : 1022
* Last occupied ROWID       : 1001
* Current row number        : 1001
* Page fill percent         : 100
* Row length                 : 113
* Number of columns         : 13
* Number of indexes         : 2
* Index files                : 1      (SY00 10)
* Data files                 : 1      (SY00 37)
```

```

INL: press any key (q for exit):
Column characteristics
MAKE CHAR (20) B-TREE FILE : 1 level : 1 top :7
MODEL      CHAR (20)
...

```

For every table column, the following information is displayed in the Column characteristics section:

- 1) column data type and, depending on data type: column length, data precision, and scale;
- 2) column attributes such as primary key, reference key, and AUTOINC;
- 3) the existence of a column index and index parameters. The level number indicates the number of index file pages that will be read preliminary when searching for values in the column. The top number shows the relative number of the index file page where the top of the index column resides.

Change User

USERNAME lets you login with another user name.

Command syntax

```
USERNAME <Username>[/Password]
```

If <Password> is omitted, you will be prompted for it. If <Username> or <Password> are entered incorrectly, you will receive an error message:

```
illegal user name or illegal password.
```

Suspend Execution

SLEEP suspends execution suspension for a specified number of seconds.

Command syntax

```
Sleep <Time>
```

<Time> is the wait time in seconds. The default is zero.

On command execution you will get the following screen message:

```
INL: Pause for n sec.
```

and program execution will be suspended for n seconds.

Toggle Commands

The following commands belong to a group that toggle a condition on and off or toggle between two states:

TIME	Timestamp query execution and display results.
PAGE	Display 20 rows of output and prompt for next 20.
FORMAT	ASCII / binary output.
COUNT	Append number of rows processed to output.

BROWSE	Optional freeform browsing mode.
OPTIMISTIC	Optimistic/pessimistic concurrency control.
EXCLUSIVE	Lock/unlock rows during transaction.

You can see the state of these toggle switches by using the LST command.

TIME

If time switch is toggled ON, INL marks the time, when the query is transmitted for execution and when the first answer to that query is received. On completion of query output, the elapsed time is shown at the top of the output.

When toggled OFF, time measurement is disabled.

Initially, TIME is toggled ON.

PAGE

When toggled ON, output is paged in blocks of 20 rows. After each 20 page block, the following prompt is displayed:

```
INL : press any key (q for exit) :
```

Typing q, for quit, terminates output and returns the SQL prompt.

When toggled OFF, output scrolls without interruption until complete.

Immediately after starting INL, PAGE is ON.



When receiving commands from a file instead of the terminal, this toggle is ignored and output cannot be divided into pages.

FORMAT

For readability, INL usually displays answers in character format. If FORMAT is toggled ON, INL displays the answers in binary form.

This option is useful when transmitting the answers to a file for processing by other tools.

Initially, FORMAT is OFF.

COUNT

When COUNT is toggled on, its initial state, all INL answers are followed by additional statistical messages, e.g.:

```
INL : number of rows deleted : 23
```

or

```
INL : number of rows shown : 12
```

If the messages become obnoxious, toggle COUNT to OFF.

BROWSE

By default, BROWSE is toggled OFF and INL displays query answers in a table with columns separated with the delimiter specified by UNLOAD. All answers are displayed strictly sequentially and in the order specified by the query.

The negative aspects of this format include:

- 1) You cannot view rows previously scrolled off the screen;
- 2) Rows that are longer than the screen width are wrapped to the next line but column alignment goes to pot. The resulting display is difficult to read.

Toggling BROWSE to ON will output query answers into a window that can be scrolled in any direction.

In BROWSE mode INL first fetches as many rows as memory allows into its own buffer and then outputs them into the convenient window.

Such a form is the most convenient for viewing answers, but it is somewhat slower, especially during initial answer reception.

MAKE	MODEL	BODYTYPE	CYLNDERS	HORSE
AMERICAN MOTORS	MAIADOR STATION	STATION WAGON	8	150
FORD	MERCURY MONTEREY	SEDAN HARDTOP	8	208
FORD	LTD COUNTRY SQUIRE	STATION WAGON	8	163
MERCEDES-BENZ	280 SE	SEDAN	6	158
LOTUS	ELAN SPRINT	COUPE	4	80

Screen 2 – Results Displayed in Browse Mode

The arrow keys enable movement in any direction. In addition, the ENTER key moves forward from column to column. You may have one or two scrollbars but their effect is minimal.

To close the window and return to INL, press the ESC key.

OPTIMISTIC

By default, Optimistic Concurrency Control is OFF.

When OPTIMISTIC is toggled ON Linter is able to generate a "warm" restart.

EXCLUSIVE

The EXCLUSIVE mode is recommended for working with multi-joined tables and/or distributed transactions, i.e., transactions that include data updates on more than one node.

In this mode, updated rows are locked and all modifications are written immediately to the database. Other users cannot read and update rows locked by the EXCLUSIVE initiator until the entire transaction is complete and the rows are unlocked.

After successful execution of the COMMIT command, modifications performed by the transaction remain a permanent part of the database.

After execution of the ROLLBACK command modifications performed by the transaction are removed from the database.

Output Formatting Commands

Two options are available in INL to alter the appearance of returned data:

HEADER: defines a header for the output, and

UNLOAD: specifies the column delimiter for output, by default, the vertical bar.

Both of these commands are followed by a colon, :, and have one parameter only.

HEADER

By default, the header shows column names over each column. A customized multi-line header may be created with the HEADER command.

To indicate a continuation of the header onto the next line, a hyphen, -, character is put at the end of the line.

Example

```
SQL>HEADER: Ford owners in Pasadena: -  
Owner name Ford Model Year
```

UNLOAD

UNLOAD specifies the delimiter to be used in separating columns in output. The syntax is:

```
UNLOAD <char>
```

By default, <char> is the pipe symbol |, but may be any printable or non-printable character.

See example of UNLOAD use in section "Unload".

File Handling Commands

The program works with files of two types: the output file of returned answers and the input file of queries. In INL, the user terminal is set as the input and output file by default.

OUTFIL

OUTFIL is used to send output to a file on disk. The syntax is:

```
OUTFIL: <filename>
```

In the output file the following may be written:

- 1) an execution timestamp if TIME is toggled on;
- 2) answers to the query binary or char format, see FORMAT;
- 3) a header specified by HEADER;
- 4) a count of the rows returned if COUNT is toggled on.

Example

```
SQL> Outfil: /usr/home/Full.ans
```

Answers to subsequent queries will be written to the file specified by OUTFIL until a new specification is given. Then, the existing file is be closed and a new output file will be opened.

Comments are always output to the terminal screen.

OUTFIL can be issued from the terminal or from a file.

Underscore ‘_’

The underscore, _, indicates that the following filename is a script containing INL commands and/or SQL statements. The syntax is:

```
_  
_<script.sql>
```

If a file extension is not specified, INL searches for a file with an .sql extension.

Example

```
SQL> _/usr/home/step9.sql
```

INL will execute in sequence, each command or statement in the specified input file. When the program reaches the end of the file, it automatically returns to terminal input with the SQL prompt.

The script file is an ordinary text file that can be written with any text editor available in the operating environment. It can also be generated by any program.

Commands and statements are typed just as they would be for terminal input, e.g.:

```
COUNT  
OUTFIL:   paper.txt  
  
UNLOAD:  |  
  
HEADER:   List of Models
```

INL includes the EDIT tool for preparing scripts without exiting to the operating system shell.

EDIT

EDIT has no parameters. On the command, INL passes the contents of the `inl.buf` file in the current directory to the editing file.

For Microsoft, a Notepad window is opened.

If no prior command have been executed in the current INL session, the editing file is empty. Otherwise it contains the last executed set of commands and statements.

Each time a query is executed INL creates the text file `inl.buf` that contains the statements and queries from the currently-executed script or terminal input. The `old.buf` file is deleted.

Modify the existing script or clear the existing text and enter new commands and statements.

For non-Microsoft systems, after query input, a semicolon at the end of the line terminates the query. Placing the cursor after the semicolon and pressing ENTER exits editing and returns you to the SQL prompt. For Microsoft systems, close the Notepad window to return to INL.

Now, enter EXEC at the SQL prompt to run the new or revised script.

After any execution the query buffer is not erased. The query may be repeated by using the EXEC command.

Query Visualization & Execution Commands

Comment

During a batch file execution, it is often desirable to output, to the terminal messages about the current execution.

The comment command `!`, is place in the first position on any line. The text follows on the same line. That text will be output to the terminal by INL during script execution. These output comments are useful when logging execution of series of queries. For example, if the `q.sql` file contents be the following:

```
outfil: answer.ans
! *** execute query q1 .sql
_q1 .sql
! *** execute query q2.sql
_q2.sql
! *** execute query q3.sql
_q3.sql
```

the following shows the command to launch and the screen output during execution:

```
SQL> _q
*** execute query q1.sql
*** execute query q2.sql
```

```
*** execute query q3.sql
SQL>
```

LIST

The LIST command takes no arguments. On execution, it displays selected INL settings, the output column delimiter, and if any, the output file and the contents of the query buffer. The following table show typical output.

```

-- Current INL switch status --

time           : on.
page:          : on.
format         : off.
count          : on.
optimistic     : off.
exclusive      : off.
priority       : 0
browse         : off.
outfil         : /develop/test.out
codepage       : MS-DOS 866 [Microsoft only]
unload        :
query         : select * from auto;
```

EXEC

Query execution can be started by either the EXEC command or by typing the terminating semicolon for the query and hitting the ENTER key.

The EXEC command has no parameters. It may be used in a script file.

On receiving the command, INL sends the last typed or executed queries for execution.

SQL Query Commands

The following SQL query statements are available in INL. They need to be issued in conformity with the syntax rules described in document «Linter's SQL».

```
ALTER    CREATE    DELETE    DROP    GRANT    INSERT PRESS
REBUILD  REVOKE    SELET    UPDATE
```

Transaction Completion Commands

These SQL commands are used to save or cancel modifications to the database made by the current transaction. COMMIT writes the modifications to disk. ROLLBACK cancels the modifications.

Working with BLOB Data

To view, load, unload, or delete BLOB data, BROWSE must be toggled ON. See section "Browse".

The row in which the BLOB operation is to be performed is chosen using the SELECT statement. If BLOB data is to be added in a new row of the table an INSERT statement must have been previously executed. That INSERT statement will have omitted any value for the BLOB column or have set it to NULL.

The browsing window that is presented following execution of the SELECT statement functions the same as in ldba. See document "User's and Administrator's Guide".

Shell Commands

INL permits execution of any valid operating system command without exiting INL. The SH command is used for this purpose and has the following syntax:

SH <OS command>

After execution of the shell command is complete, control returns to INL.

INL Messages

Informational Messages

Informational messages contain information about the program functioning. The Sense of the messages is usually obvious in the context of the session.

Error Messages

Error messages are divided into two groups: errors caused by invalid user commands, and system software errors. In the first case, the user should analyze the error message, fix the error, and retry command execution. In the second case, the system administrator should be consulted.

User Errors

INL : buffer overflow
INL : can't open output file %s. Error code %d
INL : user name not present in command line
INL : command file not present in command line
INL : server name not present in command line
INL : server name too long
INL : illegal user name
INL : illegal password
INL : channel not open. Linter wasn't loaded
INL : command file already active
INL : illegal command
INL : file name too long
INL : illegal syntax
INL : relation name isn't define
INL : illegal synonym name

System Errors

INL : channel not open. Error code %d/%d
INL : command file %s not open. Error code %d
INL : channel closing error. Error code %d
INL : operating system error. Error code %d
INL : channel opening error. Error code %d
INL : commit error. Error code %d
INL : rollback error. Error code %d

Examples

The following examples illustrates the use of INL commands. The PERSON, AUTO and FINANCE tables are used by these examples, and are contained in the DEMO database included in the Linter distribution.

<u>PERSON</u>	<u>AUTO</u>	<u>FINANCE</u>
name	make	crditercd
first name	model	crditlim
initial	bodytype	crntbal
sex	cynders	oilcrd
agse	horsepwr	networth
fmlstat	dsplcmnt	crditrat
depdnts	weight	inscmpny
numb	color	pleyamt
street	year	college
city	serialno	onvactn
state	chkdate	invstmnt
zip	chkmile	personid
phone	personid	
job		
salary		
cmission		
yrseeduc		
yrscmpny		
vacdays		
sicdays		
hobby		
personid		

Example 1

Get a list of the last names of auto owners in Pasadena:

```
SQL>SELECT Name FROM Person WHERE City='PASADENA';
INL : start time : 20:38:19 end time : 20:38:19
NAME
COLVILLE
TANIMOTO
DAVENPORT
PRATT
```

```
COHEN
DAVENPORT
INL:      number of rows shown: 6
```

Example 2

Get a list of auto owners with a salary of more than \$40000 who are not older than 40 years.

```
SQL>TIME
SQL>SELECT Firstnam, Name FROM Person
1>WHERE Salary >40000 AND Age <=40;
FIRSTNAM      NAME
ANN            HELL
CLARE         KIM
WILLY        STOGNIY
JOHN          ALDEN
MARY         DAVENPORT
JOHN         MOUREAU
ETHEL        PAPAYANOPOULO
TIMOTHY      RAY
ANNETTE     SCHOEN
RICHIE      GOSDEN
INL: number of rows shown: 10
SQL>
```

The answer to this query is formed by values from the Firstnam and Name columns of the table Person chosen from the rows matching to the conditions contained in the WHERE clause.

Example 3

Get a list of Ford auto owners whose names begin with D and the models of their autos.

```
SQL>TIME
SQL>SELECT FirstNam, Name, Model FROM Person, Auto
1>WHERE FirstNam LIKE 'D%' AND Make ='FORD' AND
2>Person.PersonId =Auto.PersonId;
INL : start time : 14:32:20 end time : 14:32:22

FIRSTNAM      NAME      MODEL
DIANA         HEAFNER  MUSTANG BOSS 351
DEBORAH       SPIEGEL  GRAN TORINO SPORT V8
DIANA         WATSON   PINTO RUNABOUT
DAISY         WOOLSEY  LTD COUNTRY SQUIRE
DIANA         COLVILLE GRAN TORINO SPORT V8
DANIEL        SRC      MERCURY MONTEREY
DAN           TANIMOTO CAPRI RS 2600
DANIEL        ALDEN    MERCURY MONTEREY
INL number of rows shown:      8
```

SQL>

Note that the typed in query remains in Query buffer. This query can be repeatedly executed with the EXEC command.

```
SQL>EXEC
In1 : start time : 14:34:14 end time : 14:34:15
FIRSTNAM      NAME          MODEL
DIANA         HEAFNER      MUSTANG BOSS 351
DEBORAH       SPIEGEL      GRAN TORINO SPORT V8
DIANA         WATSON       PINTO RUNABOUT
DAISY         WOOLSEY      LTD COUNTRY SQUIRE
DIANA         COLVILLE     GRAN TORINO SPORT V8
DANIEL        SRC          MERCURY MONTEREY
DAN           TANIMOTO     CAPRI RS 2600
DANIEL        ALDEN        MERCURY MONTEREY
INL number of rows shown:      8
SQL>
```

The 1 sec. difference in execution time for the same query occurs because a part of the needed information is already in Linter kernel memory and does not require additional processing.

Example 4

Output a list of employees whose salary is equal to the lowest salary paid.

```
SQL>SELECT DISTINCT FirstNam,Name,Salary FROM Person
1>WHERE Salary =(SELECT MIN(Salary) FROM Person);
```

```
In1 :      start time :
FIRSTNAM      NAME          SALARY
ANNA          DEKKER        2700
ART           DAY           2700
ARTHUR        SRC           2700
BENNY         RAY           2700
BOB           KING          2700
CHARLIE       DAY           2700
CHARLIE       PRATT         2700
CHARLY        ALEXANDER     2700
CHARLY        FERRARI       2700
CHARLY        PAPAYANOP     2700
CLARA         ANDERSON      2700
CLARA         SPARCK        2700
CLARE         SHOW          2700
CLAUDE        ALDEN         2700
DALIAH        GROBE         2700
DAVID         KIM           2700
DICK          CODY          2700
ETHEL         ADKINSON      2700
```

```
FORTUNA          DILWORTH    2700
GIL              CHERNYI    2700
Inl:  Press any key (q for exit)
```

In this example, INL has output only the first 20 rows. Entering q will terminate further output and return the SQL prompt. Pressing any other key will cause INL to output the next 20 answer rows.

Example 5


This example demonstrates that toggling the command is an additional method of causing the output to display the number of rows returned in the answer.

In this example, for Microsoft systems, the number of auto owners whose salary is lower than \$3000, is printed.

```
SQL>count
SQL>time
SQL>unload:
SQL>outfil : PRN
SQL>SELECT 'The number of the auto owners whose
1>salary is less than $3000 is ', COUNT(*) FROM
2>Person WHERE Salary <3000;
SQL>outfil : CON
```

The following line will be output to the printer:

```
The number of the auto owners whose salary is less
than $3000 is 53
```

 Note the command outfil CON, typed in after query execution. The output file is closed and new file is opened on the command (when exiting from INL). The output file is opened in append mode so that results of many queries can be saved in it. Therefore, small files are printed only after they are closed.

Example 6

The following example shows, in the UNIX environment, the use of the SH command that executes a shell command from within INL.

```
SQL>SH ls -l
total 8
-rw-r--r--    1  root  sys  978    May 19 20:22  c.dir
-rw-rw-rw-    1  root  sys 2035    May 12 10:43  com_proc.c
-rw-rw-rw-    1  root  sys  992    May 15 12:55  com_proc.o
-rw-rw-rw-    1  root  sys 3128    May 12 10:43  dbc.h
-rw-rw-rw-    1  root  sys 23203   May 12 10:43  dbc_tcp.c
-rw-rw-rw-    1  root  sys 14400   May 15 12:55  dbc_tcp.o
-rw-rw-rw-    1  root  sys  2774   May 12 10:43  dbcs_err.h
-rw-rw-rw-    1  root  sys  2797   May 12 10:43  dbcscomm.h
SQL>
```

Example 7

Example 7 shows use of a script file, append.sql. The script contains three queries for inserting values into the AUTO FINANCE and PERSON tables and the select query for checking.

Page

Time

Count

```
! ***** Insert Into Table AUTO *****
INSERT INTO Auto VALUES('VOLVO', '4 DR', 'COUPE', 8, 120,118,2900,
'BLACK', 70, 'O262593464330109',720303,98224,10001);
! ***** Ok. *****
! ***** Insert Into Table FINANCE *****
INSERT INTO Finance VALUES('AMERICAN EXPRESS',
300,800,'ARCO',1600, 5, 'HARTFORD INSURANCE', 21000, 'RUTGERS',
', 'SEASHORE PROP.', 'SECURITY PACIFIC NATION.AL LA.',10001);
! ***** Ok. *****
! ***** Insert Into T able PERSON *****
INSERT INTO Person VALUES('MARRIS','JENNIFER','J','F',62,
'SINGLE',2,47,'GARLAND ROAD','AUSTIN',TX', 78712,'570-
5024','CONTROLLER', 9300, 0, 12, 2, 38, 16, 'FOLK DANCING', 10001);
! ***** Ok. *****
! ***** Test Query *****
SELECT COUNT(*) FROM Auto, Finance, Person WHERE
Person.PersonId =10001 AND
Auto.PersonId=Person.PersonId AND
Finance.PersonId=Person.PersonId;
```

To process all these queries, type in the file execution command. The command and output to the terminal are as follows:

```
SQL>_append
*****      Insert Into Table AUTO      *****
*****      Ok.          *****
*****      Insert Into Table FINANCE    *****
*****      Ok.          *****
*****      Insert Into Table PERSON     *****
*****      Ok.          *****
*****      Test Query *****
I 1      I
SQL>
```

Note that if, immediately after execution of append.sql, the EXEC command entered the SELECT query will be executed again, but without comment***** Test Query *****.

Example 8

Change PersonId values in all rows of the database that were inserted in Example 7:

```
SQL>UPDATE Auto SET PersonId = PersonId +233
1 >WHERE PersonId =10001;
INL : number of rows updated : 1
SQL>UPDATE Finance SET PersonId = PersonId +233
1 >WHERE PersonId =10001;
INL : number of rows updated : 1
SQL>UPDATE Perison SET PersonId =PersonId +233
1 >WHERE PersonId =10001;
INL : execute status : 2202
UPDATE Perison SET PersonId=PersonId+233 WHEREPersonId=10001;
      ^ unknown table
SQL>UPDATE Person SET PersonId =PersonId +233 1 >WHERE 1>PersonId
=10001;
INL : number of rows updated : 1
SQL>
```

Example 9

Remove the rows inserted in the Example 7 and updated in Example 8.

```
SQL>Time
SQL>Count
SQL>DELETE FROM Auto WHERE PersonId = 10234;
INL : start time : 16:07:03 end time : 16:07:04
INL : number of rows deleted : 1
SQL>DELETE FROM Finance WHERE PersonId = 10234;
INL : start time : 16:08:01 end time : 16:08:01
INL : number of rows deleted : 1
SQL>DELETE FROM Person WHERE PersonId = 10234;
INL : start time : 16:08:39 end time : 16:08:41
INL : number of rows deleted : 1
SQL>
```